The **EMBOSS** Administrator's Guide

David Martin, EMBnet Norway
Peter Rice, LION Bioscience
Alan Bleasby, HGMP (EMBnet UK)

This guide relates to **EMBOSS** 2.5.0

[1] `http://www.gnu.org/copyleft/fdl.html`

Contents

1 Introduction

1.1 About this document

This guide has been written to assist system administrators and developers with the installation and configuration of **EMBOSS**. If you are reading this to find out how to do bioinformatics then you are wasting your time. You are referred instead to the Resources chapter below where there is a list of more relevant literature and web sites. Experienced users may find this document useful for configuring their own databases and customising their **EMBOSS** experience.

1.1.1 Credits

The original author of this guide was David Martin[1] at the Norwegian EMBnet node.[2] It is however the result of a team effort. Thanks are due in particular to Johann Visagie for the FreeBSD information. Other contributors are acknowledged in the text.

1.1.2 Reproduction

The obligatory bit of legalese. The first version of this guide was not in the public domain but has been released under the GNU Free Documentation License by the original author.

Although 'Free' in this license is usually explained as 'free as in freedom, not as in beer' the authors are likely to appreciate offers of free drinks should you ever meet them.

1.2 What is EMBOSS?

EMBOSS is a freely available suite of bioinformatics applications and libraries. It can be downloaded via the internet, copied, customised, and passed on under the terms of the various General Public Licenses. **EMBOSS** has been developed in response to the need for a powerful, adaptable suite of software that can interface readily with many different situations and meet the need of professional bioinformaticists, particularly those needing high throughput and/or scriptable capabilities.

EMBOSS has primarily been developed by those responsible for the public extensions to the GCG package. **EMBOSS** supercedes much of EGCG and includes far better database interaction. **EMBOSS** also has the benefit of freely accessible source code so novel applications can be developed rapidly and at minimal cost.

EMBOSS is currently only available for Unix/Linux systems but it has been known to compile and run on Windows NT. This document will only consider the UNIX version and will assume the reader has some familiarity with UNIX system administration.

1.2.1 Where to get it?

EMBOSS is available for download from the primary site at the UK EMBnet node by anonymous ftp.[3] This directory contains the **EMBOSS** package and several associated packages (collectively known as EMBASSY) that are distributed with **EMBOSS**. Download these to a suitable location. Documentation is available on the WWW at the **EMBOSS** web site.[4]

[1] `damartin@hgmp.mrc.ac.uk`
[2] `http://www.no.embnet.org`
[3] `ftp://ftp.uk.embnet.org/pub/EMBOSS/`
[4] `http://www.uk.embnet.org/Software/EMBOSS`

FreeBSD distributions from 4.2 onwards now include **EMBOSS** as an optional package maintained by Johann Visagie.[5] Please see section 2.5 for more information on installation on FreeBSD.

[5] `johann@egenetics.com`

2 Installation

2.1 Retrieving **EMBOSS** by anonymous ftp

2.1.1 Interactive FTP

Change directory to the location in which you wish to download the **EMBOSS** source code. In this example we will download the source to */packages/EMBOSS*. Then start your ftp client and point it to ftp.uk.embnet.org.

```
% ftp ftp.uk.embnet.org
Connected to tantalum.hgmp.mrc.ac.uk.
220 tantalum FTP server ready.
Name (ftp.uk.embnet.org:embuser):
```

We are using anonymous FTP so type the username `anonymous`.

```
Name (ftp.uk.embnet.org:embuser): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
```

Enter your email address here as the password for user *anonymous*.

```
Password:
230-#################################################################
230-
230-  Welcome to the UK HGMP Resource Centre anonymous ftp service
230-
230-       Please contact support@hgmp.mrc.ac.uk regarding
230-                any problems with this service
230-
230-#################################################################
230-
230-Please read the file README
230-  it was last modified on Wed Aug 13 15:40:25 1997 - 1810 days ago
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Move to the **EMBOSS** directory and list the files. The output has been truncated a little to save space.

```
ftp> cd /pub/EMBOSS
ftp> ls
200 PORT command successful.
150 Opening BINARY mode data connection for /bin/ls.
total 22334
...     1024 May 26 20:17 .gnu
...  9079913 May 14 21:37 EMBOSS-2.5.0.tar.gz
```

```
...       19 May 14 21:37 EMBOSS-latest.tar.gz -> EMBOSS-2.5.0.tar.gz
...   196872 May 12 18:49 EMNU-1.0.5.tar.gz
...   231485 May 15 13:55 ESIM4-1.0.0.tar.gz
...   405620 May 12 18:49 HMMER-2.1.1.tar.gz
...     1024 Jul 25 08:54 Jemboss
...   264189 May 12 18:49 MEME-2.3.1.tar.gz
...   251061 Jul  9 19:01 MSE-0.0.4.tar.gz
...   694450 May 12 18:49 PHYLIP-3.573c.tar.gz
...   200490 May 12 18:49 TOPO-0.1.tar.gz
...     1536 Jul  9 19:01 old
...      512 Jun 27 14:40 patchfiles
...      512 Feb 22 15:19 tutorials
226 Transfer complete.
ftp>
```

Now download the source files

```
ftp> get EMBOSS-latest.tar.gz
200 PORT command successful.
150 Opening BINARY mode data connection for EMBOSS-latest.tar.gz
(9079913 bytes).
...
ftp>
```

And repeat for each file. Or use `mget *gz` to download all the files at once. Exit your ftp session with the command `bye`.

2.1.2 FTP using WGET

The program WGET can be used to download a remote directory noninteractively. More details on WGET can be obtained from the Free Software Foundation.[1] Assuming you have WGET installed, use the following command which generates a lot of output on the screen:

```
% wget -m 'ftp://ftp.uk.embnet.org/pub/EMBOSS'
--15:04:41--  ftp://ftp.uk.embnet.org:21/pub/EMBOSS
           => `ftp.uk.embnet.org/pub/.listing'
Connecting to ftp.uk.embnet.org:21... connected!
Logging in as anonymous ... Logged in!
==> TYPE I ... done.  ==> CWD pub ... done.
==> PORT ... done.    ==> LIST ... done.

...
many pages truncated
...

FINISHED --15:04:55--
Downloaded: 2,657,366 bytes in 4 files
```

A new directory *ftp.uk.embnet.org* has been created and EMBOSS can be found at *ftp.uk.embnet.org/pub/EMBOSS*. You may wish to create a symbolic link to this from your */packages* directory for convenience.

[1] http://www.gnu.org

2.2 Unpacking

You will have downloaded the **EMBOSS** and EMBASSY packages to a suitable directory. For this example we will assume you have downloaded them to */packages* so you should now have the following files (or similar) and maybe more packages in EMBASSY.

```
% ls
EMBOSS-latest.tar.gz
EMNU-1.0.5.tar.gz
ESIM4-1.0.0.tar.gz
HMMER-2.1.1.tar.gz
MEME-2.3.1.tar.gz
MSE-0.0.4.tar.gz
PHYLIP-3.573c.tar.gz
TOPO-0.1.tar.gz
```

First unpack the **EMBOSS** distribution

```
% gunzip EMBOSS-latest.tar.gz
% tar xf EMBOSS-latest.tar
```

This will create a new directory, *EMBOSS-2.5.0* or similar. You may wish to use `tar xpf` for unpacking **EMBOSS**.

Enter the **EMBOSS** directory

```
% cd EMBOSS-2.5.0
```

create a directory for the EMBASSY packages

```
% mkdir embassy
```

Now move the EMBASSY packages to the EMBASSY directory

```
% mv ../MSE-0.0.4.tar.gz PHYLIP-3.573c.tar.gz \
   TOPO-0.1.tar.gz embassy
```

Go into the EMBASSY directory and unpack those packages.

