Xqueeze Specifications

Tahir Hashmi

September 2, 2003

Contents

1 About this document **2**

1.1 Copying . . . 2

1.2 Availability . . . 2

1.3 Terms of Use . . . 2

2 xqML language **3**

2.1 xqML Symbols . . . 3

2.1.1 Serialization . . . 4

2.1.2 Reserved Symbols . . . 4

2.2 xqML Grammar (Revision 4) . . . 5

2.2.1 Terminals . . . 5

2.2.2 Productions . . . 5

3 Xqueeze Association **9**

3.1 Xqueeze Association Algorithm (Version 0.2) . . . 9

3.1.1 Dynamic Associations . . . 9

3.2 Xqueeze Association Format (Version 0.3) . . . 10

4 Changes **10**

4.1 xqML . . . 10

4.1.1 Revision 4 . . . 10

4.1.2 Version 0.3 . . . 11

4.1.3 Version 0.2 . . . 11

4.1.4 Version 0.1 . . . 12

4.2 Xqueeze Association algorithm . . . 12

4.2.1 Version 0.2 . . . 12

4.2.2 Version 0.1 . . . 12

4.3 Xqueeze Association format . . . 12

4.3.1 Version 0.3 . . . 12

4.3.2 Version 0.2 . . . 12

4.3.3 Version 0.1 . . . 12

5 GNU Free Documentation License **12**

5.1 Applicability and Definitions . . . 13

5.2 Verbatim Copying . . . 13

5.3 Copying in Quantity . . . 13

5.4 Modifications . . . 14

5.5 Combining Documents 14
5.6 Collections of Documents 15
5.7 Aggregation with Independent Works 15
5.8 Translation 15
5.9 Termination 15
5.10 Future Revisions of This License 15

1 About this document

1.1 Copying

1.2 Availability

The latest version of this document can be downloaded from WWW by pointing your browser to:
http://xqueeze.sourceforge.net/xqueeze-specs.pdf

Draft versions of this document can be downloaded from the CVS Repository of Xqueeze project on Sourceforge.net by pointing your browser to:
http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/xqueeze/docs/

1.3 Terms of Use

Usage of the **Specification** in this document for developing an **Implementation** is subject to the following terms and conditions:

1. An **Implementation** advertizing itself to be based on the **Specification** should support the provisions of the **Specification** in whole or partially, provided it also fulfils condition 4.

2. An **Implementation** advertizing itself to be conforming to the Specification should put into effect the provisions of the **Specification** in whole.

3. The **Specifications** may be modified and re-distributed according to the provisions of GNU FDL (§5).

4. An **Implementation** must not advertize itself to be based on or conforming to the **Specification** if it is based on a modification of the **Specification** unless such modifications are also distributed under the terms of GNU FDL (§5) or compatible license.

5. Items 1 through 4 of these terms of use only apply to specifications and implementations that are distributed commercially or non-commercially to the public.

Glossary

Specification The document or documents as designated by the Original Contributor that defines the form, interface and semantics to the technology covered by the contents of the Specification.

Implementation An implementation of the specification is a program or set of programs that puts into effect the form, interface and semantics defined by the Specification.

Modification Any deletion, addition or alteration to the form, interface or semantics defined by the Specification.

Original Contributor The initiator of the Specification.

2 xqML language

xqML is the binary markup language that is used by Xqueeze to achieve compact document sizes as compared to XML documents. xqML is structurally very similar to XML. The greatest contributors to xqML's compact nature are the elimination of redundant information and representation of XML identifiers (NMTOKENs) whose definitions are available in the DTD/Schema with binary *xqML Symbols*.

2.1 xqML Symbols

xqML Symbols are octect sequences that represent unsigned integers written in Big-Endian (most significant byte first) order. Additionally, the least significant bit of each octet of a symbol, except the last one, should be one. For example, the integer 256 can be a valid xqML Symbol since when written in MSB order, its bit pattern is [00000001 00000000]. Thus the least significant bit of each octet in the symbol acts as a continuation flag. A 1 indicates that the next octet is a part of the symbol, a 0 indicates the end.

