

Dokeos 1.5 Developers Manual



Roan Embrechts & Hugues Peeters, April 2004
Distributed under Gnu Free Documentation License
© Dokeos & Universiteit Gent, Université catholique de Louvain

Dokeos 1.5 Developers Manual

Table of contents

1. Introduction
2. Dokeos code structure
3. Dokeos API
4. Dokeos databases structure
5. Coding conventions proposals
6. Template for addons
7. Work method & design methodology
8. Communication between developers
9. Software development in general
10. User interface design
11. User input
12. Promoting and spreading Dokeos
13. Software development in general
14. Gnu Free documentatin license
15. Edit this manual

1. Introduction

There are different Dokeos manuals for :

- Students/trainees
- Teachers/trainers
- Developers (programers)
- and Administrators

The Student and Teacher manuals have been translated in many languages. Check <http://www.dokeos.com> for the latest version.

This is an introductory manual for developers of Dokeos, a free e-learning and course management platform. The manual is neither complete nor perfect, and all suggestions and contributions are welcome. Dokeos is an evolution from the original Caroline e-learning software (version 1.4.2); Dokeos 1.5 and upwards is based on this code and adds many bugfixes, improvements and new features. The software comes under the GNU General Public License.

2. Dokeos code structure

Dokeos consists of several distinct pieces of functionality that we call modules or tools: the agenda, documents, links, forum... The code for each of these is placed in a separate folder. The code of Dokeos is mostly procedural. It is slowly evolving and becoming more mature. Functions are used more and more in the newer code, we think about encapsulation and layered design... Objects are not used very often. There is also a folder for general code that can be shared across tools: the 'inc' folder.

Inc for include

The inc folder – short for include – is a general directory that is used by all standard modules, and provides the global Dokeos variables, the standard header & footer, and many useful functions. The folder contains files and some subdirectories. In the **inc/lib** folder you can find library code, these files provide common functionality: functions for file managing, database queries, general tool functions... together these files provide an *Application Programming Interface* or API. In the **inc/conf** folder you can find configuration files for several tools. Note that some of these files are only tentative, that is they are provided but do not yet interact with the tools. These folders are a work in progress.

NOTE: The inc folder was called include in Dokeos versions before 1.5.

Since Caroline/Dokeos version 1.4, making your own module is quite easy: you only have to use three include statements, and you gain the Dokeos header, footer and global variables. See [further](#) for a template to start developing your own code. We encourage you to use the template for new development. Also, please respect the [coding conventions](#); these will keep the code of Dokeos consistent and make the program more readable for everyone.

Please keep your new code as modular as possible. Divide it into different sections, use functions, use comments often to explain stuff and divide your code into logical pieces. Use functions as often as possible, these really help the modularity of the code. If you want to go a step further, you can use object oriented programming, PHP has some support for it (and it is improved in PHP 5). Please write at least one or two lines at the start of each script file to explain the purpose of it.

TIP: If you are new to PHP functions: they don't work the way you would expect coming from almost any other programming language. PHP functions have no access to global variables, unless in the function code block, you explicitly declare the needed variables as global \$varname; This is the source of many bugs and lots of frustration...

Global code and course-specific code

Dokeos' application code is global. Changes apply to all courses on the server. There are ways, however, to write code and only use it for one specific course. This is a neat solution for special requirements not shared by all courses, and it is also handy to test and debug a customisation in "real life" conditions before applying it globally. The easiest way is to "Add a link on the start page", selecting the second possibility "Link to a site", and providing the URL of the custom code. You could for example store this code in the plugin code folder.

3. Dokeos API

There are several functions that can be useful for developers. In the future we will try to make API docs available, using e.g. [PhpDocumentor](http://pear.php.net/package/PhpDocumentor) (<http://pear.php.net/package/PhpDocumentor>). The function libraries are found in the inc – short for include – folder. Before you start programming, please read the inc folder, the inc/lib and the inc/conf folder – then you are aware of what Dokeos already has to offer.

The most recent additions are the **claro_main.lib.php**, a file in the inc/lib/ folder that has a few functions dealing with database queries and tool functions, and the **main_db.lib.php**, which contains database query functions (which are untested at the moment). These are constantly being expanded. There are also text libraries, libraries dealing with files...