```
% cd embassy

% gunzip MSE-0.0.4.tar.gz
% tar xf MSE-0.0.4.tar
```

and so on for each EMBASSY package.

Go back up one directory to the main **EMBOSS** package directory and prepare to start compilation.

2.3 Graphics Requirements

Depending on your system you may need to explicitly configure the graphics. EMBOSS includes the plplot graphics library and will link to X11 and the recent (non-GIF) releases of the gd graphics library which also require libz and libpng (and possibly libjpeg). Please see the section 'Configuring **EMBOSS** graphics' below.

To get PLPLOT to produce PNG images you will need to have the *z*[2], *png*[3] and *gd*[4] libraries installed. *gd* version >= 1.8.4 is recommended. A recent release must be used as older versions

[2] `http://www.info-zip.org/pub/infozip/zlib/`
[3] `http://libpng.sourceforge.net/`
[4] `http://www.boutell.com/gd/`

support GIF which is NOT supported in later versions because of software patent problems. If for some reason you do not have the required libraries and your system support group will not update them for the system then install all three latest versions (*z*,*gd*,*png*) to a new directory and then add this new directory to your configure line for **EMBOSS** — `./configure --with-pngdriver=my_dir` where the *z*, *png* and *gd* libraries were each installed using `./configure --prefix=my_dir`

??? It may also be helpful to ensure that the `LD_LIBRARY_PATH` environment variable is set appropriately to include the libraries in the path. ???

GD) http://www.boutell.com/gd/ Z) http://www.mirror.ac.uk/sites/ftp.cdrom.com/pub/infozip/zlib/ PNG) http://www.mirror.ac.uk/sites/ftp.libpng.org/pub/png/libpng.html

These also list the various mirror sites for non UK people.

Alternatively, using ftp :-

GD) (boutell.com no longer allows FTP, no known mirror sites, use HTTP) Z) ftp://ftp.info-zip.org/pub/infozip/zlib/zlib-1.1.3.tar.gz PNG) ftp://swrinde.nde.swri.edu/pub/png/src/libpng.1.2.1.tar.gz You can unpack the tar.gz files in any directory, and install them in a common area.

By default everything (including EMBOSS) installs in /usr/local but in the examples below we use /home/joe/local

Note: gd does not use a ./configure script, and will fail at the "make install" stage if the installation directory does not have a /bin subdirectory. You can create this directory (e.g. /home/joe/local/bin) if it does not already exist.

2.3.1 zlib

Zlib is avilable from these sites:

http://www.mirror.ac.uk/sites/ftp.cdrom.com/pub/infozip/zlib/ [5] *http://www.info-zip.org/pub/infozip/zlib/* [6] *ftp://ftp.info-zip.org/pub/infozip/zlib/zlib-1.1.3.tar.gz* [7]

To install, pick up the sources and then:

```
% gunzip -c zlib-1_1_3_tar.gz   | tar xf -
% ln -s zlib-1.1.3   zlib
%  cd zlib
%  ./configure --prefix=/home/joe/local
%  make
%  make install
%  cd ..
```

2.3.2 libpng

Libpng is avilable from these sites:

[8] [9] [10]

To install, pick up the sources and then:

```
% gunzip -c libpng-1_2_1_tar.gz | tar xf -
%   ln -s libpng-1.2.1 libpng
%   cd libpng
%   cp scripts/makefile.linux makefile
```

Libpng has no configure script so you have to do some work by hand. Edit makefile, change prefix to be /home/joe/local and any other places - some files point to ../zlib others use /usr/local/lib and /usr/local/include. On HP-UX this is trickier. CFLAGS has to match the definition for zlib.

Now build using the edited makefile:

[5] `http://www.mirror.ac.uk/sites/ftp.cdrom.com/pub/infozip/zlib/`
[6] `http://www.info-zip.org/pub/infozip/zlib/`
[7] `ftp://ftp.info-zip.org/pub/infozip/zlib/zlib-1.1.3.tar.gz`
[8] `http://libpng.sourceforge.net/`
[9] `http://www.mirror.ac.uk/sites/ftp.libpng.org/pub/png/libpng.html`
[10] `ftp://swrinde.nde.swri.edu/pub/png/src/libpng.1.2.1.tar.gz`

```
%   make
%   make install
%   cd ..
```

2.3.3 gd

Gd is available from these sites:
[11]

There is no FTP server at this site.

To install, pick up the sources, build zlib and libpng first, and then:

```
% gunzip -c gd-1.8.4.tar.gz     | tar xf -
% ln -s gd-1.8.4     gd
% cd gd
```

Now edit Makefile, change the definitions for INCLUDEDIRS, LIBDIRS, INSTALL_LIB, INSTALL_INCLUDE, INSTALL_BIN, and change all */usr/local* to */home/joe/local*

```
% make
% make install
% cd ..
```

If the gd "make install" fails with a warning about the "bin" directory, you need to create it by hand (see above).

To compile with the local version your EMBOSS configure line should now read:

```
./configure --with-pngdriver=/home/joe/local
```

This will look for the graphics libraries in your local installation under */home/joe/local* instead of a system-wide location

configure keeps a copy of the previous settings. With earlier releases of EMBOSS, or as a developer with an earlier release of autoconf, you may need to delete files *config.cache* and *config.status* if configure has been run before.

2.4 Compilation

Building **EMBOSS** is easy. It follows the usual GNU style of `./configure`, `make`, `make install`. We'll take these steps one at a time.

2.4.1 Configure

To accept the default configuration, just type `./configure` and let **EMBOSS** get on with it. You may however want to make some changes to the configuration parameters according to your local policy. This section will not cover all the possibilities, just some of the more common. The configuration script will attempt to find the necessary components in your system to determine how to successfully build **EMBOSS**. It typically expects the GNU C compiler (gcc) and several standard libraries that should already be part of your Unix/Linux system. **EMBOSS** should configure, compile and run on most modern Linux distributions straight out of the box.

[11] http://www.boutell.com/gd/

Installation directory

You need to have write permission on the directory in which you eventually wish to install **EMBOSS**. You may also wish to put it somewhere else other than the standard location of */usr/local/emboss*.

The installation directory is controlled by the `--prefix` argument. For example, you can have all third party applications owned by a non-privileged user and installed in a package specific directory under */site/prog*

```
% ./configure --prefix=/site/prog/emboss
```

will install **EMBOSS** under */site/prog/emboss*. The binaries will be installed in */site/prog/emboss/bin* with shared libraries installed in */site/prog/emboss/lib*. System wide data are installed in */site/prog/emboss/share/EMBOSS/data*, and the configuration files (ACD files) for the applications will be installed in */site/prog/emboss/share/EMBOSS/acd* (or for EMBASSY in directories corresponding to the package name.) Documentation is installed in */site/prog/emboss/share/EMBOSS/doc*. The installation directory should be specified using a full path otherwise interesting failures may occur.

The individual directories for installation can be modified with other configuration commands but this is usually not necessary. Run `./configure --help` to get more information on the directories that can be changed and other configuration options.

Run `./configure` with the options you wish to use. This may take a short time as various messages scroll up the screen.

All should be well with this and configure should exit with a message like this:

```
... much output skipped

creating ./config.status
creating plplot/Makefile
creating plplot/lib/Makefile
creating nucleus/Makefile
creating ajax/Makefile
creating emboss/Makefile
creating emboss/acd/Makefile
creating test/Makefile
creating test/data/Makefile
creating test/embl/Makefile
creating test/pir/Makefile
creating test/swiss/Makefile
creating test/swnew/Makefile
creating test/wormpep/Makefile
creating emboss/data/Makefile
creating emboss/data/AAINDEX/Makefile
creating emboss/data/CODONS/Makefile
creating emboss/data/REBASE/Makefile
creating emboss/data/PRINTS/Makefile
creating emboss/data/PROSITE/Makefile
creating Makefile
```

Configuration is now complete.

Reconfiguration

If at first you don't succeed, try, try and try again. It is not uncommon to make typos or other mistakes when running `./configure`. If you want to run configure again you should run `make clean` before running `./configure` with (hopefully) the correct options. With an earlier EMBOSS release, or as a developer with an earlier release of autoconf, you must first delete the file *config.cache* but this is no longer produced.

Configuring EMBOSS graphics

The PLPLOT library can produce output to many devices but requires certain libraries that are NOT distributed with **EMBOSS**

To get X-windows based output you must have X installed, or else PLplot will not build the required driver. You may need to specify the location of your X-windows library with the configuration options: `--x-includes=DIR` (X include files are in DIR) `--x-libraries=DIR` (X library files are in DIR)

To explicitly configure PLPLOT without X-windows, use `--without-x`.

You can explicitly tell **EMBOSS** to not include PNG support with `--without-pngdriver`.

You can tell if `./configure` has found a suitable PNG library by watching for something like the following when running `./configure`:

