The GNU Enterprise Application Server

A Whitepaper
Edition 0.0.11, 2004-02-24

Reinhard Müller

Table of Contents

1 Introduction

1.1 Purpose

In 2-tier systems, the application logic lies either in the front end or in the database (via triggers and stored procedures). The main purpose of the Application Server is to pull the application logic out of both of them and serve as a middle layer that abstracts the logic (called *business rules*) from the user interface as well as from the database backend.

1.2 Additional goals

Apart from (of course) fulfilling the purpose, we have defined several additional, ethical as well as technical, goals. The following list is sorted by priority:

Freedom GNUe Appserver must be GPL and must be built with truly free tools.

Stability GNUe Appserver must be reasonably stable. For a mission-critical application in a business environment, reasonably stable means very stable.

Security GNUe Appserver must be reasonably secure.

Maintainability
The code base of GNUe Appserver itself must be and remain maintainable by the development team, and the code must be clear enough to allow interested programmers to adapt it, fix bugs, or even take over maintenance of a part of Appserver.

Configurability
GNUe Appserver must be configurable and reconfigurable dynamically, centrally, without programming skills, without downtime, and in separate "layers" for various levels of specification.

Performance
GNUe Appserver must perform reasonably well with large quantities of users and/or data.

Database Independence
GNUe Appserver must be able to use a number of database systems as backend.

Portability
GNUe Appserver must run on multiple operating systems and architectures.

Communication Independence
GNUe Appserver must be able to use a number of communication means to communicate with the front end (CORBA, XML-RPC...).

Language Independence
GNUe Appserver must be able to deal with business methods written in different languages.

2 Features

(Note that not all features are yet implemented)

2.1 Business Objects

GNUe Appserver allows definition of data entities (for example, name and address of a customer) and of program code to perform on such entities (for example, how to build the address line from country, zipcode, and city).

The combination of a data entity with all code functions that can be performed on the entity is called a *business object*.

Appserver lets the user define *classes* of business objects. The class definitions describe both the data elements (called *properties*) and the available functions (called *procedures*) of the business object.

Procedures may also be bound to a specific event, that is, they are called automatically as soon as the event happens. These procedures are called *bound procedures*.

2.2 Properties

GNUe Appserver will provide the following property types:

Basic property
: contain most of the actual information, in the basic property types *string*, *number*, *date*, *time*, *datetime*, and *boolean*. Examples could contain customer name, item price, invoice date, and invoice payment status.

Compound properties
: are a means for combining properties that logically belong together and appear repeatedly in the same combination. Compound properties can be built from properties of any type, not only of basic properties. An example could be a monetary value consisting of the amount and the currency.

Reference properties
: point to another object and declare a relation between two objects. The value of a reference property is another object. Examples include the customer of an invoice or the preferred vendor of an item.

List properties
: point to lists of objects and declare a relation between these objects. The value of a list properties is a list of objects of a defined class, where all objects in the list will be of the same class. Lists are actually nothing more than a shortcut for frequently used filters. Examples include all line items of an invoice, or all line items of all invoices containing a specific stock item, or all invoices of a customer.

Calculated properties
: contain information that is generated by Appserver out of other properties. Calculated properties are generally read-only. An example could be the total value of an invoice item (calculated from price * pieces) or the total value of an invoice (calculated from all total item values).

2.3 Procedures

A procedure is code performed on an object. Procedures can have parameters of any attribute type (string, number, reference, etc.). Every method is passed a parameter, *self*, that identifies the object to operate on.

2.4 Modules

Modules define namespaces for classes, properties, and procedures. When module A defines a class with various properties and procedures, module B can extend the class with new properties. Another module, C, can independently extend the same class, without fear of conflicting with module B. All modules have their own namespace.

2.5 Qualified class, property and procedure names

Class, property, and procedure names can be preceded by a module name to override the current module context. In this case, the module name is separated by an underscore (_). This results in the rule that neither a module name nor a class, property or procedure name may contain an underscore.

Property and procedure names are separated from the object references by a dot (.).

Example:

Module "cust" defines a class "customer". Module "sales" defines a class "invoiceHead" and a class "invoiceItem". Module "base" defines a class "item".

Then, `cust_customer` is the fully referenced class name for the customer class, and `sales_invoiceHead` is the fully referenced class name for the invoiceHead class.

Now, let module "cust" define the properties "name" and "address" for the customer class, where address is a compound property consisting of "street" and "city".

The following are now valid property references of a customer object:

`name` or `cust_name` is a base property.

`address` or `cust_address` is a compound property.

`address.street` or `cust_address.street` is a compound member property.

Now, the module "sales" extends the customer class by a property "lastInvoice" which is a reference to an invoiceHead object.

`sales_lastInvoice` is a reference containing an invoice_head object.

If module "sales" defines the attribute "number" and "items" in "invoiceHeader", then

`sales_lastInvoice.number` is the number of the last invoice of the customer. We could call that an *indirect property*.

