

Bluefish: The Definitive Guide

**Denny Reeh
Alastair Porter**

Bluefish: The Definitive Guide

by Denny Reeh and Alastair Porter

Published (TBA)

Table of Contents

1. General Information	1
About this Manual	1
What is Bluefish?	1
History of Bluefish	1
Main Features of Bluefish	2
How Stable is Bluefish?	3
Contact Us	3
2. bluefish installation	5
requirements	5
quick standard installation	5
general installation issues	6
how to get bluefish	6
operating systems supported by bluefish	6
which bluefish version to use	7
how and when updates are released	7
system specific notes	7
installing a bluefish source distribution	8
quick installation overview	8
typical configure options	8
installing from development source tree	9
problems compiling?	10
installing a binary distribution	10
post-installation setup	11
3. tutorial introduction	12
starting bluefish	12
using bluefish, an overview	12
4. using bluefish	13
the menu	13
open a file	13
saving files	13
closing a file	13
quitting bluefish	13
changing shortcut keys	13
the toolbar	14
the file editing area	14
editing text	14
statusbar	14
quickbar	14
custom menu	14
what is the custom menu?	14
editing the custom menu	14
syntax highlighting	14
using syntax highlighting	14
modifying highlight patterns	14
bluefish preferences	16
editor preferences	16
html preferences	16
files preferences	16
user interface preferences	16
images preferences	17
file types preferences	17
syntax highlightning preferences	17
external programs preferences	17

- 5. advanced concepts in using and customizing bluefish 18
 - special things in working with programming languages 18
 - HTML 18
 - PHP 18
 - ... and some other languages 18
 - filebrowser (filtering, filetypes, etc.) 18
 - search and replace (regex patterns) 18
 - project management 19
 - more.. 19
- 6. reference 20
- 7. development guidelines 21
 - debugging bluefish 21
 - indenting and formating style 21
 - naming 21
 - declaring procedures 21
 - header files 21
 - new files 21
 - file reference 21
 - patches 21
 - cvs 21
 - usage 21
 - guidelines 21
 - translations 22
 - some tips 22
 - making releases 22
 - useful stuff 22
- A. credits 23
 - bluefish developer 23
 - supporters to bluefish 23
- B. bluefish change history 24
 - changes in release gtk2-port 24
 - changes in release gtk1-version 24
- C. manual writing tutorial 25
 - requirements for maintain the manual 25
 - conventions for writting in this manual 25
- D. GNU GENERAL PUBLIC LICENSE 26
 - Preamble 26
 - TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION 26
 - 26
 - How to Apply These Terms to Your New Programs 29

Chapter 1. General Information

About this Manual

Bluefish has a large feature set, allowing the user to customize the editing experience in numerous ways. This manual targets the novice user, providing a starting point for mastering Bluefish. However, advanced users will still find this text an excellent reference.

Chapters 1-4 are highly recommended for anyone new to Bluefish. They present general information, installation instructions, and an introduction to the main features of Bluefish.

Chapters 5 and 6 contain a nearly complete feature reference, useful for advanced users interested in customizing Bluefish. Some examples of regular expressions used for find and replace are also included here.

As a reference to the Bluefish developers, or anyone interested in contributing code, chapter 7 contains guidelines for the Bluefish source.

The manual is targeted the *end user*. Thus, we have tried to use a simple, well-explained approach for the large part of the manual. Some typographic conventions are denoted below.

- Any urls are denoted like this: `http://bluefish.openoffice.nl`
- Shortcuts look like this: Ctrl-S
- Menu options are displayed like this: File However, many of Bluefish's menus are quite complex. When referring to submenus, options are separated by the right angle-bracket (>), like: File > Open .
- When referring to user input, like issuing commands to the command prompt, a monotype font is used:

```
$ foo -bar | bang -l
```

Note: Do not write the \$ character - it's simply identifying the command prompt. For commands requiring root access, the prompt is shown as a #.

What is Bluefish?

Bluefish is a powerful editor for experienced web designers and programmers based on the GTK2 GUI interface. Bluefish supports many programming and markup languages, but focuses on editing dynamic and interactive websites.