For example, there is a general function to display the tool title; there is a function to show a [student view / course manager view] option, to show an html "dialog box"... There are no specific database functions yet, only a few general ones, but this is an area that we're improving right now. If you work on Dokeos, and need a query that returns e.g. the properties of a group, you should add a function `getGroupProperties(necessary parameters)` to the `claro_main.lib.php` and call that function. It's a bit more work in the beginning, but it will make your code smaller and easier to maintain, it also means there is only one place to change the query if it needs changing – no more updating 700 copies of the same query.

Main library

The following functions are provided in the file `claro_main.lib.php`. This file is included by default inside every Dokeos file. To be more precise, it is included when you include the **`claro_init_global.inc.php`** file.

Functions in this API file:

```
//DATABASE
function claro_sql_query($sqlQuery, $dbHandler = '#' )
function claro_sql_query_fetch_all($sqlQuery, $dbHandler = '#')

//FAILURE MANAGEMENT
function claro_set_failure($failureType)
class claro_failure -> function set_failure($failureType)
class claro_failure -> function get_last_failure()

//TOOL FUNCTIONS
```

```
function claro_disp_tool_title($titleElement)
function claro_disp_tool_view_option($isStudentView = false)
function claro_disp_message_box($message)
function is_allowed_to_edit()
```

4. Dokeos databases structure

Dokeos can be installed using only a single database, or multiple databases. Normally the multiple databases option is used. There are two main databases, one for general program stuff like users, courses and faculties; and another for statistics. Every course also gets a database.

Somebody should draw an E/R model, I haven't found the time for this yet. For now, you can learn a lot just from looking through PhpMyAdmin at the database structure, it's fairly easy.

About the spelling: earlier versions were developed by native French speakers, that's why there are some oddities here and there (cours instead of course, faculte instead of faculty...)

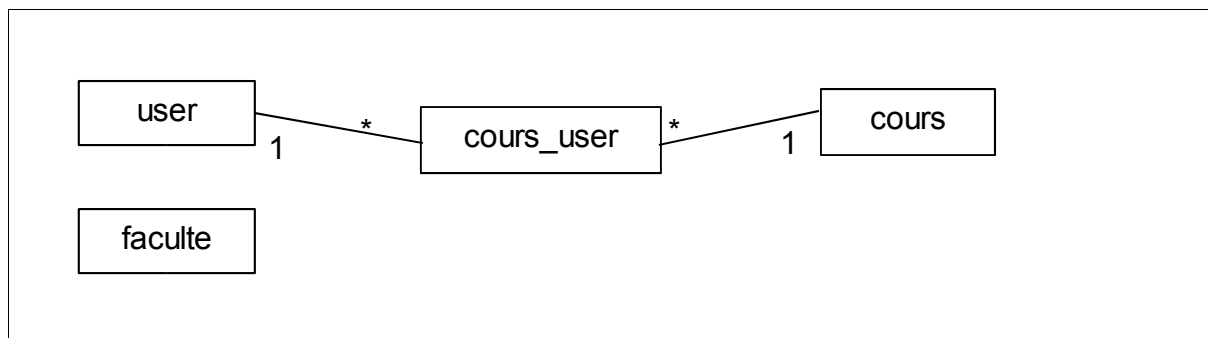
There's a rumor that the latest version of PhpMyAdmin can create E/R diagrams automatically. Let's investigate this...

One user can follow many courses.

One course can have many users.

We define an extra table for relation between users and courses: cours_user.

Every course belongs to exactly one faculte, one faculte can have many courses.



5. Coding conventions proposal

Language

The code and comments and developer instructions you write should be in English, no matter what your native language is. It's great if manuals are available in many languages, but code with comments and variables in English, French and Spanish is hard to understand.

Whitespace

Adding a blank line between lines of code creates paragraphs of code and helps to visualize it. You don't want to read books that have a thousand pages filled with line after line and no breaks; nobody wants to read code like that either.

Also, using spaces around operators and after comma's can help to make the code more readable.

```
function something (param a, paramb)
if ( (a && c) || !b)
a = b + c:
```

Indentation

Use tabs for indentation. Generally, one tab equals 4 whitespaces.

