

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 Reader and the associated tools. As OpenBerg Reader and the open-source world in general, this document is intended as a permanent work-in-progress, to be completed as we add features and specifications.

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 staff	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-source peer-written encyclopedia[17] and projects for an open-source peer-written on-line university[19]. All this is probably a good evolution, too, despite its excesses.

Of course, a number of problems are not solved by the current generation of tools. For instance, too many pieces of software (including Microsoft Word as well as a number of commercial e-book readers) rely on secret, always-changing, formats. Many documents written 10 years ago cannot be read nowadays, because the applications used to create them do not exist anymore, or because the format has changed, and old specifications were never disclosed. Closed formats are not future-proof. So-called Digital Rights Management and so-called Trusted Computing¹ technologies, in addition to being frighteningly close to privately-owned censorship [18, 12], also prevent any form of archiving or referencing by relying on the presence of some company-operated server or chip.

These are just two examples of pitfalls we need to solve by adopting open standards.

1.1 The OpenBerg Project

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.

COMPARISON WITH ADOBE READER Adobe Reader (formerly Adobe Acrobat Reader) [1] is quite a nice piece of software. It permits state-of-the-art pagination, multi-media content, a degree of interactivity, and is available on a large number of platforms. What it is not, however, is open-source, reimplementable as open-source, or easily extendable by open-source software. Indeed, Adobe Reader is based on PDF, and large

parts of PDF are covered by Adobe-owned patents. While Adobe has, so far, displayed a large degree of tolerance towards open-source software, the company has also set restriction on plug-ins and on the embedding of Adobe Reader and made very clear that no PDF generator should be in direct competition with their products [16]. These interdictions are worsened by the fact that Adobe has dropped support for a number of less-common platforms and may do so again at any time, and by the fact that Adobe seems to be moving towards more Digital Rights Management.

In addition, by opposition to OpenBerg Reader, Adobe Reader is a single-format reader.

COMPARISON WITH OTHER PORTABLE DOCUMENT FILE READERS (including KPdf and EVince). While some other PDF readers are quite impressive (most noticeably Mac OS X's built-in pdf display, KPdf and EVince), to the best of our knowledge, none supports interactive content and none are cross-platform.

COMPARISON WITH MICROSOFT READER Microsoft Reader (formerly Microsoft e-Book Reader) [3] is also a nice piece of software. It permits state-of-the-art pagination, multi-media content, a degree of interactivity, and is based on open standards. What it is not, however, is open-source, easily extendable by open-source software, or available on non Microsoft platforms. It is also linked to Microsoft Internet Explorer, to Microsoft Passport, and very much involved in Digital Rights Management.

In addition, by opposition to OpenBerg Reader, Microsoft Reader is a single-format reader.

COMPARISON WITH OSOFT THOUTREADER Osoft's ThoutReader [7] is also a nice piece of software, albeit with different objectives. It permits correct pagination, library management, it is based on open standards, it is open-source, it is cross-platform and seems free of any attached strings. It has, however, different priorities. It is, indeed, designed as a tool for technical documentation, rather than textbooks, hence it doesn't provide any support for interactive content or multimedia content. In addition, while correct, its pagination, managed by Swing, may prove insufficient for complex books. Finally, while the choice of the Java Platform and the Swing Toolkit as an implementation target is quite coherent, it is definitely not adapted to use on low-powered devices.

COMPARISON WITH ÉDITIONS VIGDOR'S LECTOR éditions Vigdor's Lector [2] permits correct pagination, a little degree of library management, it is mostly based open standards (but not completely), is cross-platform but is closed-source, in addition to which the choice of the Java Platform

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

COMPARISON WITH PLUCKR Pluckr [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. searching) for each format they recognise. For OpenBerg Reader, we decided to use the dual approach. At its core, OpenBerg Reader is only a mono-format reader, with a plug-in system used to convert books to this native format.

This choice has the following consequences

- + Isolation of most features from the actual format of the book makes it easier to add new features to OpenBerg.
- + Additional input filters can be added as plug-ins, using Mozilla's extension mechanism.
- We need to use a relatively heavyweight native format, powerful enough to represent features of numerous formats.
- Conversion from some formats might not match naturally the native format, especially if there is interactivity in the source book.

The choice of a native format is then decided by the following factors:

- rich pagination/typography
- semantic annotations (i.e. some text being described as *being a title* or *being a sidenote*)
- interactivity
- multimedia content
- mathematical formulas
- internationalisation
- patent-freedom
- openness
- screen-size independence (i.e. the same book should display properly on two screens of different sizes).

We evaluated the following formats:

GUTENBERG ASCII The Gutenberg ASCII format is essentially a non-format. It is open, patent-free and internationalisable does provide a few annotations but no pagination/typography, no semantic annotations, no interactivity, no multimedia content, no formulas.