Bluefish is not a WYSIWYG1 text editor. This is deliberate, allowing the programmer to stay in full control. To alleviate the editing process, a large number of features are at your disposal. For inserting markup and code, there are toolbars, dialogs, and predefined/user-customized menus. Syntax highlighting, advanced search/replace functionality, scalability and language function references make Bluefish a powerful tool for development.

History of Bluefish

left for someone else to write ;-) - christian
also left for someone else to write :o(- scott

¹What You See Is What You Get

Several years have passed since the first Bluefish release. Since that time, the fish has gained a reputation as an excellent editor, with qualities like stability, usability and numerous features. Also, Bluefish is small and needs only GTK2 to function, making it usable even on slow machines.

Main Features of Bluefish

This list will give you an overview of the most important or outstanding features found in Bluefish.

- A What You Write Is What You Get interface
- Multiple document interface, will easily open 500+ documents (tested 3500 with documents simultaneously).
- Customizable syntax highlighting based on Perl compatible regular expressions, with subpattern support. Default patterns are included for
 - C
 - cfml
 - ColdFusion
 - Gettext po
 - HTML
 - Java
 - Pascal
 - Perl
 - PHP
 - Python
 - R
 - XML
- Anti-aliased text window
- Multiple encodings support, can convert between different character sets, supports multibyte characters, unicode, UTF8 etc.
- Nice wizards for startup, tables, frames, and others
- Dialogs for many HTML tags, with all their attributes
- HTML toolbar and tearable menus
- User-customizable toolbar for quick access to often used functions
- Open files based on filename patterns and/or content
- Fully featured image insert dialog

- Thumbnail creation and automatically linking of the thumbnail with the original image
- Multi thumbnail generation for easy creation of photo albums or screenshot pages
- Line numbers along the document
- A custom menu, specify your own tags or sets of code, and define your own dialogs
- Custom search and replace pattern support for the Custom menu
- Very powerful search and replace, allowing POSIX and Perl Compatible regular expressions and sub-pattern replacing
- Excellent undo/redo functionality
- Configurable recent documents and recent directories functionality
- Translations in *da fr es it hu pl no ru sv*
- User customizable integration with many programs, including weblint, tidy, make, javac etc.
- XML based function reference. Currently, references are included for HTML and PHP. A GTK reference is available, and support for Perl and Python will be added. You may also create your own function reference. The XML format is described later in the manual.

As Bluefish is a part of a larger desktop environment, we've focused on making the GUI consistent with the Gnome HIG 2 . However, we prefer not following it in every detail, as some parts are intended *for the end user* , while Bluefish is *for the programmer* .

Some features from v0.7 (GTK1) are not yet implemented. The main missing piece is project management, which will be implemented before v1.0. If you depend on this feature, v0.7 may still be the version of choice.

How Stable is Bluefish?

Quite stable! The Bluefish developers aim to produce code that neither crashes nor leaks memory. Of course, that's not always easy to do. Leaks and crashes are often fixed in CVS as soon as they're discovered and hunted down. In addition to Bluefish's large userbase, the developers use Bluefish for their daily work. So, fixing bugs and preventing crashes is always a major priority. However, some nags still exist. One example being the issue of slightly sluggish copy/paste functions.

For an updated list of open bugs, please visit the todo-list on the Bluefish Wiki, at <http://bfwiki.tellefsen.net/?pagename=ToDoList> .

We appreciate any and all contributions! Please tell us if Bluefish crashes on you :-).

Contact Us

We, the Bluefish development team, welcome all comments, user requests, constructive criticisms, and contributions. Are you curious or seeking information regarding Bluefish? Would you like to contribute by translating Bluefish or its manual? Here are your options:

- <http://bluefish.openoffice.nl/> - The main website where you'll find news, updates and more information.

- <http://bfwiki.tellefsen.net/> - The Bluefish WiKi is the notebook for the developers, containing a lot of information. This includes, but is not limited to: updated project roadmaps, status of translations, feature requests, and open bugs.
- You can subscribe to the Bluefish mailinglist by sending an email containing “ subscribe bluefish-dev ” to < bluefish-dev-request@lists.ems.ru > .
- Do you want to help translate Bluefish? Please let us know by dropping an email to Walter Echarri < wecharri(at)infovia.com.ar > , our friendly translation maintainer.
- If you've got a general question, you can also drop an email to < bluefish(at)bluefish.openoffice.nl > .

