

SharpDevelop
C# Coding Style Guide

Version 0.2

Mike Krueger

January 4, 2002

Copyright (c) 2001 Mike Krueger (mike@icsharpcode.net) Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Contents

1	About the SharpDevelop C# Coding Style Guide	7
2	File Organization	9
2.1	C# Sourcefiles	9
2.2	Directory Layout	9
3	Indentation	11
3.1	Line Length	11
3.2	Wrapping Lines	11
3.3	Whitespaces	12
4	Comments	13
4.1	Block Comments	13
4.2	End of Line Comments	13
4.3	Documentation Comments	13
5	Declarations	15
5.1	Number of Declarations per Line	15
5.2	Initialization	15
5.3	Class and Interface Declarations	16
6	Statements	17
6.1	Simple Statements	17
6.2	Return Statements	17
6.3	If, if-else, if else-if else Statements	17
6.4	For / Foreach Statements	18
6.5	While/do-while Statements	18
6.6	Switch Statements	18
6.7	Try-catch Statements	19
7	White Spaces	21
7.1	Blank Lines	21
7.2	Inter-term spacing	21
7.3	Table like formatting	21
8	Naming Conventions	23

Contents

8.1	Capitalization Styles	23
8.1.1	Pascal Casing	23
8.1.2	Camel Casing	23
8.1.3	Upper case	23
8.1.4	Capitalization summary	24
8.2	Class and Class members	24
8.2.1	Class Naming Guidelines	24
8.2.2	Interface Naming Guidelines	24
8.2.3	Enum Naming Guidelines	25
8.2.4	ReadOnly and Const Field Names	25
8.2.5	Parameter/non const field Names	25
8.2.6	Variable Names	25
8.2.7	Method Names	25
8.2.8	Property Names	25
8.2.9	Event Names	26
9	Programming Practices	27
9.1	Visibility	27
9.2	No magic Numbers	27
10	Code Examples	29
10.1	Brace placement example	29
10.2	Variable naming example	30
A	GNU Free Documentation License	31
A.1	Applicability and Definitions	31
A.2	Verbatim Copying	32
A.3	Copying in Quantity	33
A.4	Modifications	33
A.5	Combining Documents	35
A.6	Collections of Documents	36
A.7	Aggregation With Independent Works	36
A.8	Translation	36
A.9	Termination	36
A.10	Future Revisions of This License	37

1 About the SharpDevelop C# Coding Style Guide

The SharpDevelop C# Coding Style Guide was written by Mike Krueger. This document can also be read as a guide to writing robust and reliable programs. It focuses on programs written in C# , but many of the rules and principles are useful even if you write in another programming language.

1 About the SharpDevelop C# Coding Style Guide

2 File Organization

2.1 C# Sourcefiles

Keep your classes/files short, don't exceed 2000 LOC, divide it up, make your structure clearer. Put every class in a separate file and name the file like the class name (with .cs as extension of course)

2.2 Directory Layout

Make a directory for every namespace. (for MyProject.TestSuite.TestTier use MyProject/TestSuite/TestTier as the path use not the namespace name with dots)

2 *File Organization*

3 Indentation

3.1 Line Length

Consider avoiding (if possible) lines longer than 80 characters, switch on the ruler in your editor to get that managed, wrap lines if necessary.

3.2 Wrapping Lines

When an expression will not fit on a single line, break it according to these general principles:

- Break after a comma.
- Break after an operator.
- Prefer higher-level breaks to lower-level breaks.
- Align the new line with the beginning of the expression at the same level on the previous line

Example of breaking method calls.

```
longMethodCall(expr1, expr2, expr3,  
               expr4, expr5);
```

Examples of breaking an arithmetic expression. The first is preferred, since the break occurs outside of the parenthesized expression (higher level rule).

```
var = a * b / (c - g + f) +  
      4 * z; // PREFER  
  
var = a * b / (c - g +  
              f) + 4 * z; // AVOID
```

3.3 Whitespaces

Nobody ever achieved to establish an indentation standard with spaces. Some people like 2, some say 4 and others die for 8, or even more, spaces. I say: use tabs. Ok you don't like tabs, but tab characters have some advantages:

- Everyone can set their own, preferred indentation level
- It is only 1 character and not 2, 4, 8 therefore it will reduce typing (even with smartindenting you have to set the indentation manually sometimes, or take it back or whatever)
- If you want to increase the indentation (or decrease) you mark one block and increase the indent level with Tab with Shift-Tab you decrease the indentation. This is true for almost any texteditor

Ok, that should be enough, I don't want to start the tabs vs. spaces war, I define the Tab as the standard indentation character.

!!! DON'T USE SPACES FOR INDENTATION USE TABS !!!!

4 Comments

4.1 Block Comments

I try to avoid block comments, for descriptions I use the `///` comments to give C# standard descriptions. When you wish to use block comments you should use the following style :

```
/* Line 1
 * Line 2
 * Line 3
 */
```

or (In this case a return should follow, don't put comments before code)

```
/* blabla */
```

4.2 End of Line Comments

You should use the `//` comment style to "comment out" code (SharpDevelop has a key for it, `Alt+/`) . It can be used for commenting sections of code too.