L^AT_EX The L^AT_EX format is actually a powerful programming language intended for compilation before display. It offers extremely rich pagination, semantic annotations, it has extensions for interactivity and multimedia content, its speciality is mathematical formulas, it supports correctly internationalisation, it is patent-free and open. It is, however, much too complex for our tasks, fragile with respect to author's errors, requires a large baggage of fonts and rather slow processing – typically the compilation of a whole book – before displaying any single page.

DVI The Device Independent Format is a typographical representation format used as an intermediate language in the compilation of L^AT_EX. It offers extremely rich pagination, including mathematical formulas, can be extended with semantic annotations, some interactivity and multimedia content, it supports correctly internationalisation, it is patent-free and open. However, it is dependent on screen size.

RTF Rich Text File is a typographical description language intended as a universal format for exchange between text processors. It offers very rich pagination, no interactivity or multimedia content, little to no mathematical formulas, and probably supports internationalisation.

LECTOR 4 Lector's semi-proprietary file format is essentially a zip file containing meta-information about a book, accompanied by a bundle of Html files representing chapters and other resources. The fact that this format is proprietary is a show-stopper.

PLUCKER Plucker's open file format is essentially a set of Html files gathered in a Database. The format is, however, not clearly specified, and it is unclear whether it relies on PalmOS internals for interpretation.

PORTABLE DOCUMENT FORMAT This format is actually a powerful programming language intended for display. It offers extremely rich pagination, interactivity, multimedia content, mathematical formulas, internationalisation, etc. While almost open, it is, however, patent-encumbered, and restricted by usage license, which prevent, among other things, its generation by non-approved programs.

OEBPS This incomplete format actually only specifies a meta-data file and the organisation of XML or XHTML + CSS resources. Taken together with the specifications of XML, XHTML and CSS, JavaScript, MathML, it offers rich pagination/typography, semantic annotations, interactivity, the possibility of multimedia contents, internationalisation, it is patent-free and open.

OEBPS, with slight modifications, is the native format of OpenBerg

Reader.

The modifications are as follows

- Not all resources need to be referenced in the meta-data. The rationale is that we wish OpenBerg Reader to scale up to huge books (e.g. Wikipedia) with several millions of resources. In this cases, referencing all resources is both tedious and extremely memory-consuming.
- The bundle of files may be compressed using PKZip compression. A file `package.opf` inside the archive will then be expected to contain the meta-data. The rationale is that we need to be able to distribute books as one single file.
- Hyperlinks inside a book or between books use a new protocol `obbp:`. A hyperlink will then be interpreted as `obbp://identifier_of_the_book/path_to_the_resource\#path_inside_the_resource`, where
 - `identifier_of_the_book` is an ASCII string representing the destination book
 - `path_inside_the_resource` is any XPath expression.

For more details on OEBPS, see the official specifications [15] and the OEBPS FAQ [5].

2.2 Platform

The choice of a UI toolkit is constrained by the following factors:

- open-source
- cross-platform
- able to properly render manipulable XHTML + CSS in real-time
- able to support internationalisation.

2.2.1 Evaluation

We evaluated the following toolkits:

TROLLTECH's QT Impressive toolkit, cross-platform, supporting internationalisation, now open-source, but doesn't provide any rendering of XHTML + CSS.

TROLLTECH's QT + KHTML This addition permits XHTML + CSS rendering but removes portability and introduces dependencies towards the full KDE.

- 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 rendering, as well as manipulation of XHTML + CSS – at the cost of huge technical complications – but still doesn't complete the support for internationalisation.
- WXWIDGETS + WXWEBKIT** This addition permits XHTML + CSS rendering, but hasn't reached alpha version yet and still doesn't complete the support for internationalisation.
- SWING** While not open-source, this toolkit is part of the standard distribution of Java, which could make it a good candidate. It is cross-platform and supports internationalisation. Unfortunately, it is extremely memory-consuming, extremely slow, and its support of XHTML + CSS is extremely limited.
- SWT** This open-source toolkit for Java, as used in Eclipse, is mostly cross-platform, can render properly XHTML + CSS and supports internationalisation. However, it is not possible to manipulate the XHTML + CSS, as the tree is not accessible from Java. Accessing it from C++ requires diving into Mozilla.
- MOZILLA XPFE** Mozilla's Cross-Platform Environment [9] is open-source, widely deployed, mostly cross-platform, can render properly manipulable XHTML + CSS and supports internationalisation.

We decided to use Mozilla XPFE as our UI toolkit.

2.2.2 Corrolaries

The choice of XPFE entails the following corollaries:

- XUL** XUL is the XML User-Interface Language. It is the recommended manner of describing user interfaces for XPFE, using a simple XML format.
- XBL** XBL is the XML Binding Language. It is the recommended manner of turning XUL descriptions into reusable components.