Chapter 2. bluefish installation

about different methods to have bluefish installed (distribution or own installation)

requirements

Bluefish aims to be portable, hopefully wherever GTK is ported. A comparatively small set of external libraries are necessary for it to work. Any recent GNU/Linux distribution or other *NIX with GTK2 installed, will be sufficient. In addition to the list of requirements below, you may also want to look at the section called “system specific notes”. Note: These requirements fit the GTK2-version. If you only have GTK1, you want the last GTK1-version, v0.7.

The main requirements:

- gtk v2.0
- glib v2.0
- pango 1.0

Optional requirements:

- libpcre3 - *for Perl Regular Expression support*
- libaspell - *spell checker*
- grep & find - *used by the advanced open dialog.*
Remember to add link to the advanced open description

Compiling Bluefish require a few additional packages. (Do remember that binary packages exists for many platforms. It is likely you won't need to compile ;-). Now, lets assume you want to compile, perhaps to get the latest and greatest from CVS. The requirements are as follows:

- Development files (header files etc) for the packages above. *These are often distributed as separate packages. There is also a high probability you have these installed already.*
- gcc - *Bluefish has been tested to compile on the 2.95 and 3.x branches.*
- gmake or BSD make
- autoconf - *only if you are going to compile from CVS*

quick standard installation

There are two main methods for installing Bluefish: Compile from source or install a binary package. The last one is the easiest to do, so we'll cover that first here. ;-) There are a few different approaches, caused by the differences between systems. We'll start off by summarizing a few really quick and simple approaches before dealing with this problem more extensively.

- Debian: `su - && apt-get update && apt-get install bluefish`
- Red Hat, Mandrake (and other Linux distributions that supports rpm): Download the latest .rpm from the bluefish website, <http://bluefish.openoffice.nl/>
- FreeBSD, NetBSD and OpenBSD distribute Bluefish through their packaging systems.
- To compile, or install on another platform, see the section called “installing a bluefish source distribution”.

general installation issues

A few considerations before installing may be necessary. The latest GTK1-version is v0.7, which is no longer updated. Unless you really want the project management feature or only have GTK1 installed, you should use the latest GTK2-version. A lot of bugs have been corrected, along with updates and new features.

If you want the latest and greatest, read about CVS in the section called “installing a bluefish source distribution” below. If you simply want to use Bluefish, get the latest stable package that fit your system.

how to get bluefish

Bluefish is available through several different channels. Many Linux distributions ship a version of Bluefish or make it available through their package systems. I.e., Bluefish is available through the Debian apt-system and FreeBSD's ports. You may check if Bluefish is available through your favourite software installer.

The main source is the Bluefish website, where the software and a few contributions are available. The download page is reachable at <http://bluefish.openoffice.nl/download.html>. Here, you may download the source code and binary packages for Debian, Red Hat and Mandrake.

operating systems supported by bluefish

Bluefish has been reported to work on a number of systems. The Bluefish team mainly support these platforms:

- Mandrake Linux
- Red Hat Linux
- Debian Linux
- FreeBSD

Actually, any GNU/Linux distribution with GTK2 is fine -- many distributions also include Bluefish. Actually, the fish will likely work nicely on any POSIX compatible OS where GTK2 is available. Bluefish has been reported to work on

- NetBSD - distributed in pkgsrc
- OpenBSD - available through their ports system
- SGI IRIX - see <http://freeware.sgi.com/>

- Mac OS X
- Sun Solaris
- Tru64
- AIX
- HP-UX
- Win32-cygwin - with a few nags.

which bluefish version to use

Currently, four versions of bluefish are available:

- The GTK1-version (v0.7) is deprecated and no longer updated, but is the choice for those of you still running GTK1.
- The latest GTK2-version (v0.10) is the version of choice for most users. It's the latest addition to the Bluefish tree, and is regarded as stable enough for daily use.
- The latest development snapshot is always one step further than the latest stable release. You'll find some new features, bugs fixed and a prettier gui. The catch is that it may have unfinished, perhaps buggy, features. Try this if you want to see new features, or if there's a bug in the latest stable release that bothers you.
- CVS - the bleeding edge of Bluefish development. You want this if there's a bug that bothers you, but is unfixed even in the latest snapshot, if you want to contribute a patch, or simply want the bleeding edge feature set. The CVS may be unusable for short periods of time, but will often be stable enough for daily use.