```
checking if png driver is wanted... yes
checking for inflateEnd in -lz... (cached) yes
checking for png_destroy_read_struct in -lpng... (cached) yes
checking for gdImageCreateFromPng in -lgd... (cached) yes
```

This means that the configuration script has located the PNG libraries on your system. If you see a message indicating that `./configure` could not find the libraries or that the version of *gd* was too old then you should install the latest versions of the libraries yourself and rerun configure with the correct `--with-pngdriver` value.

When you run an EMBOSS graphical application you can see the list of installed graph devices by giving '?' as the response to the 'Graph type' prompt.

2.4.2 Configuring for 64 bit systems

EMBOSS configure looks for GCC and uses this of preference when compiling **EMBOSS**.This is not ideal for those who wish to have a compiled and linked 64bit version of **EMBOSS**.The current version is NOT 64 bit clean (ie. it does not necessarily use 64 bit representation internally) but will compile and run quite happily on 64 bit systems.

Additional notes are appended below for the various operating systems we have information on.

IRIX 6.5.10

In order to compile for 64 bit on IRIX you have to specify the native compiler in 64 bit mode (`cc -64`) and the linker in 64 bit mode (`/bin/ld -64`). The following notes were provided by Jose Ramon Valverde[12].

We have succeeded in compiling EMBOSS for IRIX using 64 bit compilation.

It required some tweaking, but works. The recipe for those willing to give it a try is:

- remove '*gcc*' from your path
- define *COMPILER_DEFAULTS_PATH* appropriately (see *pe_environ*) to look for a *compiler.defaults* file containing e.g. `:abi=64:isa=4:proc=r10k`
- `./configure` in **EMBOSS** and all EMBASSY subdirs
- search in all files for '`CC = cc`' and substitute it for '`CC = cc -64`'
- same for '`LD = /bin/ld`' to '`LD = /bin/ld -64`'
- `make`

[12] jrvalverde@cnb.uam.es

The reason is that compiling depends on the Makefile and on libtool, as well as linking. We didn't spend much in looking at configure since the above steps where so straightforward. We know we should look into the configure script and add an option for 64-bit-irix-compile or some such, but that'll have to wait till we have time for it.

Yes, we know, the search and substitute thing looks tedious, but it isn't, honest: create a 'chfile.sh' out of the EMBOSS source hierarchy containing:

```
#/bin/sh
cp \$1 \$1.orig
mv \$1 tmpfile
sed -e 's/CC="cc"/CC="cc -64"/g' tmpfile | \
sed -e 's/CC = cc/CC = cc -64/g' | \
sed -e 's/\/bin\/ld/\/bin\/ld -64/g' \$1
rm tmpfile
## if you are sure, uncomment this
#rm \$1.orig
```

'`cd`' to the emboss directory and run

```
find . -type f -exec /path/to/chfile.sh {} \; -print
```

and you are done with the CC *changes.* LIBTOOL *requires special treatment since it uses quotes.*

2.4.3 Building EMBOSS

Building **EMBOSS** is a matter of typing '`make`' and going to find something else to do for the next ten minutes to half an hour depending on the speed of your system. **EMBOSS** will first build the shared libraries (*PL_PLOT*, *AJAX*, and *NUCLEUS*) and then build the applications.

You may see plenty of warnings (especially on SGI systems) complaining about libraries not being used to resolve any symbols. These can be safely ignored.

If all goes according to plan you should have built **EMBOSS** successfully. If not you will have to try to work out why the build failed. If you can't work it out yourself, send an email describing the problem to emboss-bug@embnet.org preferably with a copy of the output from the installation.

Assuming that compilation was successful, you can[13] now type '`make install`'. After a few minutes and many pagefuls of messages, **EMBOSS** should be installed where you specified in the `--prefix` option (or in the default location of */usr/local/emboss* if `--prefix` was not specified).

2.4.4 Post compilation setup

You will now need to make a few adjustments to your enviromnent to ensure that **EMBOSS** runs smoothly. **EMBOSS** looks for certain environment variables to determine where the libraries and data are found. These instructions assumed you installed **EMBOSS** in */site/prog/emboss*. Adjust these instructions to suit your installation. Insert the following lines at the end of */etc/cshrc* (or *~/.cshrc* for a personal installation)

```
setenv PLPLOT_LIB /site/prog/emboss/lib
set path=( /site/prog/emboss/bin \${path} )
```

Or for bash/ksh/sh users, insert the following at the end of */etc/profile* or *~/.bashrc*

```
PLPLOT_LIB=/site/prog/emboss/lib
PATH=/site/prog/emboss/bin:\$PATH
export PLPLOT_LIB PATH
```

EMBOSS should now be ready for use.

[13] You don't have to do this. You can leave **EMBOSS** where it is and just add the path to the *emboss* directory to your `PATH`

2.4.5 **EMBOSS** data files

EMBOSS will by default install the data files (including those installed with REBASEEXTRACT, PROSEXTRACT PRINTSEXTRACT AAINDEXEXTRACT or CUTGSEXTRACT) in the default directory *share/EMBOSS/data* in the install prefix directory. If **EMBOSS** is not installed (for example, your own personal installation) the data files are written to *emboss/data* in the directory where emboss was built.

If you want to place your data files elsewhere, or have a separate set of datafiles you wish to use, you can set the `EMBOSS_DATA` variable in *emboss.default* or, for personal use, in your *.embossrc* file.

2.4.6 Testing your **EMBOSS** installation

You can test your **EMBOSS** installation by trying the program 'wossname'

```
% wossname -auto |more
```

This should give a long list of programs that are available. Press space to page down through the list. This is just the **EMBOSS** programs and doesn't include any of the EMBASSY programs, but only because they are not yet installed. (Note: Although wossname does have a -noembassy option this does not work with installed programs because wossname can no longer find any difference between EMBOSS and EMBASSY)

2.5 Installing EMBASSY

As well as the base libraries and standard EMBOSS distribution, various extra packages (EMBASSY) are distributed with EMBOSS.

To install an EMBASSY package, go to the relevant directory. For example to install PHYLIP (which was unpacked into */packages/EMBOSS-2.5.0/embassy/PHYLIP-3.573c* earlier) go to the relevant directory.

```
% cd  /packages/EMBOSS-2.5.0/embassy/PHYLIP-3.573c
% ./configure --prefix=/site/prog/emboss
... output not shown
% make
... output not shown
% make install
... output not shown
```

Note. You **MUST** use the same arguments for `./configure` that you used for the installation of the main **EMBOSS** package. It may be necessary to add other options as required by individual packages (see below).

Repeat as necessary for the other EMBASSY packages. It should also be noted that certain EMBASSY packages may require additional libraries.

You should now find that running WOSSNAME as before lists the EMBASSY programs.

2.5.1 EMBASSY package specific notes

In most cases, EMBASSY packages should build with no problems. Known problems are described below.

Packages with no known problems

So far ESIM4, HMMER, MEME, MSE, PHYLIP and TOPO appear to install without a problem using the same arguments to `configure`.

EMNU

EMNU requires the *curses* or *ncurses* libraries that come as standard on most Unix-like systems. In particular EMNU requires two header files *form.h* and *menu.h* that are not distributed with all implementations.

If your *curses/ncurses* library is installed in a strange place then you may need to instruct `configure` with the option

```
--with-curses=/path/to/curses
```

2.6 Installing EMBOSS in package format

EMBOSS can be installed on almost all Unix/Linux operating systems using the instructions above, but the package format can be far more convenient. A package is a precompiled set of binaries with installation instructions that can be set up on your system with a minimum of work. In some cases the package will check for the correct libraries and install those as necessary.

Brief instructions are given here for the packages of which we are aware. These are maintained separately from the main source tree and may also install some files in operating system standard locations instead of the locations used by the 'raw' **EMBOSS** distribution. Please read the more detailed instructions that accompany each package.

2.6.1 Installing EMBOSS on FreeBSD

A FreeBSD **EMBOSS** package has been created by Johann Visagie[14] of Electric Genetics. This will be distributed on the installation CD's and through the normal distribution channels from FreeBSD version 4.2 onwards.

For the FreeBSD user with an up-to-date ports tree[15], installing **EMBOSS** reduces to two simple commands (as root):

```
# cd /usr/ports/biology/emboss
# make install
```

The FreeBSD specific parts of the port are that *emboss.default* is included with the other configuration files under */usr/local/etc* as *emboss.default.sample*, and the **EMBOSS** documentation is installed in */usr/local/share/doc/EMBOSS* instead of the default location. For further information on installation under FreeBSD you are referred to the Resources chapter.

[14] `johann@egenetics.com`

[15] FreeBSD users can update their ports tree through a variety of mechanisms. Please see the FreeBSD specific guide produced by Johann for more information

3 Configuration

EMBOSS can be readily configured to match your requirements. In a standard installation of **EMBOSS** the configuration directives are looked for in the following locations and in the following search order:

1. A file *emboss.default* in the *share/EMBOSS* subdirectory of your **EMBOSS** installation.[1][2]
2. A file *.embossrc* in the directory specified by the `EMBOSSRC` environment variable.
3. A file *.embossrc* in the users home directory.
4. A file *.embossrc* in the current directory.

emboss.default and *.embossrc* are plain text files that can readily be edited to suit.[3] Redefinitions of configuration parameters will override those previously defined. In the descriptions that follow only *.embossrc* will be mentioned but all directives can be placed in *emboss.default* for site wide configuration.

Several aspects of **EMBOSS** can be defined. These are:

- **EMBOSS** environment variables
- **EMBOSS** databases
- Default behaviour of **EMBOSS** programs

Databases are by far the most complex of these.

EMBOSS will ignore blank lines in the *emboss.default* and *.embossrc* files. It will also ignore any lines beginning with `#` or `!` allowing comments to illuminate the declarations in the file.

3.1 EMBOSS environment variables

EMBOSS environment variables are set with an '`env`' or a '`set`' declaration. '`env`' and '`set`' are interchangeable. The most important environment variable is the location of the *.acd* files that describe each program.

