XSL Formatting Objects

XSL Formatting Objects

http://en.wikipedia.org/wiki/XSL_Formatting_Objects

This Book Is Generated By WikiType

using

RenderX DiType, XML to PDF XSL-FO Formatter

29 September 2008

Table of Contents

XSL Formatting Objects

See Picture License Information Here

This article does not cite any references or sources.
Please help improve this article by adding citations to reliable sources. Unverifiable material may be challenged and removed. *(February 2008)*

XSL Formatting Objects

Filename extension	.xml
Internet media type	application/xml, text/xml (deprecated)
Uniform Type Identifier	public.xml
Developed by	World Wide Web Consortium
Type of format	Markup language
Contained by	XML
Standard(s)	1.1

XSL Formatting Objects, or **XSL-FO**, is a markup language for XML document formatting which is most often used to generate PDFs. XSL-FO is part of XSL, a set of W3C technologies designed for the transformation and formatting of XML data. The other parts of XSL are XSLT and XPath. As of December 12, 2006, the current version of XSL-FO is v1.1.

XSL-FO basics

Unlike the combination of HTML and CSS, XSL-FO is a unified presentational language. It has no semantic markup in the way it is meant in HTML. And, unlike CSS which modifies the default presentation of an external XML or HTML document, it stores all of the document's data within itself.

The general idea behind XSL-FO's use is that the user writes a document, not in FO, but in an XML language. XHTML, DocBook, and TEI are all possibilities, but it could be any XML language. Then, the user obtains an XSLT transform, either by writing one themselves or by finding one for the document type in question. This XSLT transform converts the XML into XSL-FO.

Once the XSL-FO document is generated, it is then passed to an application called an FO processor. FO processors convert the XSL-FO document into something that is readable, printable or both. The most common output of XSL-FO is a PDF file or as PS, but some FO processors can output to other formats like RTF files or even just a window in the user's GUI displaying the sequence of pages and their contents.

The XSLT language itself was originally conceived only for this purpose; it has obviously now been supplanted for more general XML transformations. This transformation step is taken so much for granted in XSL-FO that it is not uncommon for people to call the XSLT that turns XML into XSL-FO the actual XSL-FO document itself. Even tutorials on XSL-FO tend to be written with XSLT commands around the FO processing instructions.

The XSLT transformation step is exceptionally powerful. It allows for the automatic generation of a table of contents, linked references, an index, and various other possibilities.

An XSL-FO document is not like a PDF or a PostScript document. It does not definitively describe the layout of the text on various pages. Instead, it describes what the pages look like and where the various contents go. From there, an FO processor determines how to position the text within the boundaries described by the FO document. The XSL-FO specification even allows different FO processors to have varying responses with regard to the resultant generated pages.

For example, some FO processors can hyphenate words to minimize space when breaking a line, while others choose not to. Different processors may even use different hyphenation algorithms, ranging from very simple to more complex hyphenated algorithms that take into account whether the previous or next line also is hyphenated. These will change, in some borderline cases quite substantially, the layout of the various pages. There are other cases where the XSL-FO specification explicitly allows FO processors some degree of choice with regard to layout.

This differentiation between FO processors, creating inconsistent results between processors is often not a concern. This is because the general purpose behind XSL-FO is to generate paged, printed media. XSL-FO documents themselves are usually used as intermediaries, mostly to generate either PDF files or a printed document as the final form to be distributed. This is as opposed to how HTML is generated and distributed as a final form directly to the user. As such, a person wanting to generate a printed document only has to select the FO processor that fulfills

their needs, usually in the realm of layout quality and reduction of unnecessary whitespace, rather than having to test their XSL-FO document on multiple processors.

XSL-FO language concepts

The XSL-FO language was designed for paged media, in much the same way that HTML and CSS were designed for unpaged (or screen-based) media. As such, the concept of pages is an integral part of XSL-FO's structure, and FO gives the user significant power in dealing with how information is displayed on a page.

FO works best for what could be called "content-driven" design. This is the standard method of layout for books, articles, legal documents, and so forth. It involves a single flowing span of fairly contiguous text, with various repeating information built into the margins of a page. This is as opposed to "layout-driven" design, which is used in newspapers or magazines. If content in those documents does not fit in the required space, some of it is trimmed away until it does fit. XSL-FO does not easily handle the tight restrictions of magazine layout; indeed, in many cases, it lacks the ability to express some forms of said layout.

Despite the basic nature of the language's design, it is capable of a great deal of expressiveness. Tables, lists, side floats, and a variety of other features are available. These features are comparable to CSS's layout features, though some of those features are expected to be built by the XSLT.

XSL-FO document structure

XSL-FO documents are XML documents, but they do not have to conform to any DTD or schema. Instead, they conform to a syntax defined in the XSL-FO specification.