As commented in the section called "how and when updates are released", the long time between stable releases make the CVS snapshots and current CVS an interesting choice.

how and when updates are released

the development is always a deep black hole, but sometimes there is an new release :-)

-- Danny Reeh (Bluefish development team)

A long time pass between each release. The Bluefish development is not too rapid, there's a small number of developers, and we occasionally want to *use* Bluefish, not only develop it. :-)

Because of the long periods of time between releases, the current CVS or CVS snapshots may be what you want to use. Bugs will be fixed and new features introduced. We do try to keep the CVS version usable at any time (actually, the CVS version is used by most of the development team, daily).

system specific notes

Different systems have different approaches to solutions and packaging. You might find the information below interesting.

Mandrake:

- libpcre: Breaks pcre into 3 different pieces, make sure pcre-devel is installed if compiling from source. Try this command:

```
$ rpm -ql pcre-devel
```

- ... more nags with Mandrake?

Debian:

- If you are running *unstable*, you may experience instability (ha ha) with Bluefish. Make sure it's not just caused by a buggy version of GTK2. If the bug is very obscure and strange, GTK may be the villain.
- ... more nags with Debian? Of course not...? -- Christian

installing a bluefish source distribution

By installing Bluefish from source, you may be able to get a newer version (CVS Snapshot, perhaps) than those distributed as binaries. You may also need to compile from source if there is no available binary for your system.

quick installation overview

This is the short installation description. Consult the other chapters if you are in doubt.

Bluefish installed using the standard 'configure, make, make install' steps. Assuming you have downloaded a bluefish source package, for instance `bluefish-ver.tar.gz` (naturally, change the filename to what's appropriate), you complete the installation with the following steps:

1. `tar -zxvf bluefish-ver.tar.gz`
2. `cd bluefish-ver`
3. `./configure`
4. `make`
5. `su -c 'make install'`
6. Now, type `bluefish` to run. You may delete the `bluefish-ver` dir.

typical configure options

The `configure` script is used to automatically find the appropriate settings for your system. Because of differences between systems, this compile-time configuration is necessary, and `configure` solves this challenge easily -- with an added bonus of telling whether you have everything you need to compile.

The `configure-script` can be configured, specifying locations of shared libraries or similar. This is

something you most likely won't need to do, but it's easy to do if necessary. For a complete list of configure options, type

```
$ ./configure --help
```

The Bluefish-specific configure options are

- `--disable-splash-screen` - simply disables the splash-screen.
- ...

You might also need these:

- `--prefix=/usr` - If you want to install in `/usr`. Usually, `/usr/local` is preferred.
- ...

installing from development source tree

So, you want the bleeding edge? This guide will help you download and install the latest source from our CVS repository.

CVS3, a version control system, is a much used software development tool. It keeps track of all changes to the sourcecode, who did what and why they did it. If you want to read more about CVS, have a look at the CVS-book by Karl Fogel, available at <http://cvsbook.red-bean.com/cvsbook.html>.

The Bluefish project's CVS is generously hosted by SourceForge.net4. For more information about them, see their site. The project homepage is <http://sourceforge.net/projects/bluefish>. Our CVS repository contains the current Bluefish sourcecode, and even this manual. The repository is accessible by anyone, and is updated almost daily by the developers.

To access the repository, you need a few small utilities. You may have to install them. They are likely to be available through your favourite source of software (ports, apt etc). The above mentioned CVS-book is a great source for information.

The first step is to `cd` to the directory you want to put the sources in, and then log in. The latter is accomplished with this command:

```
$ cvs -d :pserver:anonymous@cvs.bluefish.sourceforge.net:/cvsroot/bluefish login
```

(just hit Enter at the password prompt) The next step is to *check out* the sourcecode. You can check out single modules, and the one you want is called `bluefish-gtk2`. The command is:

```
$ cvs -z3 -d :pserver:anonymous@cvs.bluefish.sourceforge.net:/cvsroot/bluefish co
```

A lot of files will be downloaded, and listed one by one. If you're on dialup, this might take a bit of time. Be patient ;-)

When the download has completed, you'll find the bluefish sources in the subdir `bluefish-gtk2`. Enter that directory now:

```
$ cd bluefish-gtk
```