```
set emboss_acdroot /site/prog/emboss/share/EMBOSS/acd
```

Environment variables are useful for simplifying maintenance of your *.embossrc*. For example you may want to specify the location of your databases as an environment variable. Then if you move the databases you only have to update one line in the configuration file.

```
set emboss_database_dir /data/databases/flatfiles
```

This would then be referred to later in *.embossrc* as

```
\$emboss_database_dir/embl
```

for the directory */data/databases/flatfiles/embl*

[1] This location may have been redefined in installations of **EMBOSS** that have been packaged for specific operating systems. See section 2.5 for further information on OS specific package installations.

[2] **EMBOSS** will also look in the *emboss* directory under the **EMBOSS** source distribution for *emboss.default.template* and install this as *emboss.default* if no existing file is found under the installation directory

[3] A sample *emboss.default* is located in *emboss/acd* under the source distribution.

3.1.1 Configuring **EMBOSS** differently for different groups of users

It may be the case that you have users who need to share a specific setup. Maybe to have access to different sets of databases or need to use a different data directory.

It can be time consuming and error prone to maintain a series of individual *.embossrc* files or to cause users to have to work in the same directory or to copy an *.embossrc* to each directory they wish to work in. The environment variable `EMBOSSRC` can be set to point to an arbitrary directory containing an *.embossrc* which can then be used to give workgroup specific configuration. Each user then only needs to set `EMBOSSRC` in their *.cshrc* (CSH) or *.profile* (BASH) to get the workgroup specific setup.

In our case we have several groups of researchers for whom we maintain biological sequence databases. These databases have been made available under restrictive licenses so that we cannot allow researchers outside the groups to access the databases. Using `$EMBOSSRC` we can set up a common configuration for the members of each group by defining the databases in the *$EMBOSSRC/.embossrc* file.

3.2 Databases

3.2.1 Database access modes

EMBOSS offers three modes for accessing databases:

Single: **EMBOSS** retrieves a single sequence indexed by ID.

Query: **EMBOSS** retrieves a set of sequences corresponding to a query that can return more than one entry, including accession numbers or wildcard IDs.

All: **EMBOSS** returns all the sequences in the database in no particular order.

Each database definition can configure one or many of these modes for database access.

Typically **EMBOSS** uses variations on the EMBLCD system of database indexing to provide rapid access in single and query modes to flat file databases. The EMBLCD method is implemented in a variety of ways depending on the original format of your database. The EMBLCD method assumes that you have one or both of ID and accession number in each record and that they are unique for the whole database index. **EMBOSS** also provides methods for retrieving sequences via the WWW and three specific methods for interaction with SRS[4] installed localy or through a remote public server. For other non flatfile databases or flat file databases in formats not currently supported by **EMBOSS** you will have to configure an external application to retrieve sequences.

3.2.2 General database configuration.

Each database is configured using a DB declaration.

The generalised form is

```
DB databasename [

Configuration options

]
```

The configuration options are tag/value pairs and must contain at least a description of the access method (using `method:` or one or more of `methodsingle:`, `methodquery:` and `methodall:`) and a description of the original format of the sequences (using `format:`). In addition to these tags there will be other tags that are needed for particular methods and other tags that are optional.

[4] `http://www.lionbioscience.com/solutions/srs`

Database access methods

The scope of each method is:

Single mode - `s` Supports retrieval of a single sequence.

Query mode - `q` Supports retrieval of a subset of the sequences in the database specified using a wild card query in the USA[5]

All mode - `a` Supports retrieval of all sequences in the database as a stream of data.

An example entry for each access method is shown.

APP Modes: `a q s`
APP is the same as EXTERNAL.

BLAST Modes: `a q s`
BLAST uses EMBLCD indices created with DBIBLAST to access databases in BLAST format, created with NCBI's `formatdb` program.

Note that the latest 'format version 4' is not yet documented by NCBI. **EMBOSS** will only work with 'format version 3' databases, indexed with:

```
formatdb -A F
```

We hope to support 'format version 4' databases in future. If you pick up a blast database from NCBI (or elsewhere) check the format. If it is in the new format, you will need to pick up the original FASTA format file, and either index it yourself with formatdb, or run `dbifasta` and use the FASTA file in **EMBOSS** (see EMBLCD access method)

The definition should use format: ncbi because this is what the blast formatdb databases store internally.

```
DB mydb [
#required parameters
   method: "blast"
   format: "ncbi"
   type: "N"
   dir: "\$emboss_db_dir/blas"t
#optional parameters
   fields: "sv des"
   release: "63.0"
   comment: "my comment"
   indexdir: "\$emboss_db_dir/blastindices"]
```

The index files can be kept in the same directory as the database, but as each EMBLCD index needs its own directory (the filenames are fixed) the indexdir is usually defined.

The EMBLCD index files include the filenames indexed by `dbiblast`. You can use the file: and exclude: attributes to create file-specific subsets from a single `dbiblast` generated index, but as blast index files are split only by the number of entries this is not generally useful.

If the database was indexed with additional fields, they can be included in the definition as fields: to allow their use in USAs.

[5]Please see the **EMBOSS** documentation for description of Uniform Sequence Address format

DIRECT Modes: a

Direct accesses the flatfile directly. It returns all the database entries, one after the other. It assumes no indexing. Queries are still possible as **EMBOSS** will read each entry and match it against the query, but are slow as the entire database must be read.

```
DB mydb [
#required parameters
   method: "direct"
   format: "embl"
   type: "N"
   dir: "\$emboss_db_dir/mydb"
   file: "*.dat"
#optional parameters
   fields: "sv des key org"
   release: "63.0"
   comment: "My own database with no indices"
   exclude: "est*.dat"
]
```

For most cases, it is simpler to use `dbiflat` for EMBL, Genbank or SwissProt format, or `dbifasta` to index FASTA or NCBI format files, and to use the EMBLCD access method.

If the file format supports additional fields, they can be included in the definition as fields: to allow their use in USAs.

EMBLCD Modes: a q s

EMBLCD uses EMBLCD indices created with DBIFLAT or DBIFASTA to access flatfile databases in the original format.

```
DB mydb [
#required parameters
   method: "emblcd"
   format: "embl"
   type: "N"
   dir: "\$emboss_db_dir/emb"l
#optional parameters
   fields: "sv des key org"
   file: "*.dat"
   release: "63.0"
   comment: "my comment"
   exclude: "est*.dat"
   indexdir: "\$emboss_db_dir/indice"s
]
```

The EMBLCD index files include the filenames indexed by `dbiflat` or `dbifasta`. You can use the file: and exclude: attributes to create file-specific subsets from a single index.

This method can require careful setup. Please read the more specific descriptions below.

If the database was indexed with additional fields, they can be included in the definition as fields: to allow their use in USAs.

EXTERNAL Modes: a q s

EXTERNAL uses an external application to retrieve sequences. The ID is passed as an argument to the application, either replacing %s in the command string (if present) or as an additional argument (if there is no %s).

EXTERNAL requires the application to return the sequence on STDOUT. If the application writes to somewhere else, simply wrap it in a script that copies the output to STDOUT.

```
DB mydb [
#required parameters
    method: "app"
    format: "fasta"
    type: "P"
    app: "getfromdb"
#optional parameters
    comment: "my own protein database with a custom retrieval program"
    app: "getfromdb mydatabase \%s"
]
```

The first app: definition will use the default call 'getfromdb mydb:id'

The alternative app: definition will use the %s format and call 'getfromdb mydatabase id'

Both will pass either the ID or accession from the query, so that USAs mydb-id:x13776 and mydb-acc:x13776 are equivalent.

GCG Modes: `a q s`

GCG uses EMBLCD indices created with DBIGCG to access databases in GCG format. This method uses the *.ref* and *.seq* files created by the GCG suite of programs.

```
DB mygcgdb [
#required parameters
   method: "gcg"
   format: "embl"
   type: "N"
   dir: "\$emboss_db_dir/gcgembl"
#optional parameters
   fields: "sv des key org"
   file: "*.seq"
   release: "63.0"
   comment: "my comment"
   exclude: "est*"
   indexdir: "\$emboss_db_dir/indices"
]
```

The EMBLCD index files include the filenames indexed by `dbigcg`. You can use the file: and exclude: attributes to create file-specific subsets from a single `dbigcg` generated index.

SRS Modes: `a q s`

SRS returns entries from a local installation of SRS using the -e switch to getz to return entries in the original format.