It is trivially evident that all xqML Symbols must be even numbers. Additionally, one bit of each octet is rendered unusable since it acts as a continuation flag. 16 bit xqML Symbols can represent 16384 different identifiers while 32 bit ones can represent over 268 million different identifiers. A conforming implementation is required to support atleast 16 bit long symbols.

xqML Symbols start from 0x02 (decimal 2) and symbols up to 0xFE (decimal 254) are reserved for special purposes. Higher values are available for generating associations using the Xqueeze Association algorithm.

A special type of xqML Symbols – *VUint* – is defined to represent variable length unsigned integral values that can represent arbitrarily large values. The difference in interpretation of the values of normal xqML Symbols and VUints is that the continuation bits do not contribute to the value of the integer. For example, while the xqML Symbol with decimal value 256 will be represented as `00000001 00000000`, a VUint will be represented as `00000101 00000000`. If we strip the continuation bits from the latter, we get `0000010 0000000` which represents decimal 256.

2.1.1 Serialization

xqML Symbols are serialized in Big-Endian (most significant byte first) order and are represented in only as many octets (8-bit groups) as required, irrespective of the encoding used for character data. As an exception, some of the symbols should be serialized as characters whose code points equal the value of the corresponding symbol. These are the symbols with values 0x02, 0x06, 0x14, 0x16, 0x18, 0x1A, 0x1C and 0x1E.

2.1.2 Reserved Symbols

xqML Symbols with values between 0x02 and 0xFE (both inclusive) are reserved for grammar specific purposes. Table 1 lists the used xqML Symbols in xqML revision 4 and their purpose. Entries in *italics* are productions from the xqML grammar listed in §2.2 and the respective production numbers are provided in brackets.

Symbol Value		**Purpose**
Hex	**Dec**	
0x02	002	*Fmt* (4)
0x04	004	
0x06	006	*Fmt* (4)
0x14	020	*ATAttribute* (13)
0x16	022	*ATAttribute* (13)
0x18	024	*APAttribute* (14)
0x1A	026	*APAttribute* (14)
0x1C	028	*NSDecl* (15)
0x1E	030	Markup flag
0x20	032	*PI* (21)
0x22	034	*EntityRef* (18)
0x24	036	*EntityRef* (18)
0x26	038	*CharRef* (19)
0x28	040	*RegId* (9)
0x2A	042	*RegId* (9)
0x2C	044	*doctypedecl* (5)
0x2E	046	*DTDSect* (7)
0x30	048	*ETag* (20)
0x32	050	*ELFlags* (11)
0x34	052	*ELFlags* (11)
0x36	054	*ELFlags* (11)
0x38	056	*ELFlags* (11)
0x3A	058	*ELFlags* (11)
0x3C	060	*ELFlags* (11)
0x3E	062	*ELFlags* (11)
0x40	064	xqA end marker

Table 1: Table of Reserved Symbols

2.2 xqML Grammar (Revision 4)

2.2.1 Terminals

- *Figures enclosed within braces ({})* are hex codes for the value of an xqML Symbol that should occur within.
- *Rev* is an octet to be interpreted as an unsigned integer.
- *xqA* is the inline Xqueeze Association with prolog (see § 3.2).
- *ELSymbol*, *ATSymbol*, *APSymbol*, *VASymbol* and *ENSymbol* are all xqML symbols derived from an Xqueeze Association to represent the vocabulary of an XML document type.
- *NSSymbol* is an xqML Symbol of the type "namespace prefix" (has a document-specific value).
- *ElementsToClose* is an octet to be interpreted as an unsigned integer.
- *VUint* is a special type of xqML Symbols that represents Variable-length Unsigned integers (see § 2.1).

2.2.2 Productions

1. $document\ ::=\ prolog\ element\ PI*$

   Every xqML document must match the above production. Thus, *document* is the starting symbol.

2. $prolog\ ::=\ [\hat{}\{0x1E\}]*\ xqMLDecl\ PI*\ (doctypedecl\ PI*)?$

   The *prolog* of an xqML document can contain anything upto the first occurrence of xqML Symbol {0x1E}.