Next, you need to generate the configure script. That's accomplished by running

```
$ autoconf
```

Then, you run `configure` with whatever options you might want. This example will cause `make install` to install Bluefish with the specified directory as prefix (I.e. the binary is installed in `/usr/local/bf-cvs/bin/bluefish`). This is most likely not what you want -- just run `configure` without parameters instead.

```
$ ./configure --prefix=/usr/local/bf-cvs
```

If `configure` fails, it'll probably give a hint telling you what's missing or wrong. Assuming it completed successfully, your next step is to compile this thing. Run

```
$ make
```

to accomplish this. When `make` has completed, you can install Bluefish: (`su` to root first, unless you specified a user writeable prefix to `configure`)

```
# make install
```

To update the sources at a later time, you run the command `cvs -z3 -q update` from within the `bluefish-gtk2` directory.

problems compiling?

Can't compile? Well, your first step is to make sure you have the necessary utilities and libraries. See the section called "requirements". Next, see if your system is mentioned in the section called "system specific notes". Below is list of well known problems that's been mentioned on the `bluefish-dev` list. If you're unable to find a solution (or if you think you have a solution others might want), feel free to contact us on the `bluefish-dev` list (See the section called "Contact Us").

- ```
make: *** No targets specified and no makefile found. Stop.
```

This will happen if `configure` fail and you try to run `make`. It also happens if you're running `make` from the wrong directory.
- ... more trouble to come ;-)

## installing a binary distribution

Different packages -- different installation. We'll cover only a few approaches here<sup>5</sup>, since the installation is very system-specific ;-). Lets have a look at some different systems:

For Debian users this is very simple. To download, install and configure bluefish in One Swift Move, run

```
$ su - && apt-get update && apt-get install bluefish
```

You can check if the version available through apt is the latest -- see the Bluefish homepage, and com-

---

<sup>5</sup>If you want to contribute a description on how to install Bluefish on your system, just drop us a note. :-)

pare the version there with what **apt-cache show bluefish** tells you. If there's a newer version on the Bluefish site, download it and install the package like this: **dpkg -i bluefish-ver.deb**

For rpm based distributions, first check if your distribution has a recent Bluefish version. If it does not, download the rpm for your distribution from any of the Bluefish mirrors. Next, installing a downloaded rpm is as simple as pointing and clicking in your favourite gui package manager, or issuing the following command from the command line: (as root)

```
rpm -Uvh bluefish-ver.rpm
```

If you're using FreeBSD, NetBSD or OpenBSD, we probably won't need to tell *you* how to use your favourite package system. ;-)

## post-installation setup

The first time you run Bluefish it will create a directory `~/.bluefish` where all configuration is stored. This includes all preferences, customized menus, highlighting-patterns, file history etc.

Bluefish will work right out of the box, but you can and should take advantage of the many customizations you are able to do. Change the font in the main textview if you don't like it, remove unused toolbars, add shortcuts to the customizable menu and edit the list of browsers and external programs.

If you are upgrading from a previous version, perhaps CVS, you should note that the syntax highlighting may have changed. To make sure you have the latest highlighting patterns, exit Bluefish and delete the `highlighting` file in your `~/.bluefish` directory. Next time Bluefish is started, the new defaults will be loaded. Note that this will also annihilate all your changes to the highlighting. A more gentle approach may be to move your current highlighting-file to `highlighting.old`, start Bluefish to get the new patterns, exit bluefish, and then run **diff -c highlighting.old highlighting** to find the differences..

If your settings should happen to be corrupted, for some reason unusable, or you simply want to revert to the defaults, you may safely delete the `~/.bluefish` directory.

---

# Chapter 3. tutorial introduction

generals about using the tutorial

## starting bluefish

generals about starting bluefish

## using bluefish, an overview

overview of the aspects in using bluefish

---

# Chapter 4. using bluefish

more concret thing

## the menu

- File
- Edit
- Tags
- Dialogs
- External
- Document
- Options

## open a file

how you can open files

## saving files

how you can save files

## closing a file

how you can close files

## quiting bluefish

how you can quit bluefish

## changing shortcut keys

Many menu entries are also accessible by a key combination, we call that a shortcut. For example pressing the ctrl+s keys together makes Bluefish save the current file to disk. You can see the shortcut key combination on the right of the menu entry.