CSS CSS is the Cascading Style Sheet language. It is the recommended manner of describing themes for XPFE interfaces.

DTD DTD is the Document Type Definition language. It is the recommended manner of internationalising XUL interfaces.

JAVASCRIPT The JavaScript language is the recommended manner of scripting XPFE user-interfaces.

.PROPERTIES Java/JavaScript `.properties` files are the recommended manner of internationalizing JavaScript programs.

XPCOM The Cross-Platform Component Object Model [14] is the recommended manner of communicating between a XPFE user interface and native components.

XPIDL The Cross-Platform Interface Description Language is the recommended manner of describing native XPCOM components.

C++ The C++ language is the recommended manner of implementing XPCOM components.

2.3 Compiler and build system

2.3.1 Under Unix

COMPILER GCC.

BUILD EXPERT SYSTEM Gnu Make.

PREPROCESSOR Sed.

ARCHIVE COMPRESSOR InfoZip's Zip.

For more information about the Unix build system, see Section C.4.

2.3.2 Under Windows

As specified by Mozilla's cross-platform build system,

COMPILER Visual C++.

BUILD EXPERT SYSTEM Gnu Make.

PREPROCESSOR Perl.

ARCHIVE COMPRESSOR InfoZip's Zip.

For more information about the Windows build system, see Section C.3.

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.

3 Modules

3.1 Overview

Figure 1 presents a high-level description of OpenBerg.

OpenBerg consists in two layers:

- the core.
- the user-interface

The core is developed in C++, using XPCOM and Mozilla's XPCOM libraries. On top of the core, several UI layers could take place, either using Mozilla's cross-platform UI, or using platform-specific toolkits for platforms which do not support Mozilla's toolkit – typically, Windows Mobile.

Communications between the core and the UI are brokered by XPCOM. This has the effect of limiting the development UI to languages which have XPCOM bridges. At the moment of this writing, XPCOM-able languages are

- C++
- XUL + JavaScript, using XPConnect (built-in Mozilla).

Note that other bridges are under development and might some day complete the landscape.

- Python, using PyXPCOM (nearly ready).
- Ruby, using RubyXPCOM.
- Perl, using PIXPCOM.
- Java, using JavaConnect
- .Net, using MonoConnect.

In the current state of things, however, the only options are C++ and XUL+JavaScript.

3.2 The installer

Firefox supports cross-platform installer mechanism for extensions. The meta-files for this installer are defined in

`src/installer`

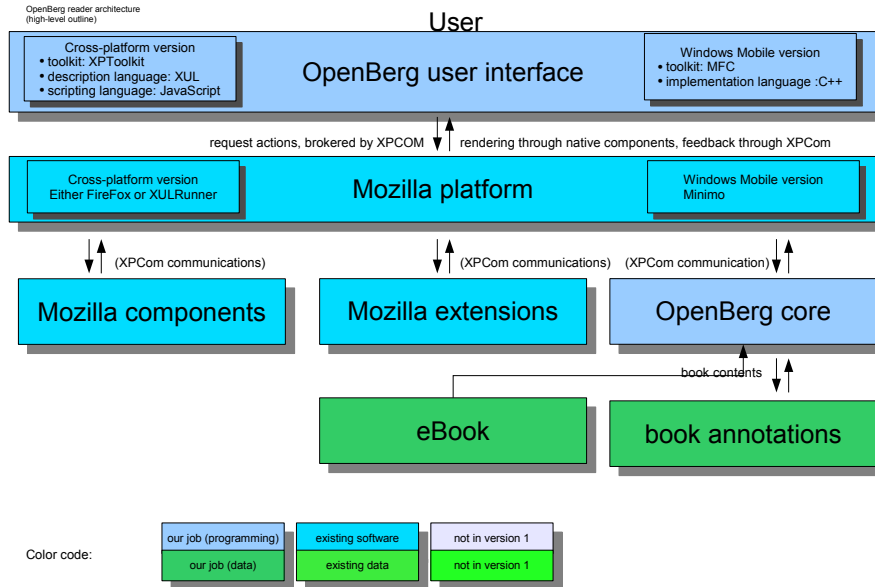


FIGURE 1: High-level outline of OpenBerg

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.