```
DB mydb [
#required parameters
   method: "srs"
   format: "embl"
   type: "N"
#optional parameters
   dbalias: "embl"
   fields: "sv des key org"
   app: "getz"
   comment: "My srs indexed database"
   release: "63.0"
]
```

This access method builds an SRS commandline query to getz. If you have getz installed under another name, define this as app:

The SRS query by default uses the EMBOSS database name. If the database has a different name in SRS, define dbalias: as the database name to pass to SRS.

SRS will return the results using 'getz -e' so the format should match the format of the original data. For some formats this can be tricky (PIR for example), so consider using SRSFASTA although this will lose information that is not included in the FASTA format SRS output.

To query using the additional fields SRS supports, add them as fields:

SRSFASTA Modes: `a q s`
As SRS but returns the sequences in FASTA format. The definition must include format: fasta so that EMBOSS will read the results in FASTA format.

```
DB mydb [
#required parameters
   method: "srsfasta"
   format: "fasta"
   type: "N"
#optional parameters
   dbalias: "embl"
   fields: "sv des key org"
   app: "getz"
   comment: "My srs indexed database"
   release: "63.0"
]
```

This access method builds an SRS commandline query to getz. If you have getz installed under another name, define this as app:

The SRS query by default uses the EMBOSS database name. If the database has a different name in SRS, define dbalias: as the database name to pass to SRS.

SRS will return the results using 'getz -f -sf fasta' so the format must be 'fasta'.

To query using the additional fields SRS supports, add them as fields:

SRSWWW Modes: `a q s`
As URL, but specific to an SRS web server. This method takes a base URL (up to wgetz) for an SRS server, and builds the rest of the URL as a valid SRS query.

By building the URL, SRSWWW access can query both ID and accession number, and can query additional fields 'sv', 'des', 'key' and 'org' if they are allowed with a fields definition.

```
DB mydb [
# required parameters
    method: "srswww"
    format: "genbank"
    type: "N"
    url: "http://www.infobiogen.fr/srs5bin/cgi-bin/wgetz?"
#optional parameters
    dbalias: "genbank"
    fields: "sv des key org"
    comment: "Genbank by SRS from InfoBiogen"
    proxy: ":"
    httpversion: "1.0"
]
```

Because queries for such fields to a remote server can find a very large number of hits, and EMBOSS will load the entire output into memory to process the HTML, many EMBOSS administrators choose not to define these fields for an SRSWWW server.

If there is sufficient demand, it should be possible to rewrite the HTML preprocessing to avoid buffering in memory.

SRSWWW support the `proxy` and `httpversion` settings described under access method URL.

URL Modes: `s`
URL uses a defined web server to retrieve a specific entry. EMBOSS may fail if the HTML causes complications with parsing of the entry.

```
DB mydb [
# required parameters
   method: "url"
   format: "genbank"
   type: "N"
   url: "http://www.infobiogen.fr/srs5bin/cgi-bin/wgetz?-e+[genbank-id:%s]"
#optional parameters
   comment: "Genbank by ID from InfoBiogen"
]
```

The %s in the URL string indicates where **EMBOSS** will insert the identifier portion of the USA.

At many sites, remote HTTP access is controlled by a proxy server. EMBOSS uses a proxy server defined as EMBOSS_PROXY with a value in the format `domain.address:port`, for example:

```
set emboss_httpversion 'proxy.mydomain.org:8080'
```

This is a global definition. For selected databases (local web-based services, for example) you can turn off the proxy inside the database definition with:

```
DB [ ...
  proxy: ":"
]
```

HTTP access by default used HTTP protocol version 1.0. EMBOSS can also support version 1.1, which provides chunked HTML results to improve improve network performance. The HTTP version is controlled by a variable EMBOSS_HTTPVERSION and by a DB attribute, for example:

```
set emboss_httpversion "1.1"
```

or

```
DB [ ...
  httpversion: '1.1'
]
```

3.2.3 Mixed access methods

For any given `method:` declaration, **EMBOSS** will use that method for those access modes supported by the method.

If you wish to specify which access mode (all, query or single) should be handled by which database retrieval method then the `methodsingle:`, `methodquery:` and `methodall:` declarations should be used instead of `method:`

```
DB mydb [
methodsingle: app
format: fasta
app: "customapp myproteindb"
methodall: direct
dir: \$emboss_db_dir/myproteindb
file: myproteindb.dat
type: P
comment: "single and all access for myproteindb"
]
```

You can mix these, for example, to use a script to query a file, and direct acces to read all entries,

```
  methodall: 'direct'
  methodquery: 'external'
```

3.2.4 Indexing and configuring flatfile databases

Flatfile databases are plain text files in a defined format such as those released by EMBL, Swissprot and so on. The **EMBOSS** program DBIFLAT is used to generate EMBLCD indices that can be used for all types of database access. DBIFLAT can process databases in EMBL, SWISSPROT and GENBANK format. Pseudo EMBL format databases which do not have unique ID and AC entries may cause DBIFLAT to do mysterious things and should be avoided.

DBIFLAT (and the EMBLCD access method) requires the databases to be uncompressed. The examples given here will not probe the deeper secrets of DBIFLAT (for which the reader is referred to the documentation, or failing that the source code) but will show a typical installation for a common database.

We assume that **EMBOSS** has been installed and works. This can be tested with the command `wossname -auto` which should list all the programs available.

In this example we will index and configure the EMBL database for use with **EMBOSS**.

First download and unpack the EMBL database. This will require a considerable amount of disk space. If you do not have sufficient space available then just download a subset of the database.

Use `cd` to move the directory in which you have unpacked EMBL. This should look something like this when you run `ls`:

```
% ls
est_fun.dat
est_hum1.dat
est_hum10.dat
.
Output truncated
.
syn.dat
unc.dat
vrl.dat
vrt.dat
```

Run DBIFLAT to create the EMBLCD indices.

```
% dbiflat

Index a flat file database
      EMBL : EMBL
     SWISS : Swiss-Prot, SpTrEMBL, TrEMBLnew
        GB : Genbank, DDBJ
```

```
Entry format [SWISS]: EMBL
Database name: embl
Database directory [.]:
Wildcard database filename [*.dat]:
Release number [0.0]: 63.0
Index date [00/00/00]: 31/07/00
```

DBIFLAT should happily chug away for some considerable time (up to a few hours depending on the speed of your machine) and will generate (eventually) the following index files:

```
% ls
acnum.hit
acnum.trg
division.lkp
entrynam.idx
```

Now we create an entry in the **EMBOSS** configuration files to acces sthe database. It is probably a good idea to try new database definitions in your local configuration file first.

Put the following entry in your *.embossrc*

```
DB embl [
   type: N
   method: emblcd
   format: embl
   dir: \$emboss_db_dir/embl
   file: "*.dat"
   release: "63.0"
   comment: "EMBL release 63.0"
]
```

you will have needed to predefine `$emboss_db_dir` using a directive such as

```
set emboss_db_dir /path_to_databases
```

somewhere in your *emboss.default* or *.embossrc*.

Save *.embossrc* and try SHOWDB. You should see a line that looks like:

```
% showdb
.. output deleted
embl          N    OK  OK  OK  EMBL release 63.0
.. output deleted
```

3.2.5 Fine tuning the installation:

It is probably a good idea to set up subsections of the database so that end users can search just the regions they wish to search. This section applies to all access methods that use EMBLCD style indexes and probably to others as well.

Files can be included with the declaration `file:` or excluded with the declaration `exclude:`. It is a good idea to put the wild card directory specifier (*/)in front of the filename to ensure that any path that may be included in *division.lkp* will be matched. Please note especially the notes for GCG formatted databases indexed with DBIGCG.

In order to just take the EST files in our EMBL database try the following:

```
DB emblest [
   type: N
   method: emblcd
```

```
   format: embl
   dir: \$emboss_db_dir/embl
   file: "est*.dat"
   release: "63.0"
   comment: "EMBL release 63.0"
]
```

Files can also be given as a space separated list enclosed in quotes. For example to set up a database of all mamallian sequences (except genomes) try the following:

```
DB emblallmam [
   type: N
   method: emblcd
   format: embl
   dir: \$emboss_db_dir/embl
   file: "rod*.dat hum*.dat mam*.dat"
   release: "63.0"
   comment: "EMBL release 63.0"
]
```

As you can see from these two examples, the `file:` tag takes a space delimited list of filenames enclosed in quotes that can contain normal wildcard (`?*`) characters.

It can be quite tedious to set up a long list of sequences to search. In many cases you can use the `exclude:` tag to make things easier.