Brackets

Please use this notation

```
if ($wwwww > $jjjjj)
{
    code...
}
```

instead of this one

```
if ($wwwww > $jjjjj) {
    code...
}
```

Variables and functions

- We use under_scores to separate words, instead of camelCase.
- Variables and functions are named this way:

```
$my_variable
function test_database_api()
```

and not this way:

```
$myVariable
function testDatabaseApi()
```

- If possible, try to limit the variable name to no more than three words.
- Try to name the variables in singular mode rather than plural, whatever the context.

Namespaces

Please try to avoid using global variables directly. PHP does not yet have the concept of namespaces: every variable used is global and can be seen in the entire program (except in functions). To avoid problems, you have to rethink your software design. Make more use of functions and/or objects, try to avoid using global variables directly in your code. Don't use 700 times `$global_some_var`, use the function `get_global_var()` instead. Your program will become easier to maintain, to fix bugs in...

Constants

- Constants should be in UPPERCASE, and if MULTIPLE_WORDS_LIKE_THIS.

Classes

- Class names should begin with an uppercase letter (User, not user).

Comments

Comments should make the code easier to read; do not explain what is obvious; assume that the reader knows the programming language.

- Don't use Perl like comments `#`
- Always use the C like comment `/* */`
- Use the C++ like comment `//` for very short comments. These comments should be above the concerned code, if at all possible.

Each Dokeos file should begin with the following heading. If you contribute to a file, add your name to the list of contributors to that file.

```
/*
-----
Dokeos - elearning and course management software

Copyright (c) 2004 Dokeos S.A.
Copyright (c) 2003 University of Ghent (UGent)
Copyright (c) 2001 Universite catholique de Louvain (UCL)
Copyright (c) Jean Programmer (jean@somewhere.net)

For a full list of contributors, see "credits.txt".
The full license can be read in "license.txt".

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.
```

See the GNU General Public License for more details.

Contact: Dokeos, 181 rue Royale, B-1000 Brussels, Belgium,
info@dokeos.com

*/

The sections are structured according to the conventions below

```
/*
=====
                        LEVEL 1
=====
*/
```

```
/*
-----
                        LEVEL 2
-----
*/
```

For the class and function interfaces, use the conventions from the Java documentation, so we can retrieve these comments later to build complete documentation with PHP Documentor (<http://pear.php.net/package/PhpDocumentor>, from the [Pear](#) collection of reusable PHP components).

Details of the Java Documentation convention can be found at :
How to Write Doc Comments for the Javadoc Tool
<http://java.sun.com/j2se/javadoc/writingdoccomments/index.html>

A JavaDoc comment example:

```
/**
 * Draws as much of the specified image as is currently available
 * with its northwest corner at the specified coordinate (x, y).
 * This method will return immediately in all cases, even if the
 * entire image has not yet been scaled, dithered and converted
 * for the current output device.
 *
 * If the current output representation is not yet complete then
 * the method will return false and the indicated {@link ImageObserver}
 * object will be notified as the conversion process progresses.
 *
 * @author Sami Shaio
 * @author Arthur van Hoff
 *
 * @param img      the image to be drawn
 * @param x        the x-coordinate of the northwest corner of the
 *                 destination rectangle in pixels
 * @param y        the y-coordinate of the northwest corner of the
 *                 destination rectangle in pixels
 * @param observer the image observer to be notified as more of the
 *                 image is converted. May be null
 * @return         true if the image is completely loaded and was
 *                 painted successfully;
 */
```



```
*           false otherwise.
* @see      Image
* @see      ImageObserver
* @see      imageWrite
* @since    JDK1.0
* @deprecated imageWrite performs it faster
*/
```

```
function drawImage($img, $x, $y, $observer);
```