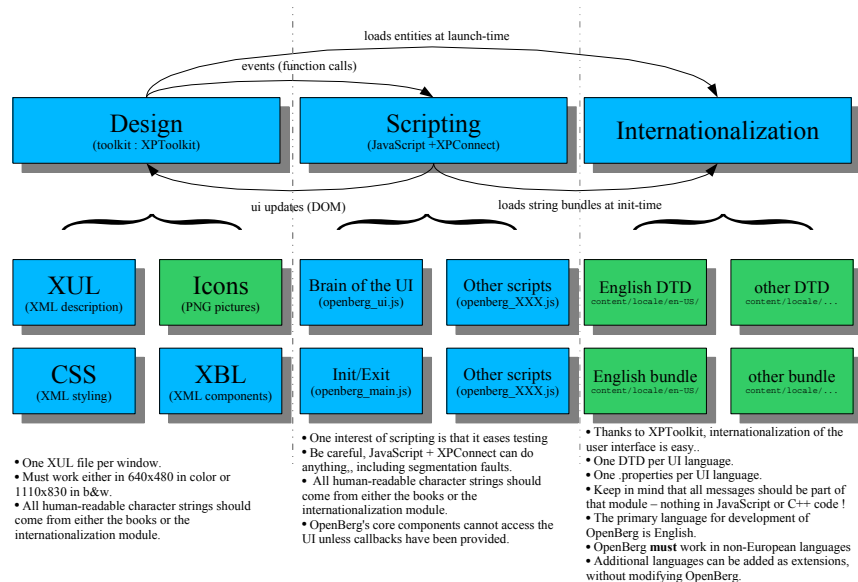


FIGURE 2: OpenBerg's cross-platform UI

Note that XUL takes advantage of the real JavaScript language (not one of its more lenient variants commonly found in browsers), and provides warning for the most common ambiguities or errors, provided the appropriate preferences are set[13].

3.3.2 Priorities

- The user-interface must be fully localised/localisable.
- The user-interface must work in 1024x768, black and white.
- The user-interface must work in 640x480, in colours.
- The user-interface must not depend on Firefox-specific code (so as to permit migration to XULRunner).
- The user-interface must not contain native code (so as to permit sanity checks and easier customisation).

3.3.3 Organisation

`src/content/openberg` The meat of the UI: XUL descriptions, JavaScript scripts, XBL bindings. Once installed in FireFox/XULRunner, these files

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

Note that these definitions may be extended/customised by extensions, using Mozilla's overlay mechanism.

`src/defaults/preferences` Default values for the preferences, as JavaScript scripts.

`src/icons/default` Window icons.

`src/locale` Locale definitions.

Note that additional locales may be added by extensions, using Mozilla's locale extension mechanism.

`src/locale/en-US/openberg` Definition of the US English locale, as

- a set of `.properties` string bundles
- a set of `.dtd` DTDs

Once installed in FireFox/XULRunner, these files may be referenced as `chrome://openberg/locale/...` This URL will point either to `en-US/openberg/...`, `fr/openberg/...`, or to the corresponding directory, depending on the current locale of the system and the set of OpenBerg locales installed.

`src/locale/en-US/openberg/help` Online help e-Book for the US English locale.

`src/locale/en-US/openberg` Definition of the French locale, as

- a set of `.properties` string bundles
- a set of `.dtd` DTDs

`src/locale/en-US/openberg/help` Online help e-Book for the French locale. Once installed in FireFox/XULRunner, these files may be referenced as `chrome://openberg/locale/...` This URL will point either to `en-US/openberg/...`, `fr/openberg/...`, or to the corresponding directory, depending on the current locale of the system and the set of OpenBerg locales installed.

`src/skin/default` The default theme for OpenBerg, as

- a set of `.css` style sheets
- a set of PNG icons
- the PNG logo of OpenBerg.

Note that this skin may be replaced by an extension.

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

- NSPR, Netscape's comprehensive and portable counterpart to libc
- Hans Boehm's garbage-collector
- the lightweight, in-process, Berkeley DB
- zip (de)compression library ZLib
- low-level XML parser driver Expat
- ...

all of which we may use.

The core will be divided in logical units, with the following roles

LOADER Initialisation of components.

UTILITIES high-level DOM/XML management, RDF management, OEBPS management, String utilities, etc.

LIBRARY book loading, caching, storing and recovering book information.

ANNOTATIONS serialisation of bookmarks, application/deapplication of annotations.

FIND locating possibly complex content in the book.

Figure 3 presents an overview of the structure of OpenBerg's core.

3.5.2 A few conventions

CLASS NAMES To differentiate OpenBerg's classes from Mozilla's classes, we prefix their name with **ob**, while Mozilla's classes see their name prefixed with **ns**. Example: **obBookZipped**, **nsStream**.