```
DB emblnoest [
   type: N
   method: emblcd
   format: embl
   dir: \$emboss_db_dir/embl
   file: "*.dat"
   exclude: "est*.dat"
   release: "63.0"
   comment: "EMBL release 63.0"
]
```

This configures the *emblnoest* database to contain all of EMBL except the EST's.

3.2.6 Indexing and configuring GCG format databases

EMBOSS can access GCG formatted databases, thus avoiding having multiple copies of the same databases in different formats for those who still use GCG alongside the flatfiles. **EMBOSS** creates EMBLCD like indices for the GCG format databases using the program DBIGCG. This runs in much the same way as DBIFLAT. You will need the GCG format *.seq* and *.header* files in order to create an EMBLCD indexed database.

Move to the GCG database directory containing your data and run DBIGCG

```
Index a GCG formatted database
      EMBL : EMBL
     SWISS : Swiss-Prot, SpTrEMBL, TrEMBLnew
        GB : Genbank, DDBJ
       PIR : NBRF
Entry format [EMBL]:
Database name: embl
Database directory [.]:
```

```
Wildcard database filename [*.seq]:
Release number [0.0]: 63.0
Index date [00/00/00]: 31/07/00
```

The program will chug along for a while and will then generate the EMBLCD index files for the GCG format database.

When DBIGCG prompts for the entry format (`Entry format [EMBL]:`) you should enter the original database format before you ran EMBLTOGCG or similar to generate the GCG databases.

The following entry should be put in your *.embossrc*

```
DB gcgembl [
   type: N
   method: gcg
   format: embl
   dir: \$emboss_db_dir/embl
   file: "*.dat"
   release: "63.0"
   comment: "EMBL release 63.0"
]
```

SHOWDB should show your newly configured database.

You can configure subsets of the databases in the same way as for the original format databases, described in section 3.2.4 above. One difference to DBIFLAT indexing is that both the *.seq* and *.header* files are listed in the *division.lkp* file. `file:` and `exclude:` directives should therefore be of the form `exclude: */em_est*` instead of just `*/em_est*.seq`.

3.2.7 Indexing and configuring BLAST databases

BLAST format databases are generated for efficient homology searching using the BLAST programs. It can be convenient to avoid redundant copies of databases so **EMBOSS** provides a mechanism for accessing these databases.

BLAST format databases are those generated using the tools distributed with NCBI-BLAST or with WU-BLAST.

For indexing of one BLAST database, move to the directory containing your BLAST format databases and run DBIBLAST

```
Index a BLAST database
Database name: blastsw
Database directory [.]:
database base filename [blastsw]:
Release number [0.0]:
Index date [00/00/00]:
         N : nucleic
         P : protein
         ? : unknown
Sequence type [unknown]: p
         1 : wublast and setdb/pressdb
         2 : formatdb
         0 : unknown
Blast index version [unknown]: 2
```

The program will chug along for a while and will then generate the EMBLCD index files for the BLAST format database.

The following entry (or one like it that is more appropriate to your particular installation) should be put in your *.embossrc*

```
DB blastsw [
   type: P
   method: blast
   format: ncbi
   dir: \$emboss_db_dir/blastsw
   file: "blastsw"
   release: "38.9"
   comment: "BLAST format Swissprot"
]
```

SHOWDB should show your newly configured database.

Because of the way BLAST works, many sites may group their BLAST databases in the same directory. You can index these *in situ* with DBIBLAST but this may require some extra steps if your databases are not of the same type as generation of subsequent index files will overwrite those that already exist. To avoid overwriting of index files you can index many databases with one set of index files, or you can use the `indexdir` options to place the indices in a different directory.

There are two requirements for indexing several databases together in one index. The first is that the databases are the same type (protein/nucleic acid) and generated with the same tool (pressdb or formatdb); the second is that all the ID and accession numbers in the combined databases are unique.

Run DBIBLAST as before but specify all the databases you wish to be included when prompted for the database filename.

```
Index a BLAST database
Database name: alldbs
Database directory [.]:
database base filename [alldbs]: dbone dbtwo dbthree dbfour
Release number [0.0]:
Index date [00/00/00]:
         N : nucleic
         P : protein
         ? : unknown
Sequence type [unknown]: p
         1 : wublast and setdb/pressdb
         2 : formatdb
         0 : unknown
Blast index version [unknown]: 2
```

These can then be configured as described in section 3.2.4 above by using the '`file:`' and '`exclude:`' tags as appropriate.[6]

When you have databases of different types, generated with different programs or where the ID/accession numbers are duplicated between databases the preferred strategy is probably to keep the source data for the individual databases in separate directories and index them there.[7]

Alternatively you can place the index files in a separate directory. This requires that you run DBIBLAST with the `-indexdirectory` option and set the `indexdir:` tag in the database configuration to point to the correct database. The example below illustrates database configuration using the `indexdir` options.

[6]There is one difference to the standard EMBLCD access method in that the database indexes will not allow the generation of exclusive subsections of the combined database. If an ID or accession number is specified that is present in the index then the sequence will be returned irrespective of which database it is in.

[7]Keeping one directory with symbolic links for your BLAST installation will ensure that BLAST continues to function correctly if you set BLASTDB to point to the directory containing the symbolic links. The EMBOSS indices can be placed wherever you wish as long as you remember to run DBIBLAST with the appropriate options and put an appropriate `indexdir` tag in the DB configuration in your /.embossrc

```
% dbiblast -indexdir=/databases/indices/mydb
Index a BLAST database
Database name: mydb
Database directory [.]:
database base filename [mydb]:
Release number [0.0]:
Index date [00/00/00]:
         N : nucleic
         P : protein
         ? : unknown
Sequence type [unknown]: p
         1 : wublast and setdb/pressdb
         2 : formatdb
         0 : unknown
Blast index version [unknown]: 2
```

The corresponding entry in *~/.embossrc* (or *emboss.default*) would look like:

```
DB mydb [
   type: P
   method: blast
   format: ncbi
   dir: \$emboss_db_dir/blastsw
   indexdir: /databases/indices/mydb
   file: mydb
   release: "1.0"
   comment: "My BLAST DB with an index in a different directory"
]
```

Again, multiple indices cannot coexist in the same directory so care should be taken when using the `indexdir` options that an existing database index is not overwritten.

3.2.8 Indexing and configuring FASTA databases

The FASTA specifications just define the sequence file as a header line that begins with `>` and subsequent lines containing the sequence. The header line can be present in an almost infinite number of formats, several of which can be processed by **EMBOSS**. **EMBOSS** attempts to determine the accession number and/or ID for each sequence. For indexing purposes there is no semantic difference between an accession number and an ID. In the real world, acession numbers are immutable, ie. they do not change with subsequent releases of the dataabse, but ID's may change. In any case IDs and accession numbers are unique, and that is all that matters for database indexing **EMBOSS**.

The program used to process FASTA format databases is DBIFASTA. It can recognise the following header line formats, specified on the command line:

simple	`>id ...`
idacc	`>id accno ...`
gcgid	`>db:id ...`[7]
gcgidacc	`>db:id acc ...`[7]
dbid	`>db id ...`[8]
ncbi	`>...[|accno]|id ...`[9]

[8] *db* is one word

[9] The ID is always taken to be the characters after the last bar (|). The previous field is also indexed but ONLY if it looks like an accession number (e.g. AC00001).

Other header formats will not be recognised by DBIFASTA and will cause indexing and/or database lookup to fail. If you have a different header format that DBIFASTA cannot yet handle you have two options:

1. (The preferred option) Get a C programmer to modify the source code for DBIFASTA and recompile. If you are a community spirited person you will also contribute these changes to the main **EMBOSS** source tree. (email emboss-dev@embnet.org for more information on contributing changes to the **EMBOSS** source code and/or read the **EMBOSS** developers documentation)

2. (The quick hack) Write a custom script (using e.g. BioPerl[10]) to access your database and use `method: external` to configure it. This is less desirable as you may be limited in the access modes you can use.

To index a FASTA format database, run DBIFASTA.