6. Template for add-ons

Use the following template to start developing your own code. This template can also be found in the “plugin” folder of the Dokeos code. The first strange symbols “# \$Id\$” are directions for the CVS – it means the code system can add version and name information there.

```
<?php # $Id$

/*
-----
Dokeos - elearning and course management software

Copyright (c) 2004 Dokeos S.A.
Copyright (c) 2003 University of Ghent (UGent)
Copyright (c) 2001 Universite catholique de Louvain (UCL)
Copyright (c) Jean Programmer (jean@somewhere.net)

For a full list of contributors, see "credits.txt".
The full license can be read in "license.txt".

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.

See the GNU General Public License for more details.

Contact: Dokeos, 181 rue Royale, B-1000 Brussels, Belgium,
info@dokeos.com
-----
*/

/*=====
           DOKEOS INIT
=====*/

// name of the lang file which needs to be included
// 'inc.php' is automatically appended to the file name
// $langFile = "langFile";

@include("../inc/claro_init_global.inc.php"); // settings initialisation

/*
Optional extra http header
If you need to add some HTTP/HTML headers code
```

```

like e.g. javascript functions, put them here.
*/
// $httpHeadXtra[] = "";
// $httpHeadXtra[] = "";
// ...
//
// $htmlHeadXtra[] = "";
// $htmlHeadXtra[] = "";
// ...

$nameTools = ""; // title of the page (comes from the language file)

$query_string=''; // used for the breadcrumb
                  // when one needs to add a parameter after the
filename

@include($includePath."/claro_init_header.inc.php");

/*=====
      MAIN CODE
=====*/

// PUT YOUR CODE HERE ...
echo "<h1><center>Hello world!<center></h1>";

/*=====
      DOKEOS FOOTER
=====*/

@include($includePath."/claro_init_footer.inc.php");
?>

```

Adding JavaScript to the Dokeos Html header

If you want to include JavaScript, you should add it in the *\$htmlHeadExtra* variable, which must be set *before* including the Dokeos header. If you use double quotes in your JavaScript, then it's a good idea to use single quotes for your variable. Otherwise, you'll have to escape all double quotes in the JavaScript by adding a backslash in front of it: `\`. Warning: using single quotes means that you cannot use php variables in the Javascript, such as language variables. If you need them, you'll have to use double quotes and use the backslash `\` trick.

```

$htmlHeadXtra[]='<script type="text/javascript" language="JavaScript">

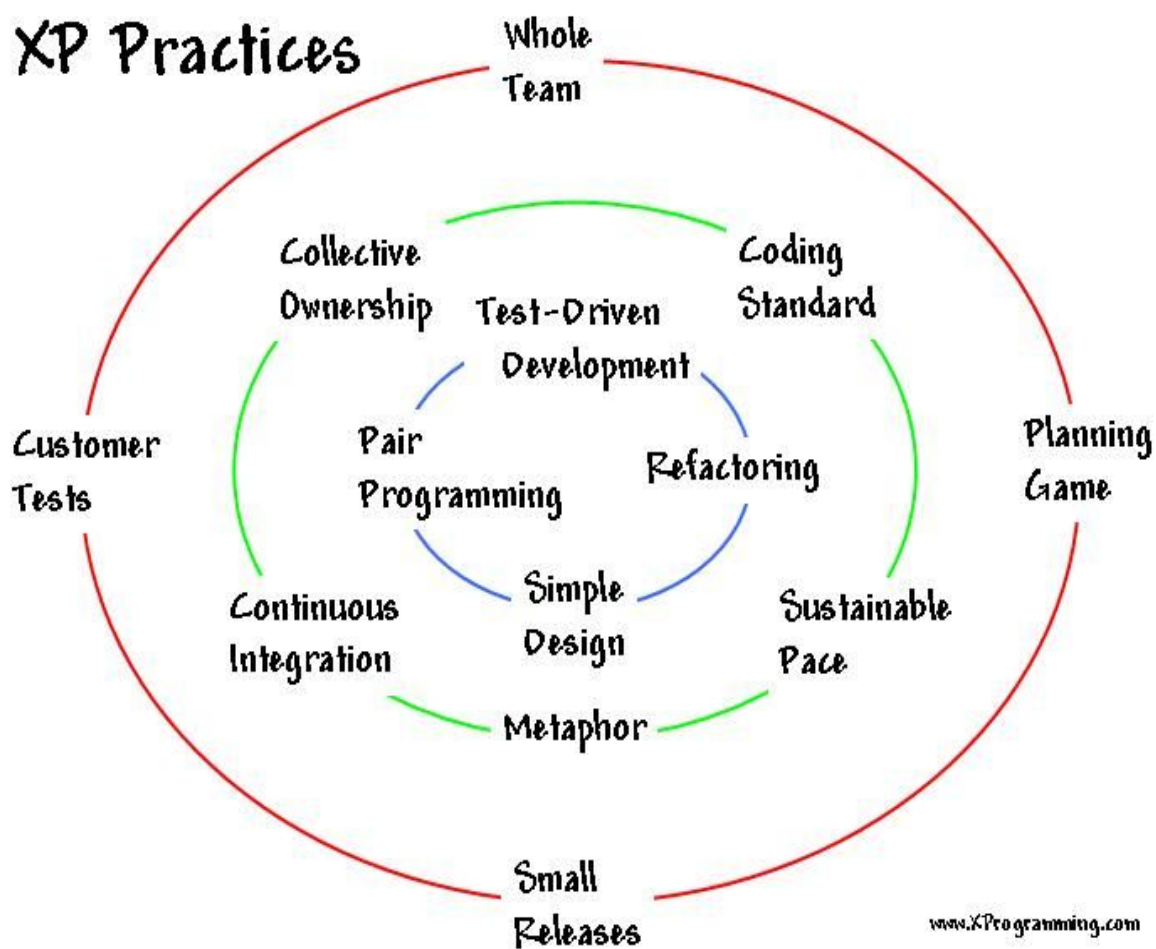
<!-- //Begin my fantastic javascript code
function helloworld(param1, param2) {
[...]}
// End -->
</script>
';

