

jAPS2.0 - Plugin Pattern

Do It The jAPS Way

**Eugenio Santoboni, AgileTec s.r.l. <e.santoboni@agiletec.it>
William Ghelfi, Tzente s.r.l. <w.ghelfi@tzente.it>**

jAPS2.0 - Plugin Pattern

by Eugenio Santoboni and William Ghelfi

Publication date 2010-03-25

Copyright © 2010 Tzente s.r.l.

Abstract

This document covers the characteristics, structure and components of a jAPS-Plugin. It also provides the guidelines which Java developers must follow in order to build new plugins for the jAPS2.0 framework.

Legal Notice

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 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 Appendix entitled "GNU Free Documentation License".

The copyright holders make no representation about the suitability of this Document for any purpose. It is provided "as is" without expressed or implied warranty. If you modify this Document in any way, identify your resulting Document as a variant of this Document.

Table of Contents

1. Scope of the document	1
Introduction	1
Target audience	1
Prerequisites	1
Resources	1
2. What is a jAPS-Plugin	2
3. Structure of the package	3
Structure of the package	3
The directory layout of a Plugin	3
Document: Readme	5
Directory: src	5
Directory: doc	7
4. Naming convention	9
Naming convention	9
Name and Id	9
Sources	9
Name of the jar	9
Directory under WEB-INF/plugins	9
Static resources	9
Beans (Spring Objects) in the core of the Plugin	9
Beans (Spring Objects) of the Actions (Struts2 Objects)	10
Tiles	10
i18n	10
Permissions	11
Bean for a sub-menu in the backend	11
Database tables	11
Configuration items (table sysconfig, database *Port)	11
A. GNU Free Documentation License	12

List of Examples

3.1. The Directory Layout of a typical Plugin (continue)	3
3.2. (continued) The Directory Layout of a typical Plugin	4
3.3. From a domain name to a package name	5
3.4. The proper name of a tld file for the public frontend	6
3.5. The proper name of a configuration file for Tiles	7
3.6. The proper name of a tld file for the administration area	7
3.7. Some proper subdirectories	7
3.8. Plugin jpuserprofile: proper path for the CSS of the public frontend	7
4.1. Name and id of a Plugin	9
4.2. From a domain name to a package name	9
4.3. Unit tests: From a domain name to a package name	9
4.4. Naming convention for a Struts2 Action	10
4.5. A proper path for a Struts2 Action	10
4.6. A proper package id for a Struts2 Action	10
4.7. Some sample labels for the frontend	10
4.8. Some sample labels for the backend	10

Chapter 1. Scope of the document

Introduction

This document covers the characteristics, structure and components of a *jAPS-Plugin*. It also provides the guidelines that Java developers must follow in order to create new plugins for the jAPS2.0 framework.

Target audience

This document is for developers aiming to build new plugins for jAPS2.0.

Prerequisites

In order to take the maximum advantage of the present document, it is necessary to have basic knowledge about: the Java platform, the Eclipse IDE, the Apache Tomcat servlet engine, the PostgreSQL DBMS.

Resources

Additional informations may be obtained through the following two mailing-lists:

- <japs-devs@lists.sourceforge.net>, focused on developers
- <japs-users@lists.sourceforge.net>, focused on final users

Is it also possible to refer to the documentation found in the:

jAPS 2.0 Project - Developer Website [<http://dev.japsportal.org/>]

Chapter 2. What is a jAPS-Plugin

A jAPS-Plugin is a *module* which can add new features to the jAPS2.0 framework or modify some existing ones.

The jAPS-Plugin is commonly also known as *service*, thus the principles guiding the process of writing a Plugin are basically the same guiding the development of a new service as described in the document *jAPS2.0 - Service Creation and Integration Pattern*.

The Plugins are classified as follows:

- PurePlugins (a.k.a. *so-called Plugins*)
- Modifications

A PurePlugin provides new features without any modification to the existing ones.

The PurePlugins are what we call *ideal plugins* because their impact on existing features tends to null, the use of two or more PurePlugins at the same time in the same instance of jAPS2.0 not causing any conflicts just because each and every module works well regardless of any other module.

As opposite, the Modifications are the ones which either expand, substitute, or modify (from slightly to deeply) the behaviour of the core elements of jAPS2.0.

Because of their massive impact on the system, they might come with specific install procedures and you may need to be very careful when installing a Modification in a system with other Plugins (both PurePlugins and Modifications). That because is not rare to have conflicts under such conditions.

In order to reduce to the minimum the risk of conflicts, the development of a Modification has to be done trying to limit the elements which interact with or modify the core; this fundamental achievement can be reached taking advantage of the elements given by the system and by the inner frameworks (like Spring and Struts2), i.e. the *Event Notification* service, the AOP, the Interceptors architecture which comes with Struts2, and many others.