What many people do not know is that they can be changed. Move the mouse over a meny entry, and press the key combination you would like to use. Immediately this combination will show up on the right of the menu entry. An entry can also be removed, press the backspace key when you move the mouse over a menu entry to remove the shortcut.

To save the shortcut key combinations for later Bluefish sessions you can store them. In the Options menu choose Save shortcut keys. This will store the settings in file `~/.bluefish/menudump_2`. If you want to restore the default combinations simply remove this file and restart Bluefish.

## the toolbar

a little toolbox

## the file editing area

here you are the most time

## editing text

some things in editing text

## statusbar

infos are here

## quickbar

The quickbar is a user defined toolbar. All HTML toolbar buttons can be added to the quickbar by simply right-clicking them and selecting "Add to quickbar". If you want to remove items from the Quickbar, right-click them and select "Remove from Quickbar"

## custom menu

here you can "bookmark" things

## what is the custom menu?

The custom menu allows you to add "often used" strings or search and replace patterns to a menu. Upon install Bluefish will create some default entries, these will give you an idea what can be done with the custom menu.

## editing the custom menu

changing and editing entries

## syntax highlighting

Syntax highlighting is the coloring of words that have a special meaning in the language you are writing. Obviously the patterns are different for every language. The "<title>" word for example means "start of title" in HTML, the "function" word means "start of function" in PHP.

## using syntax highlighting

why using syntax highlighting?

## modifying highlight patterns

The highlight patterns are build from perl compatible regular expressions. A pattern has options for coloring and style of the text it matches. Within a match other patterns can be used to color parts of that

match. There are three types of patterns:

- 1 Two patterns, match from the start to the end pattern
- 2 One pattern that matches from start to end
- 3 Match a subpattern from the parent pattern

One specific pattern can also be used within several other parent patterns. The parent-match option is a regular expression that defines all parents for a certain pattern. If empty it will default to `^top$`, so basically it will be on the top level.

So how does it work? Let's take a look at a little example text, a piece of PHP code within some HTML code:

```
<p align="center">
<?php
// this is a comment ?>
?>
```

The first thing the highlighting engine does is finding the pattern that has the lowest match. Using the default patterns for PHP, the pattern named `html` has a match at position 0:

```
<p align="center">
```

So now the highlighting engine searches for the lowest match in all subpatterns of `html`, in the region matched by the type 2 `html` pattern. Again, the lowest match will count. The pattern named `html-tag` has a match at position 1. This pattern is a type 3 pattern, so it matches a subpattern of the parent:

```
p
```

the match from subpattern `html-tag` ends at position 2 and it does not have any child patterns, so the highlighting engine continues at position 2 with all subpatterns from `html`. A type 2 pattern named `html-attr` has the lowest match:

```
align="center"
```

This pattern does have a child pattern, again a type 3 pattern called `html-attr-sub2` matching:

```
"center"
```

The pattern `html-attr-sub2` does not have any child patterns, and subpatterns of `html-attr` do not have any more matches, and also `html` subpatterns do not have any more matches. So we are back on the main level, the remaining code to highlight is:

```
<?php
// this is a comment ?>
?>
```

Now a pattern named `php` has the lowest match. This is a type 0 pattern, so the highlight engine continues with all the remaining code, but it will not only search for the lowest match of the child patterns of `php`, but it will also use for the end pattern of `php`. The lowest match in this example is a pattern named `php-comment-C++`. As you can see the `?>` within the comment does not end the `php` pattern, because it lies within a subpattern of `php`:

```
// this is a comment ?>
```

The pattern `php-comment-C++` does not have any child patterns, so the remaining code for the `php` subpatterns is:

```
?>
```

It is very obvious now, the lowest match will be the end pattern of the `php` pattern, so we're back on the main level, and we have matched all of the code!