```

7. Work method & design methodology

Software is developed according to a certain way of working which we call a methodology. A methodology can also be regarded as a certain philosophy or set of ideas about software development. Some methodologies are very strict — which can be necessary e.g. in development of software for space shuttles — others are very simple and flexible. Dokeos is developed according to the *Extreme Programming* (XP) methodology (for an introduction see [What is XP](#)). This methodology is simple and flexible.

Furthermore, Dokeos is a free software program. Free software and Open Source software works so well because it encourages everyone to participate. In order to encourage this, you should communicate often; and release modifications as soon as possible: bugfixes, new pieces of code, a new tool, a piece of documentation... Release even when something is not complete yet: **release early, release often**. The user community will help you fix the problems. You will in time create a perfect piece of work, not by trying to do it all by yourself, but by a community that perfects code. This approach also eliminates bugs quicker than other approaches. You can read more about this approach in Eric S. Raymond's famous work, *The Cathedral and the Bazaar* ([here](#) and [here](#)).



8. Communication between developers

The cooperation between developers can make or break the development of software. Communication, an important element of cooperation, is more difficult when the developers are geographically apart. A good cooperation requires a good model of communication.

In free software, and extreme programming, no one owns the code, and everyone has to accept that his or her code will be changed and criticized. Dokeos has a project at sourceforge, we use this for CVS. CVS is a system to store code files, with version history, and is crucial to good cooperation. Every developer gets read access to the CVS, and a smaller “kernel” of developers also has write access.

How will developers communicate? The current developers have agreed on using a web forum as means of communication. A web forum has many advantages: it's easily reachable, all information is neatly organized and archived, and people can discuss developer – and other – issues as extensive as they want. Concentration of all the info on one medium has the advantage of not causing the problems that would appear when using for example a forum, a newsgroup and email: some people only look in one place, or just not often enough in the other places...

Communication to the outside world can happen using many different forms, of which the website and forum are usually the preferred form:

- A [website](#), with a “changelog” text describing the most recent changes
- A forum (see the [Dokeos forum](#))
- Email: Dokeos info <info@dokeos.net>

9. Software development in general

There are a few very important software development concepts. These have often been established from years of practice in which key strategies that have succeeded or failed are evaluated. They can help you write better software.

Information hiding

Encapsulation or information hiding is perhaps the most important concept for software development. Object oriented languages are best for this, but it is possible in other languages too. The idea is that you try to hide things that should not be seen from outside. For example, local variables in a function cannot be seen outside the function. Class variables (also called members) cannot be seen outside objects of that class.