3. $xqMLDecl\ ::=\ '\{0x1E\}'\ Fmt\ Rev\ Char*$

   Every xqML document must declare what it is (xqML), its binary format and the version of its encoding. *Rev* is an octet that represents the revision number of the xqML encoding used (see the change in § 4.1.1). This octet should be interpreted as an unsigned integer.

4. $Fmt\ ::=\ '\{0x00\}\{0x02\}'\ |\ '\{0x06\}'$

   Format is a sequence that informs the parser whether the stream is encoded in 8-bit format (like UTF-8) or a 16-bit format encoding (like UTF-16). Note that this is not entirely dependent on character encodings since there may be multiple character encodings in each format. For example, the ISO-8859 family of encodings is 8-bit.

   As an example, the xqML counterpart of the XML declaration:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   ```

   looks like:

   ```
   ␣␣␣␣UTF-8
   ```

where ␣ is a visual representation of an xqML Symbol. The symbols in the above example are 0x1E, 0x00, 0x02, and *Rev* – in that sequence. Each xqML revision number corresponds to a specific XML version number. The special attribute “standalone” is not written and is always assumed to be “no”.

5. $doctypedecl \ ::= \ ('\{0x1E\}\{0x2C\}' \ DoctypeName) \mid xqA \mid DTDSect$

   An xqML document may declare its document type in one of three ways:

   (a) Declare a *DoctypeName* (production 6) that identifies an external xqA specification

   (b) Include an xqA specification (including prolog) inline

   (c) Include a DTD inline in a *DTDSect* (production 7)

6. $DoctypeName \ ::= \ Char*$

   DoctypeName should be a valid URI from which an xqA specification may be retrievable. However, the parser is not responsible for checking the validity of a *DoctypeName*.

7. $DTDSect \ ::= \ '\{0x1E\}\{0x2E\}' \ Char*$

   DTDSect contains an internal DTD in the format specified in XML 1.0 specification, including the `DOCTYPE` tag. An xqML parser must be capable of generating an xqA specification out of the DTD but is not always required to do so.

8. $element \ ::= \ RegId* \ STag \ (content \ ETag?)?$

   This corresponds to an XML Element. The element must have a start tag *STag*. The start tag also contains an indication of whether the element is empty or not. If the element is not empty, it would also contain *content* and a closing tag. The closing tag *ETag* is optional since several consecutive closing tags are combined into one in xqML.

9. $RegId \ ::= \ '\{0x1E\}' \ ('\{0x2A\}' \mid ('\{0x28\}' \ NSSymbol)) \ Char*$

   This production corresponds to an identifier registration in the Dynamic Association mapping of the document (see §3.1.1). The string at the end of this production is taken as the identifier to be registered. The declaration may explicitly indicate association with a particular namespace through the use of an *NSSymbol*.

10. $STag \ ::= \ '\{0x1E\}' \ (ELFlags \ NSSymbol?)? \ ELSymbol \ (attribute \mid NSDecl)*$

    This represents an element start tag. *ELFlags* is an octet that has three status flag bits. *NSSymbol* is a symbol for XML Namespace prefix. ELSymbol is the symbol for the element's identifier. This may be followed by any number of attributes or XML Namespace declarations (*NSDecl*).

11. $ELFlags \ ::= \ 0x32 - 0x3E$

    This octet contains three status flags in its 2^{nd}, 3^{rd} and 4^{th} least significant bits to signify the following:

    (a) *Empty Element*: The second least significant bit of the octet is set if the element is empty

(b) *Namespace Prefix*: The third least significant bit is set if an *NSSymbol* follows

(c) *Close Previous*: If the fourth least significant bit is set, it indicates that the last open element should be closed.

The four most significant bits are `0011`. Therefore this octet can have values between 0x30 and 0x3E. However, if all the flag bits are unset, the resultant value, 0x30, is never written. This value is used to indicate one or more closing tags (See production 20).