INTERFACE NAMES To differentiate OpenBerg's interfaces from Mozilla's interfaces and from implementation classes, we prefix their name with

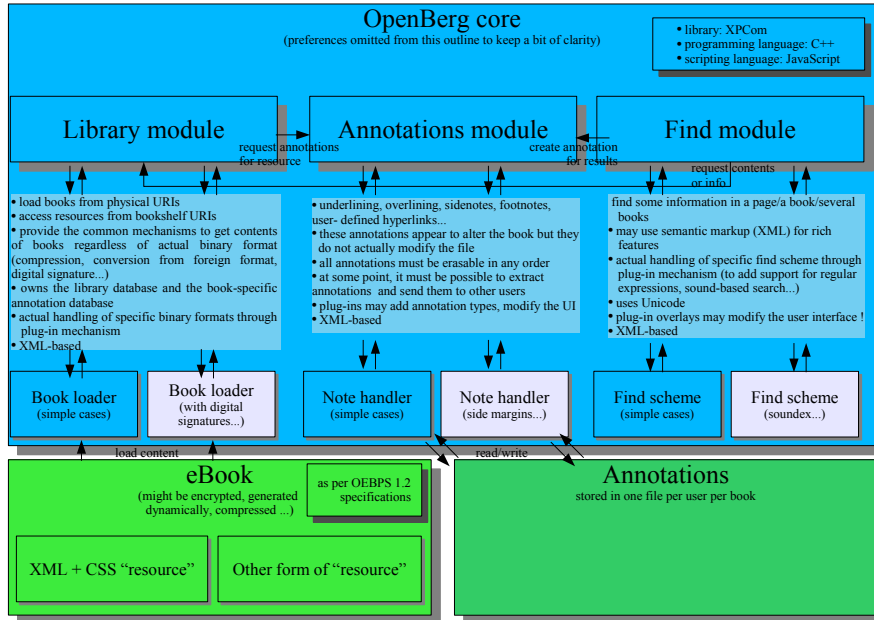


FIGURE 3: OpenBerg's core

obI, while Mozilla's interfaces see their name prefixed with nsI. Examples: obIBook, 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 Hoff'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 `nsIModule`, 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 `nsIFactory` 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 `obIBook` 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 `obIBookLibrarian`.

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 cloture of OpenBerg. They are accessible through the *Persistence Service*, itself a component implementing interface `obIPersistenceService`.

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

- information on how to find books (i.e. mapping from bookshelf uri to physical uri)
- book metadata (e.g. last opening date, author's name, summary, etc.)

Main repository-related functions are usable as soon as OpenBerg is loaded and until the cloture of OpenBerg. They are accessible through the *Persistence Service*, itself implementing `obIPersistenceService`.

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

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

```

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

```

Later, the user may wish to add custom hyperlinks

This¹ is an important text².

or

```

<span class=" user:emphasis">
  This
  <span class=" user:hyperlink">
    <a href=" obbp://grammar/nominative">1</a>
  </span>
  is an
  <span class=" emphasis">important</span>
  <span class=" user:hyperlink">
    <a href=" obbp://grammar/accusative">2</a>
  </span> text
</span>.

```

Now, if the user attempts to find all references to book **grammar**, the search will yield at least two results, both of which will be highlighted

```

<span class=" user:emphasis">
  This
  <span class=" search:occurrence">
    <span class=" user:hyperlink">
      <a href=" obbp://grammar/nominative">1</a>
    </span>
  </span>
  is an
  <span class=" emphasis">important</span>
  <span class=" search:occurrence">
    <span class=" user:hyperlink">
      <a href=" obbp://grammar/accusative">2</a>
    </span>
  </span>
  text
</span>.

```

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_annotation` before that of `another_annotation` ?

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.

A third choice is to manually implement the merging of annotations. However, this will most likely cause problems with annotations which actually modify the text. One such annotation is the addition of custom hyperlinks with a number. If we are to make this choice, we will most likely forbid certain kind of annotations, most likely restricting their influence on the text to CSS alterations.

The specifications have yet to be defined.

Dependencies

This unit depends on

LIBRARY Annotation uses Library to access book contents and book storage.

UTILITIES Annotation uses various functions of Utilities.

3.5.6 Find

Overview

The Find unit contains the low-level primitives used by the UI to perform complex queries in a chapter or in a book.

Details

The specifications of the Find unit have not been defined yet.

This unit should provide the low-level features required to build complex queries for content – with plug-ins to provide additional manners of searching, such as (language-specific) phonetic search or synonyms, paragraph-based searching, index-based searching. . .

In practice, the Find unit will permit the following

- management of find plug-ins
- loading and streaming of XML content to plug-ins
- transformation of the results in annotations.

A complementary unit of the UI module will provide the associated interface. Plug-ins should be defined as XPCOM components, installed using Mozilla's extension system and may modify that interface through Mozilla's UI overlay system.

Note

The current version of OpenBerg has an implementation-in-progress of a live search feature, developed as a part of the cross-platform UI module.

Most likely, this implementation will stay in the future versions of OpenBerg Reader as *simple search* and will be completed by a dialog-based *advanced search* for queries.