The config file for highlighting is a colon separated array with the following content:

```
mode:
patternname:
case_sensitive(0-on/1-off):
start reg-ex:
end reg-ex:
start & end pattern(1), only start(2), subpattern(3):
parent-match:
foreground-color:
background-color:
don't change weight(0), non-bold(1), bold(2):
don't change style(0), non-italic(1), italic(2):
```

## bluefish preferences

see in the menu Options/Preferences

## editor preferences

editor

## html preferences

html

## files preferences

files

## user interface preferences

user interface

## **images preferences**

images

## **file types preferences**

file types

## **syntax highlighting preferences**

syntax highlighting, also see above

## **external programs preferences**

integrating external commands and browsers If you want to use for example a sed command as a filter, you can add it like this to the external commands and filters (in the preferences dialog): sed -e 'some sed command' > '%f' < '%s'

---

# Chapter 5. advanced concepts in using and customizing bluefish

for people using bluefish intensively

## special things in working with programming languages

support for special programming languages

### HTML

HTML is obviously the most supported language in Bluefish. There is a special HTML toolbar with many dialogs, and several menu sections to work with tags. You can also right-click a tag and bring up the dialog from there.

In the reference tree on the left panel there is also HTML reference available.

There are several special search and replace actions in the menu Edit-Replace special. These can be used to convert special characters (like < and &), or iso characters to their HTML entities.

### PHP

PHP

### ... and some other languages

and other supported languages. maybe also in extra subsections

### filebrowser (filtering, filetypes, etc.)

the left panel

### search and replace (regex patterns)

with the search and replace you can do incredible things. We'll start with a simple example. In some HTML table we have several table data tags where we actually want table header tags. Table data is `<td></td>` and we want `<th></th>`.

we can do two normal replaces: one where we replace `<td>` with `<th>` and then another where we replace `</td>` with `</th>`

we can also do one replace using regular expressions: find `(<|</)?td>` and replace with `\0th>` For more information about regular expressions you might want to read man 7 regex, or read any of the great internet sites about regular expressions. The `\0` in the replace string refers to the first subpattern match in the search pattern, the `\1` to the second etc.

if you understand the above example, you will realize that you can do much more. Suppose you also want to match a table data tag that does have some attributes like `<td class="myclass">`, and you want to

keep the option while converting to table header. The following pattern will do this: find `(<|</>td[^>]*>` and replace with `\0th\1>`

if you have any search and replace patterns you use often, you can also add them to the Custom Menu. Check the Custom Menu section of this manual for more information.

## project management

The projects are a sort of 'saved state' of Bluefish. All files open when the project is saved, are automatically opened the next time you open the project. Also the recently used files in that project are shown in the recent menu. Furthermore, a basedir can be set, so the filebrowser in the left panel will only show the files in the basedir and its subdirectories. If Bluefish is installed with `gnome-vfs` support the basedir might be remote, for example `sftp://someserver/somedir` or `smb://user:password@server/someshare/`.

If the webdir is entered in the project settings, Bluefish will launch the browser to the appropriate URL. If your basedir for example is `/var/www/` and your webdir `http://localhost/` Bluefish will use this information to launch the browser to the correct URL. This can be very convenient for testing server side scripting languages like PHP, JSP or other.

The template field can point to a template file. If the 'new file' button is clicked, the contents of this file will be automatically loaded into the edit window.

The projects will be expanded to have more Bluefish settings, so a project can be a bit of customized Bluefish setup. Currently the state of various toolbars and menubars is saved in a project file.

The project file itself is simply a textfile in the standard Bluefish format (same format as the config file). This format is 'key: value'.

## more..

more..

---

# Chapter 6. reference

... list all options in the preferences and their config file and config-name

---

# Chapter 7. development guidelines

what a developer should do and what not

## debugging bluefish

finding bugs

## indenting and formating style

your text is our text

## naming

why others should find your functions

## declaring procedures

naming and so

## header files

place something in headers

## new files

about new files

## file reference

references

## patches

if there are code, so there are bugs too. patches.

## CVS

what cvs is

## usage

use it

## guidelines

what to do

## **translations**

users all over the world will use bluefish

## **some tips**

development tips

## **making releases**

what is to do

## **useful stuff**

links and so

---

# Appendix A. credits

credits

## **bluefish developer**

developer

## **supporters to bluefish**

supporters

---

# Appendix B. bluefish change history

history

**changes in release gtk2-port**

...

**changes in release gtk1-version**

...

---

# Appendix C. manual writing tutorial

history

## requirements for maintain the manual

technical things you need, if you will generate html out of xml

## conventions for writting in this manual

good taste in writing manuals :-)

---

# Appendix D. GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 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 licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program 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. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have

made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public 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.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these

terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.