`sales_lastInvoice.items` is an indirect list property.

And if yet another module, "acct", extends the invoice_header class by a property "paid", then

`sales_lastInvoice.acct_paid` could be a boolean property that tells you whether the customer has paid his last invoice or not, and you would access it just as easy as the "name" attribute. However, this attribute would only be available if all three modules "cust", "sales", and "acct" are installed.

2.6 Bound Procedures

Bound procedures are procedures that are automatically called upon occurance of defined *events*; for example, on every change of a specific property or before a commit of a changed object.

Bound procedures are always procedures of the object where the event occurs. Because every module can extend any class with a procedure of arbitary name, calling of bound procedures could be automated by method name.

Example:

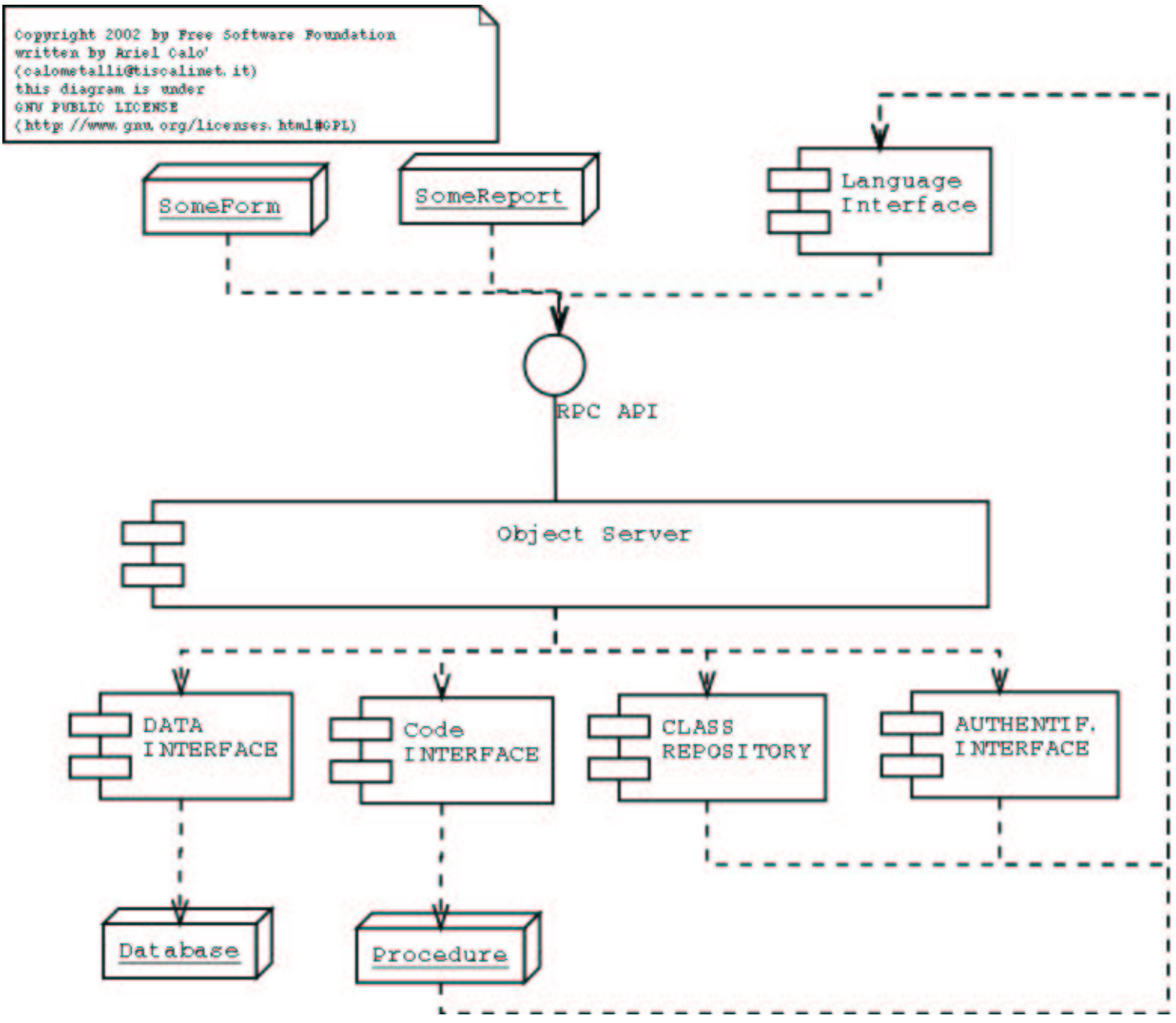
The cust module, which defines the customer class, defines a procedure "OnChange-Name", whose fully qualified name is of course `cust_OnChangeName`.