Dependencies

This unit depends on

LIBRARY Find uses Library to access book contents.

ANNOTATIONS Find uses Annotations to present its results.

UTILITIES Find uses various functions of Utilities.

3.6 Extending OpenBerg

OpenBerg is extendable either by modifying the source code or by applying standard Firefox/XULRunner extensions. The Mozilla Developer Connection maintains a subsite dedicated to this topic [4].

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 UI, the Library's asynchronous loading mechanism and their interactions, as well as the cross-book hyperlink mechanism.

Since then, we have been able to add interesting yet easier features such as

- opening compressed books
- live search (prototype)
- organising books and searching for them in the library
- zooming
- full-screen browsing
- ...

We believe that our work on the infrastructure is robust enough that the features we intend to add in the near future will also be much easier than the beginnings of the project.

4.2 One Laptop Per Child

(Extract from OLPC's website)

Reader

The number one required feature for an ebook reader HAS to be the clearest, most easily readable text display possible on the hardware. Anti-aliasing or font-smoothing can be a great asset if implemented well or a horribly "smudgy" mess if implemented poorly...

Useful features might include:

- User definable "style sheets" to control font size and type face, etc.
- Display graphics in various formats; both vector and bitmap formats

- Display complex formulae
- Link in multimedia elements such as sound or video files
- Bookmarking, preferably sharable via network
- Annotations, preferably sharable via network
- 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.
- Human readable, yet efficient markup language
- Compressed files (easily decompressed to access original files)
- Simple, intuitive interface
- Localizeable
- Supports complex-text language rendering (for example, arabic and thai)

Library

Not all eBook readers do an effective job of organizing the books stored in the library. Some provide rudimentary cataloging features and some offer nothing more than a jumble of titles and authors. On a machine with only enough storage to hold a dozen or fewer titles this is not much of a problem, but if the machine can store many dozens or hundreds of books, articles, stories, essays, etc. cataloging, organizing and searching will become crucial.

Useful features might include:

- A standardized schema of metadata for categorizing work
 - Sorting on any of the fields in the schema, plus system generated data like last page read, etc.
 - Full-text search capability across all titles - MUST be speedy so indexing will be necessary
 - Network access to larger library to allow downloads of additional titles
-

It is our intention to comply, as much as possible, with these guidelines.

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'Reily, 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 Hoff. 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>.
- [13] The MozillaZine Community. Setting up extension development environment. URL http://kb.mozillazine.org/Setting_up_extension_development_environment.
- [14] Doug Turner and Ian Oeschger. *Creating XPCOM Components*. The Mozilla Foundation, 2003. URL <http://www.mozilla.org/projects/xpcom/book/cxc/>.

- [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 staff

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 complete tasks. Some tasks consist only in 10 lines of JavaScript, while others demande thousands of lines of C++. M1, M2, M3, M4 represent the Milestones, Milestone 4 being the first release candidate of OpenBerg Reader 1.0. We are currently moving towards Milestone 2.

TABLE 1: List of features planned for OpenBerg Milestone releases

	M1	M2	M3	M4	Beyond	Current
Platforms supported						
Linux	Works	Works	Works	Works	Works	Works
Windows 98+	Works	Works	Works	Works	Works	Works
Mac OS X/PPC	Works	Works	Works	Works	Works	Works
Mac OS X/Intel	Works	Ongoing	Works	Works	Works	Works
Windows Mobile		Ongoing	Ongoing	Works	Works	Works
Linux Embedded	Untested	Untested	Untested	Untested	?	Untested
RiscOS	Untested	Untested	Untested	Untested	?	Untested
PalmOS			?	?	?	?
Nature						
Firefox extension	1.0.x	1.5.x	Works	Works	Works	Works
XULRunner application		1.8.0.1	Works	Works	Works	Ongoing
Input formats supported						
OEBPS subsets	Works	Works	Works	Works	Works	Works
Zip-compression		Works	Works	Works	Works	Works
Gutenberg e-Books		Works	Works	Works	Works	Ongoing
Microsoft LIT					?	
Adobe PDF					?	
Lector 4			Works	Works	Works	
(huge books)			Works	Works	Works	Specifying
Book contents (note: this content is almost entirely handled by Gecko)						
XHTML	Works	Works	Works	Works	Works	Works
CSS 3.0	Works	Works	Works	Works	Works	Works
MathML	Limited	Limited	Limited	Limited	Works	Ongoing
SVG subset	Works	Works	Works	Works	Works	Works
Full SVG	Works ¹	Works ¹	Works ¹	Works ¹	Works	Works ¹

(continued on next page)