Chapter 3. Structure of the package

Structure of the package

A Plugin has to be released in a package containing 3 fundamental elements:

1. Document: Readme
2. Directory: `src`
3. Directory: `doc`

The directory layout of a Plugin

The directory layout resembles the Standard Directory Layout from the Apache Maven Project [<http://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html>].

Example 3.1. The Directory Layout of a typical Plugin (continue)

```
wiz@id:~/jAPS2-Plugin-jpuserprofile-1.0$ tree -dx
```

```
.
|-- doc
|   |-- jAPS2.0-Plugin_UserProfile
|       |-- fo
|           |-- img
|       |-- html
|           |-- css
|           |-- img
|       |-- pdf
|       |-- ps
```

Example 3.2. (continued) The Directory Layout of a typical Plugin

```

|-- src
|-- java
|   |-- main
|   |   |-- com
|   |   |   |-- agiletec
|   |   |   |   |-- plugins
|   |   |   |   |   |-- jpuserprofile
|   |   |   |   |   |   |-- aps
|   |   |   |   |   |   |   |-- externalframework
|   |   |   |   |   |   |   |   |-- common
|   |   |   |   |   |   |   |-- system
|   |   |   |   |   |   |   |   |-- services
|   |   |   |   |   |   |   |   |   |-- profile
|   |   |   |   |   |   |   |   |   |-- model
|   |   |   |   |   |-- apsadmin
|   |   |   |   |   |   |-- common
|   |   |   |   |   |   |   |-- attribute
|   |   |   |   |   |   |   |   |-- action
|   |   |   |   |   |   |   |   |-- list
|   |   |   |   |-- user
|   |-- test
|   |   |-- test
|   |   |   |-- com
|   |   |   |   |-- agiletec
|   |   |   |   |   |-- plugins
|   |   |   |   |   |   |-- jpuserprofile
|   |   |   |   |   |   |   |-- aps
|   |   |   |   |   |   |   |   |-- externalframework
|   |   |   |   |   |   |   |   |   |-- common
|   |   |   |   |   |   |   |-- system
|   |   |   |   |   |   |   |   |-- services
|   |   |   |   |-- apsadmin
|   |   |   |   |   |-- common
|-- sql
|-- webapp
|   |-- WEB-INF
|   |   |-- lib
|   |   |   |-- plugins
|   |   |   |   |-- jpuserprofile
|   |   |   |   |   |-- aps
|   |   |   |   |   |   |-- jsp
|   |   |   |   |   |   |   |-- externalFramework
|   |   |   |   |-- apsadmin
|   |   |   |   |   |-- conf
|   |   |   |   |   |   |-- jsp
|   |   |   |   |   |   |   |-- common
|   |   |   |   |   |   |   |   |-- template
|   |   |   |   |   |   |   |   |-- extraresources
|   |   |   |-- conf
|   |   |   |   |-- managers
|   |-- resources
|   |   |-- plugins
|   |   |   |-- jpuserprofile

```

62 directories

Document: Readme

The Readme is a short document (max 40 rows of 80 columns) explaining the features of the Plugin and the version of jAPS under which the Plugin is declared to have been developed and tested. The scope of the document is to give an immediate review of the Plugin.

Directory: src

This directory contains all the components necessary for integrating the Plugin in a jAPS2.0 based project using the Eclipse IDE (version 3.4 *Ganymede*) or later.

The directory contains:

Java sources

The Java classes of the Plugin **must** be written following the coding standard of jAPS2.0 framework. They must be gathered under the package `<R_DOMAIN>.plugins.<PLUGIN_ID>` where `<R_DOMAIN>` is the reversal of the domain name of the developer (company, association, web site, person...) of the Plugin.

Example 3.3. From a domain name to a package name

Company: MyCompany.com, package: `com.mycompany.*`

Every Class **must** come with proper Javadoc in english for an easy understanding of the code and of the application flow. Every *interface* and *concrete class* must have its Javadoc in the head part, and for every *method* and *public or private variable*. You may also want to write some Javadoc for complex or not-so-readable *private methods*.

If the Plugin adds elements to the administration GUI of jAPS2.0, in the root of java sources is mandatory to have a file named `<PLUGIN_ID>-japs-struts-plugin.xml` containing the references to file for configuration of defined actions.

The `...<CODICE_PLUGIN>/src` directory contains `aps` and `apsadmin`. The `apsadmin` always contains the `global-messages_<LOCALE_CODE>.properties`: the file(s) contains the global labels of the menu items and eventually of the interfaces.