XSL-FO documents contain two required sections. The first section details a list of named page layouts. The second section is a list of document data, with markup, that uses the various page layouts to determine how the content fills the various pages.

Page layouts define the properties of the page. They can define the directions for the flow of text, so as to match the conventions for the language in question. They define the size of a page as well as the margins of that page. More importantly, they can define sequences of pages that allow for effects where the odd and even pages look different. For example, one can define

a page layout sequence that gives extra space to the inner margins for printing purposes; this allows more space to be given to the margin where the book will be bound.

The document data portion is broken up into a sequence of flows, where each flow is attached to a page layout. The flows contain a list of blocks which, in turn, each contain a list of text data, inline markup elements, or a combination of the two. Content may also be added to the margins of the document, for page numbers, chapter headings and the like.

Blocks and inline elements function in much the same way as for CSS, though some of the rules for padding and margins differ between FO and CSS. The direction, relative to the page orientation, for the progression of blocks and inlines can be fully specified, thus allowing FO documents to function under languages that are read different from English. The language of the FO specification, unlike that of CSS 2.1, uses direction-neutral terms like start and end rather than left and right when describing these directions.

XSL-FO's basic content markup is derived from CSS and its cascading rules. As such, many attributes in XSL-FO propagate into the child elements unless explicitly overridden.

Capabilities of XSL-FO v1.0

XSL-FO is capable of a great deal of textual layout functionality. In addition to the information as specified above, XSL-FO's language allows for the specification of the following.

Multiple columns

A page can be defined to have multiple columns. When this is the case, blocks flow from one column into the next by default. Individual blocks can be set to span all columns, creating a textual break in the page. The columns above this break will flow into each other, as will the columns below the break. But no text is allowed to flow from the above section to the below section.

Because of the nature of XSL-FO's page specification, multiple pages may actually have different numbers and widths of columns. As such, text can flow from a 3 column page to a 5 column page to a 1 column page quite easily.

All FO features work within the restrictions of a multi-column page.

Lists

An XSL-FO list is, essentially, two sets of blocks stacked side by side. An entry consists of a block on the "left", or start inline direction, and a block sequence on the "right", or end inline direction. The block on the left is conceptually what would be the number or bullet in a list. However, it could just as easily be a string of text, as one might see in a glossary entry. The block on the right works as expected. Both of these blocks can be block containers, or have multiple blocks in a single list entry.

Numbering of XSL-FO lists, when they are numbered, is expected to be done by the XSLT, or whatever other process, that generated the XSL-FO document. As such, number lists are to be explicitly numbered in XSL-FO.

Pagination controls

The user can specify Widow and Orphan for blocks or for the flow itself, and allow the attributes to cascade into child blocks. Additionally, blocks can be specified to be kept together on a single page. For example, an image block and the description of that image can be set to never be separated. The FO processor will do its best to adhere to these commands, even if it requires creating a great deal of empty space on a page.

Footnotes

The user can create footnotes that appear at the bottom of a page. The footnote is written, in the FO document, in the regular flow of text at the point where it is referenced. The reference is represented as an inline definition, though it is not required. The body is one or more blocks that are placed by the FO processor to the bottom of the page. The FO processor guarantees that wherever the reference is, the footnote cited by that reference will begin on the same page. This will be so even if it means creating extra empty space on a page.

Tables

A FO table functions much like an HTML/CSS table. The user specifies rows of data for each individual cell. The user can, also, specify some styling information for each column, such

as background color. Additionally, the user can specify the first row as a table header row, with its own separate styling information.

The FO processor can be told exactly how much space to give each column, or it can be told to auto-fit the text in the table.

Text orientation controls

FO has extensive controls for orienting blocks of text. One can, in the middle of a page, designate a block of text to be oriented in a different orientation. These oriented blocks can be used for languages in a different orientation from the rest of the document, or simply if one needs to orient the text for layout purposes. These blocks can contain virtually any kind of content, from tables to lists or even other blocks of reoriented text.

Miscellaneous

- Page number citations. A page that contains a special tag can be cited in text, and the FO processor will fill in the actual page number where this tag appears.
- Block borders, in a number of styles.
- Background colors and images.
- Font controls and weighting, as in CSS.
- Side floats.
- Miscellaneous Inline Elements

Capabilities of XSL-FO v1.1

Version 1.1 of XSL-FO adds a number of new features to version 1.0.

Multiple flows and flow mapping

XSL-FO 1.0 was fairly restrictive about what text was allowed to go in what areas of a page. Version 1.1 loosens these restrictions significantly, allowing flowing text to be mapped into multiple explicit regions on a page. This allows for more newspaper-like typesetting.