¹Requires platform-dependent plug-ins

(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	Works ¹
Flash applets	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹
PNG images	Works	Works	Works	Works	Works	Works
GIF images	Works	Works	Works	Works	Works	Works
JPG images	Works	Works	Works	Works	Works	Works
OGG sounds	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹
MP3 sounds	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹
WAV sounds	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹
MPEG videos	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹
QuickTime videos	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹
Media Player videos	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹
⋮	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹
TEI comprehension					Works	
Library management						
Books database	Works	Works	Works	Works	Works	Works
Displaying library	Ongoing	Works	Works	Works	Works	Works
Finding in library		Works	Works	Works	Works	Works
Importing catalogues			Works	Works	Works	Specifying
Links between books	Works	Works	Works	Works	Works	Works
Opening distant books	Works	Works	Works	Works	Works	Works
Annotations						
Bookmarks		Works	Works	Works	Works	Ongoing
Annotation framework			Works	Works	Works	Specifying
Annotation management			Works	Works	Works	
Highlighting			Works	Works	Works	
User hyperlinks			Works	Works	Works	
Exporting annotations			Works	Works	Works	
Importing annotations			Works	Works	Works	
Searching						
Livesearch in a chapter		Works	Works	Works	Works	Ongoing
Livesearch across chapters		Works	Works	Works	Works	Ongoing
Saving search results			Works	Works	Works	
Exporting search results				Works	Works	
Complex queries				Works	Works	
Ignore case		Works	Works	Works	Works	Ongoing
Search for tags		Works	Works	Works	Works	Ongoing
Phonetic search					Works	
Support for plug-ins			Works	Works	Works	
Indexation				?	?	

(continued on next page)

(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 of OpenBerg Reader)						
Conversion from \LaTeX		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 proje
```

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 don't work. But you're free to try.

C.2 Obtaining the code – as a developer

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. 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).

Overview

This build mechanism uses the official guidelines for creating extensions with native components for Firefox/Mozilla.

The system is the following :

1. Download the source code of Firefox.

2. Make sure that you have what's necessary to build Firefox (the best way to do this is to actually try and build Firefox).
3. Download OpenBerg.
4. Tell Firefox's build system to build OpenBerg as an extension.
5. Build Firefox, and OpenBerg along the way.

The actual set of tools and prerequisites you need to build Firefox depends on your platform and configuration. You may find these instructions, as well as the source code for Firefox, on

<http://www.mozilla.org/build>.

C.3.1 Set up the configuration

1. Download the source code of Firefox. Any version of the 1.5.x series should be good. You may find it at <http://www.mozilla.org/build>.
2. Optionally, Build Firefox, using the instructions available on <http://www.mozilla.org/build>.
Most likely, if you are under Windows, you will meet numerous problems during this step. For help on this, consult Mozillazine's forums at <http://www.mozillazine.org>.
3. If you haven't done so already, download the source code of OpenBerg Reader.
4. Copy directory `openberg2`, from OpenBerg Reader's source code, into directory `extensions`, in Firefox's source code.
5. Alter your Firefox configuration (most likely, this will be a file called `.mozconfig`) to

- permit shared libraries

```
--ac-add-option --enabled-shared
```

- compile OpenBerg

```
--ac-add-option --enable-extensions=(any extension you want),xmlextras,
```

C.3.2 Build the program

Compile Firefox exactly as per the official instructions (still on <http://www.mozilla.org/build>). If the compilation succeeds, OpenBerg will also have been built and installed on this version of Firefox.

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

<http://forums.mozillazine.org>.

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 Firefox on your computer.

If you have specified an OBJDIR, go to directory

```
<yourmozillasourcedirectory>/<yourOBJDIR>/extensions/openberg2/src/installer
```

otherwise, go to directory

```
<yourmozillasourcedirectory>/extensions/openberg2/src/installer
```

Either way, type

```
make xpi
```

If this step succeeds, it will create a file called

```
openberg.xpi
```

This file is a standard installer for Firefox 1.5.x. You will need to open it from Firefox and follow the instructions.

C.3.4 Install Firefox

This is an optional but useful step, especially if you do not have Firefox installed on your computer.

To install Firefox, follow Mozilla's official instructions. If you have just compiled Firefox with OpenBerg, your version of Firefox will contain OpenBerg.

C.3.5 Enjoy

Congratulations, OpenBerg is now installed. Your Firefox should now contain an item in the Tools menu for opening OpenBerg. If you have installed OpenBerg under Mozilla or XULRunner, you may have to go to the following URL:

```
chrome://openberg/content/mainwindow.xul
```

As of this writing, OpenBerg is usable. It is, however, an alpha version. If you meet strange behaviours, please report them on http://www.openberg.org/bug_report.php.

If you want a full report of error messages, you should set variable `NSPR_LOG_MODULES` to `openberg:5`, most likely by writing