```
% dbifasta
Index a fasta database
    simple : >ID
     idacc : >ID ACC
     gcgid : >db:ID
  gcgidacc : >db:ID ACC
      ncbi : >blah|...[|ACC]|ID
ID line format [idacc]:
Database name: mydb
Database directory [.]:
Wildcard database filename [*.dat]: mydb.fasta
Release number [0.0]:
Index date [00/00/00]:
```

DBIFASTA will chug along for a little while and will produce the index files. You can use the same `indexdir` options as for DBIFLAT,DBIGCG and DBIBLAST to place the indices in a different directory.

Place the following entry in your *.embossrc*

```
DB mydb [
        type: P
        method: emblcd
        format: fasta
        dir: \$emboss_db_dir/mydb
file: mydb.fasta
        comment: "My database"
]
```

`format:` should be `dbid`, `ncbi` or `fasta` (for every format except `dbid` or `ncbi`. The same `file:` and `include:` tags can be used as for the other database indexing programs.

3.2.9 Configuring EMBOSS to use SRS for database lookup.

`method: srs` is really a special case of `method: external` with some additional features.

SRS is a powerful database querying system that can cross reference between different databases, launch applications and so on. SRS can be run either through a web interface (see the description of the URL method above for an example) or via the command line program GETZ. Indexing and configuring databases for SRS is outside the scope of this document which will describe how to connect to preconfigured and indexed SRS databases.[11] If GETZ is already in your `PATH` environment variable then insert the following (or similar) in your *.embossrc*:

[10] `http://www.bioperl.org`

[11] For information on configuring and indexing SRS databases please look at the SRS administrators guide *www/doc/srsadmin.pdf* in your SRS 6 installation

```
 DB emblgetz [
    type: N
    method: srs
    release: "63"
    format: embl
    comment: 'EMBL using getz'
    dbalias: embl
    app: getz
]
```

This will provide access to the SRS database 'embl' as `emblgetz:acc`. If the SRS database has a different name to the **EMBOSS** database (as is the case here) then the `dbalias:` tag should be used to access the correct SRS database.

This configuration can be extremely slow for the all access mode. It is probably a better idea to set up the database as follows:

```
 DB emblgetz [
    type: N
    methodquery: srs
    release: "63"
    format: embl
    comment: 'EMBL using getz'
    dbalias: embl
    app: getz
    methodall: direct
    file: "*.dat"
    dir: \$emboss_db_dir/embl
]
```

which will use `method: srs` for the `query` access mode but will use `method: direct` for the `all` access mode, thus speeding up reading of the whole database.

The SRSFASTA access method is identical to the normal SRS method except that it returns the sequence in FASTA format and so does not need a `format:` tag.

3.2.10 Indexing and configuring other databases

Many institutions may have local databases set up in their own Laboratory Information Management System. **EMBOSS** provides a simple mechanism for interfacing with such systems.

As long as a program is available that can be called noninteractively and returns the specified sequence on standard output, **EMBOSS** can interface with it. Use method: app or external (the two are equivalent) and app: "program command". The ID given in the USA will be appended to the command used to run the program. It is probably best to specify the methods available using the method subsets, methodall:, methodquery: and methodsingle: rather than using the generic method: tag.

3.3 Other data

EMBOSS can be integrated with some common biological databases. These are described in this section.

3.3.1 REBASE

Rebase is the restriction enzyme database maintained by New England Biolabs. It is needed for programs such as remap and restrict.

The latest version of Rebase can be obtained by anonymous FTP.[12] **EMBOSS** needs the *withrefm* file. The data is extracted for **EMBOSS** with the program REBASEEXTRACT.

```
% mkdir /site/prog/emboss/data/REBASE
% rebaseextract
Extract data from REBASE
Full pathname of WITHREFM: /data/rebase/withrefm.208
```

Rebase is now installed and ready to use.

3.3.2 TRANSFAC

Transfac is the transcription factor binding site database. It is available by anonymous FTP.[13] Unpacking the distribution reveals a file called site.dat. This is the one **EMBOSS** needs.

Run TFEXTRACT to extract the data from TRANSFAC.

```
% tfextract
Extract data from TRANSFAC
Full pathname of transfac SITE.DAT: /databases/transfac/site.dat
```

TFSCAN can now access the TRANSFAC database.

3.3.3 PROSITE

Prosite is a database of regular expressions that match potentially diagnostic regions for structural/functional classification of proteins. **EMBOSS** needs this database for the patmatmotifs program.

PROSITE can be obtained via anonymous FTP.[14]

You may need to create a PROSITE subdirectory under data in the **EMBOSS** installation directory.

Then run PROSEXTRACT to build the **EMBOSS** Prosite database.

```
% prosextract
Builds the PROSITE motif database for patmatmotifs to search
Enter name of prosite directory: /data/prosite
```

PROSITE is now integrated into your EMBOSS installation.

3.3.4 PRINTS

Prints is a database of diagnostic patterns of blocks of sequence homology in protein families. The PRINTS database can be searched using the **EMBOSS** program PSCAN.

PRINTS can be obtained via anonymous FTP.[15] The database is made available as compressed files which should be uncompressed using GZIP before integrating them into **EMBOSS**

PRINTS is integrated with **EMBOSS** using the program PRINTSEXTRACT

```
% printsextract
Extract data from PRINTS
Input file: /data/prints/prints27_0.dat
```

The PRINTS database is now integrated with **EMBOSS**.

[12] ftp://ftp.ebi.ac.uk/pub/databases/rebase
[13] ftp://transfac.gbf.de/pub/transfac/ascii/
[14] ftp://ftp.ebi.ac.uk/pub/databases/prosite
[15] ftp://ftp.ebi.ac.uk/pub/databases/prints

3.3.5 AAINDEX

An amino acid index is a set of 20 numerical values representing any of the different physicochemical and biological properties of amino acids. The AAindex1 section of the Amino Acid Index Database is a collection of published indices together with the result of cluster analysis using the correlation coefficient as the distance between two indices. This section currently contains 437 indices in release *4.0* of the database.

The **EMBOSS** programs PEPWINDOW and pepwindowall plot hydrophobicity using the data from an Aaindex entry. If Aaindex is installed these programs can plot the other amino acid properties.

Aaindex can be obtained via anonymous FTP.[16]

Aaindex is integrated with **EMBOSS** using the program AAINDEXEXTRACT

```
% aaindexextract
Extract data from AAINDEX
Full pathname of file aaindex1: /data/aaindex/aaindex1
```

The AAINDEX database is now integrated with **EMBOSS**.

3.3.6 CUTG

The CUTG database contains a series of codon usage tables calculated from GenBank.

CUTG can be obtained via anonymous FTP.[17]

CUTG is integrated with **EMBOSS** using the program CUTGEXTRACT which writes files to the CODONS data directory.

```
% cutgextract
Extract data from CUTG
CUTG directory [.]: /data/cutg/
```

The CUTG database is now integrated with **EMBOSS**.

3.3.7 Miscellaneous data files

Other data files should be kept in the data directory under the main **EMBOSS** installation. Individual users personal data files can be kept in the current working directory, a subdirectory *.embossdata* of the current directory, their home directory or a subdirectory *.embossdata* of their home directory. **EMBOSS** will search these locations in this order and will stop as soon as it finds a matching file. If the personal directories do not contain the desired file, **EMBOSS** will search the system wide data directory, */site/prog/emboss/data* in this example.

Apparently inexplicable errors when running **EMBOSS** programs may be caused by the system not using the data files one expects. The search path can be displayed in search order using the command EMBOSSDATA.

3.4 Default program settings

As with many other areas, the default behaviour of programs can be controlled by setting appropriate values in *.embossrc*.

All general qualifiers[18] can be specified as

```
set emboss_QUALIFIER 1
```

[16] ftp://ftp.genome.ad.jp/pub/db/genomenet/aaindex/aaindex1

[17] ftp://ftp.ebi.ac.uk/pub/databases/cutg/ or ftp://ftp.kazusa.or.jp/pub/codon/current/

[18] See the **EMBOSS** Quick Guide or the web documentation (or use `wossname -help -verbose`) for an overview of general qualifiers.

where QUALIFIER is one of the general qualifiers and the value can be 1 or 1 for true, or 0 or N for false.

Setting the qualifier value to true has the effect of running every program with that qualifier set.[19] Qualifiers can be set and will work in the same way as if you set them when running the program. For example you can set emboss_verbose Y and the program will run normally, but when the program is run with the -help qualifier, the output will be in verbose form.

There is no point in globally setting options that are there for producing help output.

Qualifiers that can be set:

VERBOSE Causes -help to print verbose text.

STDOUT Causes all output to go to *STDOUT* as default. Programs will usually build a default output file name form the input sequence and the program name.

DEBUG Writes debugging output to a file. Useful for finding bugs as a command line option.

OPTIONS Enable prompting for optional parameters.

FILTER Take input from *STDIN* and send it to *STDOUT*, and turn on -auto

AUTO Do not prompt for any options but accept the defaults if no values are given.

WARNING Print warning messages to *STDERR* (default is true)

ERROR Print error messages to *STDERR* (default is true)

FATAL Print fatal messages to *STDERR* (default is true)

DIE Print crash messages to *STDERR*

These general qualifiers are typically used by advanced users (-options, -verbose) or by developers (-debug -acdlog).

Other program options that can be set are emboss_format, emboss_acdroot, and emboss_data. The value of emboss_format determines which default sequence format to use for output. for example, if you are running **EMBOSS** alongside GCG you may wish to have the following entry in your .EMBOSSRC

```
set emboss_FORMAT gcg
set emboss_OUTFORMAT gcg
```

which has the effect of using GCG format by default.[20]

emboss_acdroot */path/to/acd* can be set if you wish to use a different directory for the ACD files, and emboss_data */path/to/data* if you wish to use a separate data directory.

3.5 Logging

Many system administrators may wish to make use of the logging facilities of **EMBOSS**Ṡetting the variable emboss_logfile in *emboss.default* or *.embossrc* allows the system to keep a log of which programs are used when and by whom.