Bookmarks

Many output formats for XSL-FO processors, specifically PDF, have bookmarking features. These allow the format to specify a string of text in a separate window that can be selected by the user. When selected, the document window scrolls immediately to a specific region of the document.

XSL-FO v1.1 now provides the ability to create named bookmarks in XSL-FO, thus allowing the processor to pass this on to an output format that supports it.

Indexing

XSL-FO 1.1 has features that support the generation of an index that might be found at the back of a book. This is done through referencing of properly marked-up elements in the FO document.

Last page citation

The last page can be generated without providing an explicit in-document reference to a specific anchor in the FO document. The definition of "last page" can be restricted to within a specific set of pages or to cover the entire document. This allows the user to specify something like, "Page 2 out of 15", where page 15 is the page number of a last page definition.

Table markers

Table markers allow the user to create dynamic content within table headers and footers, such as running totals at the bottom of each page of a table or "table continued" indicators.

Inside/outside floats

XSL-FO 1.1 adds the keywords "inside" and "outside" for side floats, which makes it possible to achieve page layouts with marginalia positioned on the outside or inside edges of pages.

Inside refers to the side of the page towards the book binding, and outside refers to the side of a page away from the book binding.

Refined graphic sizing

XSL-FO 1.1 refines the functionality for sizing of graphics to fit, with the ability to shrink to fit (but not grow to fit), as well as the ability to define specific scaling steps. In addition, the resulting scaling factor can be referenced for display (for example, to say in a figure caption, "image shown is 50% actual size").

Advantages of XSL-FO

Because it is an XML language, only an XSLT transform (and an XSLT processor) are required to generate XSL-FO code from any XML language. One can easily write a document in TEI or DocBook, and transform it into HTML for web viewing or PDF (through an FO processor) for printing. In fact, there are many pre-existing TEI and DocBook XSLTs for both of these purposes.

Another advantage of XSL-FO is the relative ease of use. Much of the functionality of the language is based on work from CSS, so a CSS user will be familiar with the basics of the markup attributes. Understanding what a specific section of an FO document will look like is usually quite easy. This is compared to layout and typesetting packages like TeX, which are generally considered arcane.

Compared with commercial typesetting and page layout products, XSL-FO can offer a much lower cost solution when it otherwise meets the typographic and layout requirements (see below). The initial cost of ownership is low (zero if the free implementations, such as Apache FOP and xmlroff, meet your requirements), especially compared to the cost of commercial composition tools. The skills required (primarily XSLT programming) are widely available. There are a number of good books on XSL-FO as well as online resources and an active user community.

XSL-FO has been designed to work for all written human languages and the implementations have largely achieved that goal. This makes XSL-FO particularly well suited for composing documents localized into a large number of national languages where the requirement is to have a single tool set that can compose all the language versions of documents. This is especially valuable for technical documentation for things like consumer electronics, where Asian and

Middle Eastern languages are important because those parts of world represent huge markets for things like mobile phones and computer peripherals.

With the publication of XSL-FO 1.1, XSL-FO is proving to be a mature standard with a number of solid commercial and non-commercial implementations. There is no other comparable standard for page composition.

Drawbacks of XSL-FO

XSL-FO was specifically designed to meet the requirements of "lightly designed" documents typified by technical manuals, business documents, invoices, and so on. While it can be and is used for more sophisticated designs, it is inherently limited in what it can do from a layout and typographic perspective. In particular, XSL-FO does not provide a direct way to get formatting effects that depend on knowing the page position relationship of two formatting objects. For example, there is no direct way to say "if this thing is on the same page as that thing, then do X, otherwise do Y". This is an explicit design decision reflecting the two-stage, transform-based abstract processing model used by XSL-FO. This limitation can be addressed by implementing a multi-pass process. Unfortunately, there is currently no standard for how the result of the first pass would be communicated back to the second pass. Most, if not all, implementations provide some form of processable intermediate result format that can be used for this, but any such process implemented today would, by necessity, be implementation specific.

By the same token, there are important layout features that are simply not in XSL-FO, either because they were not of high enough priority or because designing them was too difficult to allow inclusion in version 1.1, or because there were insufficient implementations to allow their inclusion in the final specification per W3C rules.

In addition to these architectural limitations, the current XSL-FO implementations, both commercial and open source, do not provide the same level of typographic sophistication that high-end layout tools like QuarkXPress, Arbortext Advanced Print Publishing, and InDesign provide. For example, no current implementation provides features for ensuring that text lines on facing pages are lined up vertically. There is nothing in the XSL-FO specification that prevents it but nothing that requires it either. For most documents for which a completely-automated composition solution is sufficient, that level of typographic sophistication is not needed. However, for high-end publications and mass-market books, it usually is.