4.3 Documentation Comments

For documentation comments see the Microsoft Documentation.

4 Comments

5 Declarations

5.1 Number of Declarations per Line

One declaration per line is recommended since it encourages commenting. In other words,

```
int level; // indentation level
int size;  // size of table
```

Do not put different types on the same line.

Example:

```
int a, b; //WRONG!
```

5.2 Initialization

Try to initialize local variables where they're declared. Example:

```
string name = myObject.Name;
```

or

```
int    val = time.Hours;
```

5.3 Class and Interface Declarations

When coding C# classes and interfaces, the following formatting rules should be followed: No space between a method name and the parenthesis "(" starting its parameter list. The open brace "{" appears in the next line as the declaration statement, the closing brace "}" starts a line by itself indented to match its corresponding opening statement.

Example :

```
class MySample : MyClass, IMyInterface
{
    int myint;

    public MySample(int myint)
    {
        this.myint = myint;
    }

    void Inc()
    {
        ++myint;
    }

    void EmptyMethod()
    {
    }
}
```

6 Statements

6.1 Simple Statements

Each line should contain only one statement.

6.2 Return Statements

A return statement should not use parentheses.

6.3 If, if-else, if else-if else Statements

if, if-else and if else-if else statements should look like :

```
if (condition) {
    DoSomething();
    ...
}

if (condition) {
    DoSomething();
    ...
} else {
    DoSomethingOther();
    ...
}

if (condition) {
    DoSomething();
    ...
} else if (condition) {
    DoSomethingOther();
    ...
} else {
    DoSomethingOtherAgain();
    ...
}
```

6.4 For / Foreach Statements

A for statement should have following form :

```
for (int i = 0; i < 5; ++i) {  
    ...  
}
```

or single lined :

```
for (initialization; condition; update) ;
```

A foreach should look like :

```
foreach (int i in IntList) {  
    ...  
}
```

6.5 While/do-while Statements

A while statement should have following form:

```
while (condition) {  
    ...  
}
```

An empty while should have the following form:

```
while (condition) ;
```

A do-while statement should have following form:

```
do {  
    ...  
} while (condition);
```

6.6 Switch Statements

A switch statement should have following form:

```
switch (condition) {  
    case A:  
        ...  
        break;  
    case B:  
        ...  
        break;  
    default:  
        ...  
        break;  
}
```

6.7 Try-catch Statements

A try-catch statement should have following form:

```
try {  
    ...  
} catch (Exception) {}
```

or

```
try {  
    ...  
} catch (Exception e) {  
    ...  
}
```

or

```
try {  
    ...  
} catch (Exception e) {  
    ...  
} finally {  
    ...  
}
```

6 *Statements*

7 White Spaces

7.1 Blank Lines

Blank lines improve readability, set off code blocks which are logically related. Two blank lines should always be used between:

- Sections of a source file
- Class and interface definitions

One blank line should always be used between:

- Methods
- Local variables in a method and its first statement
- logical sections inside a method to improve readability

7.2 Inter-term spacing

There will be a single space after a comma or a semicolon, example:

```
TestMethod(a, b, c);    don't use : TestMethod(a,b,c)
                        or
                        TestMethod( a, b, c );
```

A single space will surround operators, example:

```
a = b;                // don't use a=b;
for (int i = 0; i < 10; ++i) // don't use for (int i=0; i<10; ++i)
                        // or
                        // for(int i=0;i<10;++i)
```

7.3 Table like formatting

A logical block of lines should be formatted as a table:

```
string name    = "Mr. Ed";
int    myValue = 5;
Test   aTest   = Test.TestYou;
```

7 *White Spaces*

8 Naming Conventions

8.1 Capitalization Styles

8.1.1 Pascal Casing

This convention capitalizes the first character of each word (like in `TestCounter`)

8.1.2 Camel Casing

This convention capitalizes the first character of each word except the first one. (like in `testCounter`)

8.1.3 Upper case

Only use all upper case for identifiers if it contains an abbreviation that are two characters long or one, identifiers of three or more characters should use Pascal Casing instead. For Example:

```
public class Math
{
    public const PI = ...
    public const E = ...
}
```

8.1.4 Capitalization summary

Type	Case	Notes
Class / Struct	Pascal Casing	
Interface	Pascal Casing	Starts with I
Enum values	Pascal Casing	
Enum type	Pascal Casing	
Events	Pascal Casing	
Exception class	Pascal Casing	End with Exception
public Fields	Pascal Casing	
Methods	Pascal Casing	
Namespace	Pascal Casing	
Property	Pascal Casing	
Protected/private Fields	Camel Casing	
Parameters	Camel Casing	

8.2 Class and Class members

Generally the use of underscore characters and hungarian notation is not allowed. Hungarian notation are a defined set of pre and postfixes which are applied to names to reflect the type of the variable, this is not allowed if you follow this guide. And remember that a good name describes the semantic not the type.