12. $attribute \ ::= \ ATAttribute \mid APAttribute$

Attributes may have unspecified values (*ATAttribute*) or values that have been assigned symbols in the xqA specification (*APAttribute*).

13. $ATAttribute \ ::= \ (('\{0x14\}' \ NSSymbol) \mid '0x16') \ ATSymbol \ Char* \ (Reference \ Char*)* \ '\{0x16\}'$

An attribute is started by the symbol 0x16, or by the symbol 0x14 followed by an *NSSymbol*. The symbol for the attribute identifier, *ATSymbol*, comes next. The attribute is closed by the symbol 0x16. Any character data or references before the closing delimiter is taken to be the value of the attribute.

14. $APAttribute \ ::= \ (('\{0x18\}' \ NSSymbol) \mid '0x1A') \ APSymbol \ VASymbol$

Attributes with predefined values begin with the symbol 0x18, or by the symbol 0x1A followed by an *NSSymbol*. *APSymbol* is the symbol for the attribute identifier and *VASymbol* is the symbol for it's value. These attributes are completely represented by symbols.

For example, the xqML counterpart of `<ufn:file path="/etc/issue.net" binary="no"/>`, where the attribute "binary" has enumerated values "yes" and "no", would be:

`␣␣␣␣␣␣/etc/issue.net␣␣␣␣`

Here we have six symbols, followed by the string "/etc/issue.net" followed by four more symbols. The symbols would be:

(a) 0x1E

(b) 0x36 (ELFlags, indicating an empty element and a namespace prefix to follow)

(c) A document specific symbol for the namespace prefix "ufn"

(d) The symbol for element identifier "file"

(e) 0x16 – to signify an attribute of type *ATAttribute*

(f) The symbol for attribute identifier "path"

The value of "path" follows as char data. The next four symbols would be:

(a) 0x16 – to mark the end of attribute "path"

(b) 0x1A – to signify an attribute of type *APAttribute*

(c) The symbol for attribute identifier "binary"

(d) The symbol for attribute value “no”

15. $NSDecl\ ::=\ '\{0x1C\}'\ Char*\ '\{0x1E\}'\ Char*\ '\{0x1E\}'$

    These are the xqML equivalents of xmlns declarations in XML. For example, the declaration:
    `xmlns:xsl="http://www.w3.org/1999/XSL/Transform"`
    would be encoded as:

    ```
    ␣xsl␣http://www.w3.org/1999/XSL/Transform␣
    ```

    There may be a null string instead of “xsl” in the above example.

16. $content\ ::=\ Char*\ ((element \mid Reference)\ Char*)*$

    An element may contain character data and any number of child elements, references or character data in any order. Restrictions imposed by document type specifications (DTD, XML Schema etc.) may apply while validating.

17. $Reference\ ::=\ EntityRef \mid CharRef$

18. $EntityRef\ ::=\ RegId*\ '\{0x1E\}'\ (('\{0x24\}'\ NSSymbol) \mid '\{0x22\}')\ ENSymbol$

    This production matches an entity reference. ENSymbol is the symbol for the entity identifier, *not* its expansion.

19. $CharRef\ ::=\ '\{0x1E\}\{0x26\}'\ VUint$

    This production matches a Character Reference. *VUint* is a Variable-length Unsigned integer, whose value equals the code point of the desired character.

20. $ETag\ ::=\ '\{0x1E\}\{0x30\}'\ ElementsToClose$

    The closing tag has an octet *ElementsToClose* which should be interpreted as the binary representation of an unsigned integer, whose value signifies the number of elements to close in correct (stack) order.

21. $PI\ ::=\ '\{0x1E\}\{0x20\}'\ PITarget\ '\{0x1E\}'\ PIContent\ '\{0x1E\}'$

    This is a representation of an XML Processing Instruction. *PITarget* is the equivalent of targets in XML PIs. *PIContent* is the data that is passed on to the application. For example, a hypothetical SSI include directive for a web server may be written in XML as `<?ssi includefile("headers.shtml")?>`. The xqML equivalent of this would be:

    ```
    ␣␣ssi␣includefile("headers.shtml")␣
    ```

    where the symbols are 0x1E, 0x02, 0x1E and 0x1E in that order.