Layered Design

Code should be designed into several layers. In typical standalone applications you often see a three layered design: one layer each for code related to database, the user interface and the code that does the real work, the “business” layer. This helps for improving and

maintaining the code. If you want to change the way the program looks, you have to change nothing in the database layer, if you change some calculations, you don't have to read the gui code. Dokeos is also divided into three layers: the kernel code which forms the heart of Dokeos ; the tools code; and the – very young – API layer which lays between tools ad kernel and provides a clean way for tools to talk to the kernel code.

When to apply

Software development techniques should be “a tool, not a hammer”. Use them when they apply to a certain situation and you can use them for an advantage. Don't use them just for the sake of using them. Apart from the philosophy aspect, which is very interesting, software development is in the end about one thing only – software that works. To quote Steve Jobs, “Real artists ship”!

10. User Interface Design

Just a few short rules:

- **A user interface is well-designed when the program behaves exactly how the user thought it would.**
- The more options, the more complex an interface becomes.
- The program should look and feel elegant, while providing all the tools to get the job done.

In a standalone application, you can put often used options in a toolbar and put all the rest in the menus. However, most web applications don't have a menu bar, so thinking about which features are necessary becomes very important.

Every time you provide an option, you're asking the user to make a decision. That means they will have to think about something and decide about it. It's not necessarily a *bad* thing, but, in general, you should always try to minimize the number of decisions that people have to make.

This doesn't mean eliminate *all* choice. There are enough choices that users will have to make anyway: the way their document will look, the way their web site will behave, or anything else that is integral to the work that the user is doing.

— Joel Spolsky, [User Interface Design for Programmers](#), Chapter 3, 2000

11. User Input

Ultimately, the users decide in which direction Dokeos will go. This does not of course exclude suggestions and views of the developers, but software is made for the users, not the programmers. (But in free software, it is often the case that the users are also the programmers). **The ultimate goal of an elearning environment such as Dokeos is to help people learn better.**

12. Promoting and spreading Dokeos

Dokeos is a free software elearning and course management platform. Spread the word: Dokeos is good, getting better, easy to use and adaptable to your own needs. Whatever your reasons may be, it's a good idea to make sure the user base of Dokeos is as broad as possible. The more users, the more feedback the developers get. Bugs, user friendliness, features and problems... everyone's contribution counts. The more programmers, the more stable and thorough the development and maintenance process is. Everyone can freely look at the code, modify and redistribute it.

Open source development is superior to old-school closed development.

One of the great advantages of free software is that it is capable of helping people cooperate to realize a goal. If you think Dokeos is—or can be—a good tool, do not hesitate to inform educational institutions, elearning managers, software developers... of our existence, and refer them to the [website](#).

13. Relevant links

General

[PHP.net](#) and the [PHP manual](#) (<http://www.php.net/docs.php>)

The [Pear](#) collection of reusable PHP components (<http://pear.php.net/>)

Free software & Opensource

[GNU.org](#)

[The Cathedral and the Bazaar](#) (<http://catb.org/~esr/writings/cathedral-bazaar/>)

Dokeos

[Dokeos.com](#)

[Forum](#) (<http://www.dokeos.net/forum/>)

[SourceForge](#) (<http://www.sourceforge.net/projects/dokeos/>)

XP introduction

[What is XP](#)

Design Patterns

[Portland Pattern Repository](#)

[Patterns FAQ](#)

Christopher Alexander is an architect (of cities and buildings) who invented patterns. See [Notes on Christopher Alexander](#).

14. GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language. A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format

whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work. In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects. You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document 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.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation 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. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

[How to use this License for your documents](#)

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:
Copyright (c) YEAR YOUR NAME.

```
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
  Texts. A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
  Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
If you have Invariant Sections without Cover Texts, or some other combination of the three,
merge those two alternatives to suit the situation.
```

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

15. Edit this manual

If you want to edit and improve this manual, ask an OpenOffice or a Ms-Word format copy of it :

info@dokeos.com
Dokeos
181 rue Royale
B-1210 Brussels
Belgium, Europe
Tel. 0032 2 227 00 26