```
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

- A developer package for the corresponding version of Firefox

UNDER DEBIAN this package is called libxul-dev, and is available as part of the standard repositories

UNDER UBUNTU this package is called firefox-dev, and is available as part of the standard repositories

UNDER RED HAT Mandrake and other Rpm-based systems, this package is called firefox-devel, and is available as part of the standard repositories

UNDER MAC OS X this package is called firefox-dev, and is available as part of Fink

OTHER OSES might provide different packages containing

- the full headers and interfaces of Firefox (note that gecko-sdk is not sufficient !)
- either the program `firefox-config` or the programs `xulrunner-config` and `nspr-config`
- the program `xpidl`

UNDER ALL PLATFORMS you can obtain this by compiling the source tree for Firefox

- zip 2.30 (by InfoZip, other zip utilites should also work)
- gcc 3.2 (other compilers might also work)
- GNU make 3.8 (other make utilities might also work)
- GNU sed 4.1.4 (other versions of sed might also work)

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 The tool used to compile XPIdl interfaces (fortunately, it's called `xpidl`).

MOZILLA_CONFIG The tool used to configure the compilation of most libraries. Depending on your installation, it's going to be called either `firefox-config` or `xulrunner-config`.

NSPR_CONFIG If you are using `firefox-config`, set that variable to `echo`. Otherwise, fill in with the path to `nspr-config`.

GECKO_IDLS On most platforms, you don't have to change this.

GECKO_DEFINES On most platforms, you don't have to change this.

After this, do

```
make -f Makefile.linux installer
```

Note that the Makefile file is named `.linux` as it has only been regularly tested under Linux. According to our experiments, it should work on any version of Unix.

This should create a file called

```
openberg-(youros)-(yourcpu).xpi
```

This file is a standard installer for Firefox. You will need to open it from Firefox, accept the choices, leave Firefox and restart Firefox. Congratulations, OpenBerg is now installed. If you have succeeded, please come and drop us a line on our forums, on <http://www.openberg.org>, specifying your configuration.

To actually run OpenBerg, go to your "Tools" menu and choose "OpenBerg ebook Reader".

Unfortunately, there are almost no book available for testing. We plan to make books available as soon as time permits it.

C.4.2 *If you want to reinstall OpenBerg*

The Makefile contains a quick-and-dirty utility to rebuild OpenBerg and reinstall it without having to uninstall it first. You will need to close Firefox, though.

To do so, you will need to set

```
PROFILE_DIRECTORY
```

to get it to point to the profile directory for which you have installed OpenBerg (this is different on each computer, the name seems partly randomly

generated by Mozilla).

Once you have the correct directories, do make update Et voil, OpenBerg has been updated.

Compilation messages

At the end of compilation, you will receive a message along the lines of

```
libxpcom.so => /usr/lib/libxpcom.so (0xb7f1a000)
[...]
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x80000000)
```

this is perfectly normal. This is just a check that the library you compiled is complete and will thus be loaded as a plug-in by Mozilla.

If you see a message warning about unidentified symbols, worry. OpenBerg will not be loaded at all by Mozilla, and Mozilla will fail to print any error message. To fix this, you will need to resolve the symbols:

- either by adding the missing libraries, if that's the problem
- or by defining the functions or methods the linker is looking for.

A utility such as `nm` might be helpful as it can demangle the name of the missing symbols (i.e. print them in a human-readable way).

C.4.3 Enjoy

Congratulations, OpenBerg is now installed. Your Firefox should now contain an item in the Tools menu for opening OpenBerg. If you have installed OpenBerg under Mozilla or XULRunner, you may have to go to the following URL:

```
chrome://openberg/content/mainwindow.xul
```

As of this writing, OpenBerg is usable. It is, however, an alpha version. If you meet strange behaviours, please report them on http://www.openberg.org/bug_report.php.

If you want a full report of error messages, you should set variable `NSPR_LOG_MODULES` to `openberg:5`, most likely by writing

```
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 !

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

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

D.1.2 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

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) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
 - (a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - (b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - (c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, 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 executable. However, as a special exception, the source code 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.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program 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 as such parties remain

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 version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, 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

11. 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.

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 warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR 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 could even be mouse-clicks or menu items—whatever suits your program.

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

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James
Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
```

```
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

D.2 GNU Lesser General Public License, version 2.1

D.2.1 Copyright

Version 2.1, February 1999

Copyright © 1991, 1999 Free Software Foundation, Inc.

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

permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

D.2.2 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for

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 de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

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.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete 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 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-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

2. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To

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.

3. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

4. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten

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

as such parties remain in full compliance.

8. 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 Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.
9. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library 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 with this License.
10. 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 Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library 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 Library.

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.

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

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; 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.

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!