The sales module could extend the customer class by a procedure and also call this it "OnChangeName", because the fully qualified name of this procedure will be `sales_OnChangeName` and therefore different from the other procedure.

If the property "Name" is changed in a customer object, Appserver would call both procedures because both are named "OnChangeName". The order of the calls would be unpredictable.

3 Architecture

3.1 Overview



Data Interface:
: abstracts database access from a specific database API and from SQL, and caches data to minimize database traffic.

Code Interface:
: abstracts calls to methods from the language and the specific API.

Class Repository:
: holds and provides the definition of classes.

Authentication Interface:
: provides a way to authenticate a user and define his access rights.

Object Server:
: translates all requests to the business objects into appropriate database transactions and method calls, by using the other building blocks.

Remote Protocol Interface:
: provides the Appserver API to a remote client via various RPC mechanisms.

Language Interface:
: provides easy access to business objects for procedure code and for internal use by Appserver

3.2 Theory of Operation

For easy understanding, here is the basic way the Application Server will provide data:

1. The client requests an object from the app server through the Remote Protocol Interface.
2. The Object Server checks if the requested class name is valid by looking up the class definition in the Object Repository.
3. If the class name is valid, then the Object Server uses the Authentication Interface to check whether the connected user may access this class.
4. If both are OK, the Object Server translates the class name into a table name. This will include adding a prefix to enable different application modules to have their own namespace.
5. Then the Object Server passes the database request to the Data Interface to actually get the data.

3.3 Data Interface

The Data Interface provides an API that allows creation and extension of tables, reading of data and updating, adding and deleting of records in a table, all without SQL.

The Data Interface operates strictly on a table/row interface and doesn't know anything about objects, properties and procedures.

The Data Interface caches data read from the database backend as well as data modified through the RPC API. This does not only reduce the traffic between appserver and the database backend. Because the cache is shared between all cursors of a connection, updates done with cursor A become visible in cursor B immediately without re-issuing a query.

The Data Interface used in GNUe Appserver uses GNUe Common's datasources system to communicate with the database.

3.4 Code Interface

The Code Adapter abstracts the calls to business methods written in the different languages.

Python is the only language supported for the near future.

The Code Adapter, or at least parts of it, will reside in GNUe Common.

3.5 Class Repository

The Class Repository holds all the business object definitions: what properties the object consists of, what procedures exist, which bound procedures should be called on what event, and so on.

The class definitions are stored in the database and are accessible like normal business objects. Consequently, the Class Repository accesses the definitions via the Python Language Interface.

3.6 Authentication Interface

The Authentication Interface will provide a means to authenticate a user on connection to the Appserver, and to check access rights for an already connected user.

For the sake of flexibility and independence from operating system and database backend, access rights per user will probably be stored in the database and be accessible via system business objects.

3.7 Object Server

This is the central part of the Application Server.

The Object Server uses the Data Interface and the Code Interface to fulfill requests directed at business objects, after it has checked the validity of the request against the Class Repository and the Authentication Interface.

Security implementation will also select or reject based on the users authorizations to any regular query.

Example: if the division president uses a form to request all sales orders, Appserver will query the database and return only the object data that represents the divisions sales orders.

Form level (view) security will not be enforced by Appserver.

The Object Server will also provide object transparency. Meaning that there will not necessarily be a direct relationship between business objects and tables.

While accessing data from the database via the Data Interface, the Object Server will also automagically check for defined bound procedures, and call them via the Code Interface.

3.8 Remote Protocol Interface

The Remote Protocol Interface is used to export the functionality of the Application Server over the net, using a variety of RPC mechanisms.

3.9 Language Interface

The Language Interface makes business objects easily accessible for code written in supported languages. Each language has it's own Language Interface.

For example, the Python Language Interface (which is the first and currently only one implemented) lets you access business objects as if they are normal Python objects. Procedures written in Python use this interface to access business objects in a rather straightforward way. Other parts of the Application Server also use the Language Interface to access system objects like the ones defining classes or access rights.

Appendix A GNU Free Documentation License

Version 1.1, March 2000

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written 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. Secondarily, 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 that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

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. (For example, 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.

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 "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and 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 has been designed to thwart or discourage subsequent modification by readers is not Transparent. 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 designed for human modification. Opaque formats include PostScript, PDF, 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 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.

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 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 publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. 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 less than five).

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", and 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. In any section entitled "Acknowledgments" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgments 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 as "Endorsements" or to conflict in title with any Invariant Section.

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.

   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 "Acknowledgments", 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, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they 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 quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around 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 provided that

you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

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.

A.1 ADDENDUM: 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 1.1
> or any later version published by the Free Software Foundation;
> with the Invariant Sections being *list their titles*, with the
> Front-Cover Texts being *list*, and with the Back-Cover Texts being *list*.
> A copy of the license is included in the section entitled ‘‘GNU
> Free Documentation License’’.

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being *list*"; likewise for Back-Cover Texts.

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.