When considering the applicability of XSL-FO to a particular document or document design, one must consider proprietary extensions provided by the different XSL-FO implementations. These extensions add features that are not part of the core specification. For example, one product

adds support for Japanese typographic conventions that the XSL-FO specification does not address. However, use of these features makes such an XSL-FO system a little more bound to a specific implementation (but not completely bound as it would be when using a totally proprietary composition system.)

While it is theoretically possible to have an XSL-FO-based system that allows formatting adjustment following the initial composition process, no current systems provide that functionality and it is generally impractical to edit XSL-FO instances by hand (XSL-FO was designed for clarity and completeness, not ease of editing.)

When trying to decide whether or not XSL-FO will work for a given document, the following typographic and layout requirements usually indicate that XSL-FO will not work (although some of these may be satisfied by proprietary extensions):

- Need to restart footnote numbers or symbol sequence on each new page.
- Need to run text around both sides of a floated object (XSL-FO can run text around one side and the top and/or bottom, but not both sides.)
- Need to have variable numbers of columns on a single page (however, at least two commercial implementations provide extensions for creating multi-column blocks within a page.)
- Need to have column-wide footnotes (several implementations provide column footnote extensions.)
- Need to have marginalia that is dynamically placed relative to other marginalia (for example, marginal notes that are evenly spaced vertically on the page). XSL-FO only provides features for placing marginalia so that it is vertically aligned with its anchor.
- Need to create content that spreads across two pages as a float or "out of line" object in an otherwise homogeneous sequence of repeating page masters (this can be done in XSL-FO 1.1 using multiple body regions and flow maps, but it requires being able to control the page masters used for those pages.)
- Need both bottom-floated content and footnotes on the same page.
- Need to be able to run text against an arbitrary curve (though some implementation support SVG, which can be used to get around this limitation).
- Need to be able to constrain lines to specific baseline grids (for example, to achieve exact registration of lines on facing pages.)
- Anything that requires page-aware layout, such as ensuring that a figure always occurs on the page facing its anchor point.

See also

- XML
- XSL
- XSLT
- Cascading Style Sheets
- XHTML
- Apache FOP - Open source and royalty free implementation of XSL-FO

External links

- XSL-FO 1.0 Specification
- XSL-FO 1.1 Specification
- What is XSL-FO? on XML.com
- W3 Schools A good source for beginners and dabblers
- FO examples and techniques Excellent reference site set up by Dave Pawson, for those who want to use FOP, but were afraid to ask.
- XSL-FO: Ready for Prime Time? on the Gilbane Report
- XSL-FO Tutorial and Samples on Antenna House
- Data2Type (German) XSL-FO information
- XSL Formatting Objects Tutorial on RenderX
- XSL-FO Tutorial and Samples on Ecrion.com

v • d • e
XSL

XSLT (elements) • •
XPath

v • d • e
Standards of the World Wide Web Consortium

Recommendations	Canonical XML · CDF · CSS · DOM · HTML · MathML · OWL · PLS · RDF · RDF Schema · SISR · SMIL · SOAP · SRGS · SSML · SVG · SPARQL · Timed Text · VoiceXML · WSDL · XForms · XHTML · XML · XML Base · XML Events · XML Information Set · XML Schema (W3C) · XML Signature · XPath · XPointer · XQuery · XSL Transformations · · XSL · XLink
Notes	XHTML+SMIL · XAdES
Working Drafts	CCXML · CURIE · InkML · XFrames · XFDL · WICD · XHTML+MathML+SVG · XBL · XProc · HTML 5

GNU Free Documentation License

Version 1.2, November 2002

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. 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, 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.

List of Contributors

Korval
Rik G.
SecVortex
Drmacro
Shlomital
Pvthunderhead
Nigelj
IanOsgood
Gratyn
Charivari
YurikBot
VolkovBot
Tkoeppe
Tagishsimon
Sterrys
Shrewdewd
Securiger
Phe-bot
Nkour
Mild Bill Hiccup
Menteith
Margosbot
Liftarn
K-risc
Jimj wpg
HasharBot
Gareth8118
Furrykef
DStoykov
CrQAZ
Clayoquot
Bota47
Ary29
MarkGaywood
Jayvdb
Patrik008
Typhoonhurricane
Reinthal
Nikitab
JAnDbot
Gudeldar
Fredrik
ZeroOne
Xmlizer
Toussaint
Tim Pritlove
TXiKi
SmackBot
Shanes
Polluks
Pegua
MrVanBot
Michael Hardy
MaxHund
MBread
K1Bond007
Jumbuck
Imroy
Goffrie
Gaius Cornelius
Erwin
Cydebot
CmdrObot
Bunnyhop11
Bawolff