22. $PITarget\ ::=\ Char*$

23. $PIContent\ ::=\ Char*$

24. $Char\ ::=\ 0x09 \mid 0x0A \mid 0x0D \mid [0x20{-}0xD7FF] \mid [0xE00{-}0xFFFD] \mid [0x10000{-}0x10FFFF]$

    xqML characters are exactly same as XML characters. Additionaly, the characters ‘$<$’, ‘$>$’, ‘$'$’, ‘"’ and ‘&’ need not be escaped, unlike XML.

3 Xqueeze Association

Xqueeze uses an association between symbols and their corresponding XML identifiers and types as defined in a specification (DTD/Schema). This enables representation of known identifiers in the markup with symbols. Associating the type of an identifier along with it's name also makes it easy to various structural units of the document without having to use too many special characters and character-combinations.

3.1 Xqueeze Association Algorithm (Version 0.2)

This is the allgorithm that is used to map the identifiers found in a DTD/Schema to xqML Symbols. The steps of the algorithm are:

1. collect all Element identifiers
2. collect all Attribute identifiers
3. collect all Enumerated Attribute Value identifiers
4. collect all Entity References together
5. merge the above collections, discarding duplicates
6. sort the merged collection lexically on the values of unicode code-points
7. assign symbols starting from 256 in ascending order to the identifiers

This simple algorithm assures that the assignments would remain the same even if a particular specification (DTD/Schema) has slight variations in the way it's written in the generator's and consumer's copies, as long as both define the same vocabulary. Note that it is not dependent on the structure of the document.

3.1.1 Dynamic Associations

Xqueeze allows for associating symbols to identifiers within a running document through *Dynamic Associations*. This allows for generation of xqML documents without the knowledge of whole or part of the schema. Dynamic Associations cover elements, attributes and entity references. Attribute values are not covered, and should be written as string literals.

For assigning symbols to dynamically declared identifiers, the processor must maintain a separate lookup table for each namespace with which one or more dynamic identifier declarations are associated. The namespace with which to associate a dynamically declared identifier is determined by these rules:

1. Declarations appearing ahead of an element are associated with the namespace that the element is associated with
2. Declarations appearing ahead of an entity reference are associated with the namespace that the containing element is associated with
3. Declarations with explicit namespace prefixes are associated with the namespace denoted by the prefix, provided the prefix is valid and legal

While registering identifiers dynamically, duplicate declarations within the same namespace are discarded. This means that identifiers that already exist in a given namespace would not be re-assigned.[1] Symbols are assigned to identifiers in the order of their appearance in the document, starting from the first unused symbol in the Association corresponding to the namespace in context.

Portability of such associations is limited to the document that contained the declarations and parts of the document using dynamically assigned symbols can't be used elsewhere, without translation and re-assigning of symbols. Nor can the document be safely modified without preserving the declarations.

3.2 Xqueeze Association Format (Version 0.3)

Xqueeze associations are represented in a format that itself is quite compact and uses xqML Symbols themselves. The specification begins with an optional prolog whose format resembles that of an xqML *PI* (Processing Instruction):
$'\{0x1E\}\{0x20\}xqa\{0x1E\}'\ Char*\ '\{0x1E\}'$

Here, *Char** can contain the xqA declaration for the document. The prolog is followed by individual entries for identifiers.

Individual entries are listed as '{0x1E}', followed by a symbol, followed by a string that the symbol represents. The end of specifications is denoted by the sequence '{0x1E}{0x40}'. This structure enables inline specification of the symbols associations, if required by a document.