8.2.1 Class Naming Guidelines

- Class names must be nouns or noun phrases.
- Use Pascal Casing (see 8.1.1)
- Do not use any class prefix

8.2.2 Interface Naming Guidelines

- Do name interfaces with nouns or noun phrases or adjectives describing behavior. (Example `IComponent` or `IEnumerable`)
- Use Pascal Casing (see 8.1.1)
- Use I as prefix for the name, it is followed by a capital letter (first char of the interface name)

8.2.3 Enum Naming Guidelines

- Use Pascal Casing for enum value names and enum type names
- Dont prefix (or suffix) a enum type or enum values
- Do use singular names for enums
- Do use plural name for bit fields.

8.2.4 ReadOnly and Const Field Names

- Do name static fields with nouns, noun phrases or abbreviations for nouns
- Use Pascal Casing (see 8.1.1)

8.2.5 Parameter/non const field Names

- Do use descriptive names, which should be enough to determine the variable meaning and its type. But prefer a name thats based on the parameters meaning.
- Use Camel Casing (see 8.1.2)

8.2.6 Variable Names

- Counting variables are preferably called i, j, k, l, m, n. (see 10.2 for an example on more intelligent naming)
- Use Camel Casing (see 8.1.2)

8.2.7 Method Names

- Do name methods with verbs or verb phrases.
- Use Pascal Casing (see 8.1.1)

8.2.8 Property Names

- Do name properties using nouns or noun phrases
- Use Pascal Casing (see 8.1.1)
- Consider naming a property with the same name as its type

8.2.9 Event Names

- Do name event handlers with the `EventHandler` suffix.
- Do use two parameter named `sender` and `e`
- Use Pascal Casing (see 8.1.1)
- Do name event argument classes with the `EventArgs` suffix.
- Do name event names that have a concept of pre and post using the present and past tense.
- Consider naming events with a verb.

9 Programming Practices

9.1 Visibility

Dont make any instance or class variable **public**, make them **private**. Do use properties instead. You may use **public static** fields (or **const**) as an exception, but it should not be the rule.

9.2 No magic Numbers

Dont use magic number, instead declare a **const** variable which contains the number :

```
public class MyMath
{
    public const double PI = 3.14159...
}
```

9 *Programming Practices*

10 Code Examples

10.1 Brace placement example

```
namespace ShowMeTheBracket {
    public enum Test {
        TestMe,
        TestYou
    }

    public class TestMeClass
    {
        Test test;

        public Test Test {
            get {
                return test;
            }
            set {
                test = value;
            }
        }

        void DoSomething()
        {
            if (test == Test.TestMe) {
                ...
            } else {
                ...
            }
        }
    }
}
```

10.2 Variable naming example

instead of :

```
for (int i = 1; i < num; ++i) {
    meetsCriteria[i] = true;
}
for (int i = 2; i < num / 2; ++i) {
    int j = i + i;
    while (j <= num) {
        meetsCriteria[j] = false;
        j += i;
    }
}
for (int i = 0; i < num; ++i) {
    if (meetsCriteria[i]) {
        Console.WriteLine(i + " meets criteria");
    }
}
```

try intelligent naming :

```
for (int primeCandidate = 1; primeCandidate < num; ++primeCandidate) {
    isPrime[primeCandidate] = true;
}
for (int factor = 2; factor < num / 2; ++factor) {
    int factorableNumber = factor + factor;
    while (factorableNumber <= num) {
        isPrime[factorableNumber] = false;
        factorableNumber += factor;
    }
}
for (int primeCandidate = 0; primeCandidate < num; ++primeCandidate) {
    if (isPrime[primeCandidate]) {
        Console.WriteLine(primeCandidate + " is prime.");
    }
}
```

A GNU Free Documentation License

Version 1.1, March 2000

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

Preamble

The purpose of this License is to make a manual, textbook, or other written document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. 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.

A.1 Applicability and Definitions

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

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document

to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not “Transparent” is called “Opaque”.

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

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

A.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.

A.3 Copying in Quantity

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

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

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

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

A.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:

- 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.

A GNU Free Documentation License

- List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- State on the Title page the name of the publisher of the Modified Version, as the publisher.
- Preserve all the copyright notices of the Document.
- Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- 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.
- Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document’s license notice.
- Include an unaltered copy of this License.
- Preserve the section entitled “History”, and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- 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.
- In any section entitled “Acknowledgements” or “Dedications”, preserve the section’s title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- 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.
- Delete any section entitled “Endorsements”. Such a section may not be included in the Modified Version.
- Do not retitle any existing section as “Endorsements” or to conflict in title with any Invariant Section.

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

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

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

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

A.5 Combining Documents

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

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled “History” in the various original documents, forming one section entitled “History”; likewise combine any sections entitled “Acknowledgements”, and any sections entitled “Dedications”. You must delete all sections entitled “Endorsements.”

A.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.

A.7 Aggregation With Independent Works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an “aggregate”, and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document’s Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

A.8 Translation

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

A.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.

A.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 © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled “GNU Free Documentation License”.

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

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