```
set emboss_logfile /site/log/emboss.log
```

The log file structure is very simple. Three tab separated fields are stored, program name, user name, and the date and time.

[19] You can specifically unset it by using the -noQUALIFIER command line option

[20] This can of course be overridden using the -sformat and -osformat associated qualifiers. See the **EMBOSS** ACD Syntax documentation or the **EMBOSS** Quick Guide for more information.

```
prettyplot      joeuser        Wed Aug 02 14:29:13 2000
```

The file defined in emboss_logfile should be world writable. The following command ensures logging can occur.

```
chmod +w /site/log/emboss.log
```

All settings can be overridden in a users *.embossrc* files by redefining the relevant variables. So to prevent our system usage being logged we can redefine emboss_logfile by putting the following entry in our *.embossrc* file.

```
set emboss_logfile /dev/null
```

This behaviour may change in the future to prevent users redefining some system settings.

4 Graphical interfaces to EMBOSS

This chapter needs to be written. It will be written when the available GUIs are stable enough to document.

5 Resources

5.1 Web sites

5.1.1 Programs

EMBOSS source code ftp://ftp.uk.embnet.org/pub/EMBOSS

EMBOSS Documentation http://www.uk.embnet.org/Software/EMBOSS

BLAST tools Tools for generating BLAST format databases are contained in the NCBI toolkit which can be obtained from NCBI at:

http://www.ncbi.nlm.nih.gov/

SRS software The SRS software can be obtained from Lion Bioscience.[1] This is a commercial package but at the time of writing is available free of charge to academic institutions.

WGET Various useful utilities including the WGET program are available from the Free Software Foundation.[2]

5.1.2 Databases

Most of the databases mentioned in the text along with many others can be obtained via anonymous ftp from the European Bioinformatics Institute (EBI) at:

ftp://ftp.ebi.ac.uk/pub/databases

Please use a mirror site where possible to avoid overloading of the EBI's resources.

Other databases can be obtained from NCBI (Genbank,UniGene etc.)

5.1.3 Other Documentation

Please review the **EMBOSS** documentation available on the WWW at the URL above.

The EMBOSS Quick guide A pocket reference guide to using **EMBOSS**[3].

The EMBOSS Tutorial A tutorial to give an introduction to using **EMBOSS** for bioinformatics users.[4]

The updated ABC guide This is a series of bioinformatics practicals based predominantly on **EMBOSS**.[5]

EMBOSS-FreeBSD-HOWTO Detailed documentation on installation of **EMBOSS** on FreeBSD.[6]

[1] http://www.lionbioscience.com/solutions/srs
[2] http://www.gnu.org
[3] ftp://ftp.no.embnet.org/pub/EMBOSS-extra/emboss-qg.ps
[4] http://www.hgmp.mrc.ac.uk/Registered/Option/emboss.html
[5] ftp://ftp.no.embnet.org/pub/ABC
[6] ftp://ftp.no.embnet.org/pub/EMBOSS-extra/EMBOSS-FreeBSD-HOWTO

5.2 Maintainance of your EMBOSS installation

EMBOSS is a rapidly evolving software packages. It is constantly being improved, new features added and 'issues' resolved. In addition there are new applications added and you probably want to make use of these.

5.2.1 Automated installation of EMBOSS and EMBASSY

Once you have installed **EMBOSS** and got it to work you have solved the hardest part of the struggle. Updating **EMBOSS** as new releases appear[7] can be quite tedious. UNIX is designed for the lazy, so here is our lazy man's guide to always having an up to the minute **EMBOSS** installation.

The following script can be run manually (it should probably be '`source`d' rather than executed directly) or can be fired off with cron (in the early hours of the morning is a good time). It assumes you are installing **EMBOSS** outside the source directory and have write permissions to do so.

EMBOSS will update **EMBOSS** distributed files but will not alter or overwrite your own datafiles[8] or your *emboss.default*.

```
# This script should be sourced, not run.
# EMBOSS UPDATE.
# it assumes \$packages_dir/EMBOSS is a symbolic link to
# \$mirror_dir/ftp.uk.embnet.org/pub/EMBOSS
#

#site specific variables: season according to taste..

set mirror_dir=('/ftp/mirrors')
set packages_dir=('/site/newprog')
set emboss_config_options=\
('--prefix=/site/prog/emboss --with-pngdriver=/site/lib')

# Now the script proper

set oldpwd=`pwd`

cd \$mirror_dir
echo 'updating EMBOSS'
if ( `wget -m 'ftp://ftp.uk.embnet.org/pub/EMBOSS' |& \
  tail -1 | awk '/^Downloaded:/{print \$5}'` != "0" ) then

    cd \${packages_dir}/EMBOSS
    echo 'new EMBOSS programs found .. installing'
    set latest_emboss=`ls -t EMBOSS*|head -1`

    cd \$packages_dir
    rm -Rf EMBOSS-*
    tar zxf EMBOSS/\$latest_emboss
    set emboss_dir=`ls -dt EMBOSS-*[^z]|head -1`

#the next line is necessary on our system but may not be for yours.
    setenv LD_LIBRARYN32_PATH /site/lib
```

[7] **EMBOSS** is rebuilt nightly from CVS, tested, and, assuming it passes the compilation tests, the latest version is posted to the **EMBOSS** FTP server.

[8] Assuming of course that you haven't overwritten **EMBOSS** datafiles with your own to begin with.

```
    cd \$emboss_dir

# If you have any site specific changes to the source code
# that you want to include, copy them in here

    ./configure \$emboss_config_options &&\
    make && \
    make install

#Now unpack and build EMBASSY

    mkdir embassy
    cd embassy

#Unpack and build each package one at a time

    foreach embassadir ( `ls ../../EMBOSS/*gz |grep -v E
MBOSS-` )

tar zxf \$embassadir
set embassadir_arch=\$embassadir:t
set embassadir_root=\$embassadir_arch:r

cd \$embassadir_root:r
./configure  \$emboss_config_options &&\
make && \
make install

cd ..
    end
else
    echo 'No new version of EMBOSS available'
endif

cd \$oldpwd
```

5.2.2 Automated database updating

In the same way, scripts can be written to automatically update the biological databases. An example is given here for REBASE. As all the parameters for **EMBOSS** programs can be specified on the command line it is a trivial matter to include index generation in your nightly update scripts. The management of a bioinformatic resource is beyond the scope of this document, though **EMBOSS** goes a long way towards easing the burden of management.

Automated update of REBASE

This script will look for a new version of REBASE and install it in **EMBOSS** using REBASEEXTRACT.

```
# This script should be sourced, not run.
# REBASE UPDATE. Should be run just after the beginning of the month.
set mirrors_dir=('/ftp/mirrors')
set oldpwd=`pwd`

cd \$mirrors_dir
```

```
if ( ` wget -m 'ftp://ftp.ebi.ac.uk/pub/databases/rebase/*' |& \
  tail -1 | awk '/^Downloaded:/{print \$5}'` != "0" ) then
cd ftp.ebi.ac.uk/pub/databases/rebase
cp `ls -t withrefm.*.Z|head -1` withrefm.Z
uncompress withrefm.Z
rebaseextract \
  \${mirrors_dir}/ftp.ebi.ac.uk/pub/databases/rebase/withrefm
rm withrefm
endif

cd \$oldpwd
```

We make no guarantees that these scripts will work correctly on your system. If it deletes all your files, spams your associates, scratches your CD's and initiates a nuclear strike on a small unpopulated pacific island it is NOT OUR FAULT. It just happens to work for us.

6 GNU Free Documentation License

GNU Free Documentation License
Version 1.1, March 2000

0. 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. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work 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, LaTeX 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.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies 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.

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 less than five).
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", 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.
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. 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.
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 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.

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."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this

License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, 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.

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.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

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 1.1
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.

7 Acknowledgements

The acknowledgements and credits are found at the front of this guide because no one ever reads them if they are at the back.