4 Changes

4.1 xqML

4.1.1 Revision 4

- xqML will now have "Revisions" instead of version numbers. The current format can report a maximum of 255 revisions. However, this does not imply that there will not be more than 255 revisions of xqML
- Comments have now been dropped
- CDATA Sections have now been dropped
- The format now allows for generation of documents without prior knowledge of schema through *Dynamic Associations* (§ 9)
- The *xqMLDecl* represents xqML revision information in binary now
- A new terminal *Rev* has been added
- *xqMLDecl* is now mandatory for all xqML documents
- All xqML documents have the value of special attribute "standalone" as "no"
- *ELFlags* production added to combine three flags related to elements into one octet

[1] Therefore it is a good practice to declare dynamic identifiers in separate namespace(s) while mixing with various vocabularies.

- *ATAttribute* ends with ‘{0x16}’ instead of ‘{0x1E}{0x16}’
- A new terminal and xqML Symbol type, *VUint* has been added (see § 2.1)
- *CharRef* now uses VUint to encode the character’s code point value
- The production *EE_STag* has been dropped
- *ETag* now uses ‘{0x30}’ instead of ‘{0x3E}’.

4.1.2 Version 0.3

- Anything is permissible upto the occurrence of *xqMLDecl* in a document
- A new production, *PI*, has been added for Processing Instructions
- *doctypedecl* now starts with ‘{0x1E}{0x2C}’ instead of ‘{0x1E}{0x12}’
- *xqA* should necessarily include a prolog now
- *doctypedecl* may now have an inline DTD with a new production *DTDSect*.
- *element* production was erroneous till the last version
- A new prodcution *NSPrefix* has been added for XML Namespace prefixes
- The productions *EE_STag*, *STag*, *ATAttribute*, *APAttribute* and *EntityRef* can now have namespace prefixes
- *EE_STag* starts with ‘{0x1E}{0x2A}’ instead of ‘{0x1E}’
- *ATAttribute* starts with ‘{0x16}’ instead of ‘{0x1E}’
- *APAttribute* starts with ‘{0x18}’ instead of ‘{0x1E}’
- *EntityRef* starts with ‘{0x1E}{0x24}’ instead of ‘{0x1E}’
- *CDSect* starts with ‘{0x1E}{0x28}’ instead of *CDDelim* and ends with ‘{0x1E}’ instead of *CDDelim* ({0x1E}{0x14}).
- *Char* now matches the *Char* production in XML 1.0 grammar specification.

4.1.3 Version 0.2

- xqML Symbol ‘{0x1E}’ replaces ‘<’ for the latter’s role in xqML markup
- *Attribute* is split into *ATAttribute* and *APAttribute*, together referred as *attribute*.
- *ATAttribute* can contain *Reference*.
- *ATAttribute* is terminated by ‘{0x1E}{0x16}’ instead of ‘<’
- *CharRef* starts with ‘{0x1E}{0x26}’ instead of ‘&{0x26}’
- *CharRef* ends with ‘{0x1E}’. Earlier there was no end-marker
- *Comment* ends with ‘{0x1E}’ instead of *ETag?*
- *Comment* is deprecated
- *Char* is a terminal that matches any printable character
- *Num* does not contain ‘.’

4.1.4 Version 0.1

First Release

4.2 Xqueeze Association algorithm

4.2.1 Version 0.2

- Removed distinction of identifiers based on type
- Added support for Dynamic Associations

4.2.2 Version 0.1

First Release

4.3 Xqueeze Association format

4.3.1 Version 0.3

- Removed section markers
- xqA specification now ends with the sequence '{0x1E}{0x40}' instead of '{0x1E}{0x3C}'.

4.3.2 Version 0.2

- xqML Symbol '{0x1E}' replaces '<' for the latter's role in xqA format.
- The prolog format has been changed to resemble an xqML PI.
- Reserved symbols used in the previous version have been shifted 44 decimal values up. For example, the symbol for Element section is now '{0x30}' (48) instead of '{0x04}' (04).

4.3.3 Version 0.1

First Release

5 GNU Free Documentation License

GNU Free Documentation License
Version 1.2, November 2002

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.

5.1 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.

5.2 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.

5.3 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.

5.4 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.

5.5 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”.

5.6 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.

5.7 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 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.

5.8 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.

5.9 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.

5.10 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.

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 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled ”GNU Free Documentation License”.

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

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

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