Important

The property files must always contain at least the two following voices:

- `<PLUGIN_CODE>.code=<PLUGIN_CODE>`
- `<PLUGIN_CODE>.name=<plugin description>`

The libraries

There is a `jAPSPlugin-<PLUGIN_ID>-<VERSION>.jar` containing all the classes and packages described before, *plus* the optional libraries which may be

necessary to the Plugin and are not present in the standard package of the jAPS2.0 framework.

The unit tests

Each feature of the Plugin **must** have a comprehensive set of unit tests; the test classes must be into a package named as follows: `test.<R_DOMAIN>.plugins.<PLUGIN_ID>`.

The test environment, in the initialization phase, must load the whole context from the core of jAPS2.0 and the beans of the Plugin optionally accompanied by the any other necessary bean.

The `jsp` files and the configurations

The name of this directory is the code of the plugin and it is placed into `/WEB-INF/plugins/`.

The directory contains the subdirectories `aps`, `apsadmin`, `conf`. The internal organization of these subdirectories is a perfect copy of the ones placed into the directory `WEB-INF` of the core of jAPS2.0.

The subdirectory `aps` contains the files needed by the public frontend (as opposite to the backend administration area) part of the Plugin.

The directory contains:

- The subdirectory `jsp/showlets`, for any showlet coming with the Plugin
- The subdirectory `jsp/models`, for any *Page Model* coming with the Plugin
- The subdirectory `jsp/externalFramework` (optional), for the `jsp` files concerning any feature wich uses the *External Framework* special showlet.
- The subdirectory `tld`, for the definitions of any custom tag.

Example 3.4. The proper name of a `tld` file for the public frontend

`<PLUGIN_ID>-aps-core.tld`

The subdirectory `apsadmin` contains the files needed by the backed administration area (as opposite to the public frontend) part of the Plugin.

The directory contains:

- The subdirectory `jsp`, with a subdirectory for each component feature and all the `jsp` files for any needed administration GUI that has to be added to the main one
- The subdirectory `conf`, with a subdirectory for each component feature and all the configuration files for any needed *Action* class (Spring Beans) and the configuration file for *Tiles*

Example 3.5. The proper name of a configuration file for *Tiles*

```
<PLUGIN_ID>-tiles.xml
```

- The subdirectory `tld`, for the definitions of any custom tag.

Example 3.6. The proper name of a `tld` file for the administration area

```
<PLUGIN_ID>-apsadmin-core.tld
```

The subdirectory `conf` contains the configuration files of all the beans (Spring Object) used by the Plugin.

If the Plugin needs new `DataSources` or has its own paths used by any internal process, in the directory `conf` there must be two files which resemble the ones found in the standard package of the jAPS2.0 framework:

- `systemParams.properties`: containing all the configurations.
- `systemConfig.xml`: configuration file for Spring containing the placeholders for the properties defined in the other file and the definitions for the Bean `DataSources` (Spring Objects).

The optional directory `extras` contains resources or subdirectories needed by the Plugin which cannot find any other place in the filesystem as described before.

The needed static resources (CSS, images, Javascript...)

The files are placed into the directory `/resources/plugins/<PLUGIN_ID>`.

It resembles the structure, naming convention and organization of the main directory `resources`; thus, it may have two main subdirectories: `administration` and `static` with their own subdirectories.

Example 3.7. Some proper subdirectories

```
static/css,static/img,administration/js
```

Example 3.8. Plugin `jpuserprofile`: proper path for the CSS of the public frontend

```
/resources/plugins/jpuserprofile/static/css/
```

Directory: `doc`

The directory contains all the documentation for the Plugin.

There **must** be a document presenting the technical and functional characteristics, written in english, and containing the following informations:

- The complete description of all the features and services provided by the Plugin

- The declaration of the version of the Plugin, and of the version of jAPS2.0 declared compatible (ie: 2.0, or > 2.0.2, or >= 2.0, and so on). Any dependency with other Plugins (and their versions) must also be declared
- The installation procedure, optionally accompanied by any needed SQL script
- The instructions for installing the Plugin in a development environment based on Eclipse IDE version 3.4 Ganymede. You must also provide the instructions for properly configure the unit tests
- The flavour of the Plugin (*PurePlugin* or *Modification*). If the Plugin is a Modification, you must declare and give detailed explanation of each and every feature of the core of jAPS2.0 which is subject of changes; you must also declare each and every Bean (Spring Object) overwritten by the Modification and the modified GUIs.

Chapter 4. Naming convention

Naming convention

The following is the naming convention to be adopted during the development of a Plugin.

Name and Id

The id of the Plugin is unique, it must have the prefix `jp`, all lowercase, and it must contain alphabetic characters only (no numbers, special characters, nor spaces).

The id is different from the name because the name allows special characters and spaces.

Example 4.1. Name and id of a Plugin

User Profile is the name of the Plugin whose id is `jpuserprofile`.

Sources

The Java classes of the Plugin **must** be written following the coding standard of jAPS2.0 framework. They must be gathered under the package `<R_DOMAIN>.plugins.<PLUGIN_ID>` where `<R_DOMAIN>` is the reversal of the domain name of the developer (company, association, web site, person...) of the Plugin.

Example 4.2. From a domain name to a package name

Company: MyCompany.com, package: `com.mycompany.plugins.<PLUGIN_ID>`

Example 4.3. Unit tests: From a domain name to a package name

Company: MyCompany.com, package: `test.com.mycompany.plugins.<PLUGIN_ID>`

This syntax resembles the one defined in the [Java Language Specification [http://java.sun.com/docs/books/jls/third_edition/html/packages.html#7.7] by Sun Microsystems].

Name of the jar

The name of the jar must be: `jAPSPlugin-<PLUGIN_ID>-<VERSION>.jar`.

Directory under WEB-INF/plugins

The name of the directory containing all the `jsp` and configuration files must be equal to the Plugin id: `WEB-INF/plugins/<PLUGIN_ID>`.

Static resources

See the previous chapters.

Beans (Spring Objects) in the core of the Plugin

In order to avoid any overwriting of names used in the main core or by other Plugins, the ids of the Beans must be prefixed with the plugin id. In the particular case of a Manager, the id must be `<PLUGIN_ID><SERVICE_NAME>Manager`.

Moreover, if a Plugin overwrites a Bean of the main core or of another dependent Plugin the id of that Bean must be the same of the one overwritten. Detailed informations **must** also be provided in the technical documents.

Beans (Spring Objects) of the Actions (Struts2 Objects)

See the previous paragraph. Besides, also the paths of all the *Actions* must avoid overlappings with the paths used in the main core or in other Plugins.

Here are the proper syntaxes for definitions of the *Actions*:

- The paths in the namespaces must have the id of the Plugin after the string used for the mapping of the Struts2 filter.

Example 4.4. Naming convention for a Struts2 Action

```
/do/<PLUGIN_ID>/<SUB_PATH>
```

Example 4.5. A proper path for a Struts2 Action

```
/do/jpuserprofile/CurrentUser/edit.action
```

- The ids of the packages must be in the form <PLUGIN_ID>_do/<PLUGIN_ID>/<SUB_PATH>

Example 4.6. A proper package id for a Struts2 Action

```
jpuserprofile_do/jpuserprofile/CurrentUser
```

Tiles

In order to avoid any overwriting of names used in the main core or by other Plugins, the ids of the *Tiles Definitions* must use the plugin id as follows: admin.<PLUGIN_ID>.<FEATURE>.<NAME_OF_THE_VIEW>

i18n

If the Plugin has its own GUIs for the public frontend, you must provide the proper labels (which are stored in the database table `localstrings` and are outputted through the `wp:i18n` custom tag). I.e. labels for forms, subtitles, notes, and so on.

Example 4.7. Some sample labels for the frontend

- `jpuserprofile_SEARCH_FORM` = Search
- `jpuserprofile_ADDRESS_COUNTRY` = Country

The same is for any GUI in the administration backend. In this case you must provide the files `package_<LANG_CODE>.properties` as described in the document *JAPS2.0 - Service Creation and Integration Pattern*.. As always, both frontend and backend keys must avoid any overlapping of the ones in the main core or other Plugins. This can be achieved properly using the id of the Plugin as a prefix.

Example 4.8. Some sample labels for the backend

- `jpuserprofile.mylabel` = My really simple label
- `jpuserprofile.myarea.myotherlabel` = This is a note, just like the one up there

Furthermore, for the back-end interfaces, each plugin must have a property file named in the package `<R_DOMAIN>.plugins.<PLUGIN_ID>.apsadmin`. This file must include the base label plugin (labels of code `<PLUGIN_ID>.code` and `<PLUGIN_ID>.name`) as well as any other label for the general menu *Plugins*.

Permissions

If the Plugin introduces new permissions, the id of each permission must be prefixed by the id of the Plugin followed by an underscore (`_` character).

Bean for a sub-menu in the backend

The id of the Bean for a *sub-menu* of the *Plugins* menu item, must be as follows:
`<PLUGIN_ID>SubMenu`.

Database tables

The names of the tables in either the `*Port` or `*Serv` database, must be prefixed by the id of the Plugin followed by an underscore (`_` character).

Configuration items (table `sysconfig`, database `*Port`)

The names of the configuration items must be prefixed by the id of the Plugin followed by an underscore (`_` character).

Appendix A. GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. [<http://www.fsf.org/>]

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.

The “publisher” means any person or entity that distributes copies of the Document to the public.

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 under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

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 Copyleft [<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. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

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 © YEAR YOUR NAME

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 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.