Emacs Installation HOWTO

Following a brief discussion of the advantages of using the Emacs text editor, a process for obtaining and installing the software on RFPK workstations is described.

Table of Contents

Introduction: Why Emacs?

Unix depends heavily upon text files, both for data and for configuration. This contrasts sharply with Windows and various legacy mainframe operating systems, which depend on proprietary binary files, instead. Whereas text editing in a Windows environment is usually accomplished using the rudimentary Wordpad or Notepad programs, the Unix environment offers much more powerful tools. These tools make Unix a preferred environment for text work. (This also applies to Linux, the free Unix-compatible operating system, and to Cygwin, a Unix-compatible environment which runs on top of the Windows operating system.) It should also be pointed out that the growing importance of text-based markup languages, including SGML, XML and HTML, is elevating the importance of text files in all operating system environments, including Windows.

The principal text editors are Vi and Emacs. To underline the importance and the exceptional quality of these programs, it is worth noting that they were created by some of the worlds most illustrious programmers. Vi was the work of Bill Joy, the technical leader of Sun Microsystems, and one of its founders. Emacs is the work of Richard Stallman, the founder of GNU and the open source movement, along with many others, including Jim Gosling, the creator of Java.

How do you choose one or the other? This depends on the nature of the work that you do. The two editors fill somewhat different roles. For many years now, Vi has been the default editor for Unix and Unix-like systems. Whenever you log in to one of these systems, you can be nearly certain that Vi will be there. For this reason, at least some familiarity with Vi is a necessity for someone who is doing any system administration or application configuration work. Emacs, on the other hand, may be absent on a system that you need to use. Correcting that problem is the main topic of this tutorial.

Emacs came originally out of the MIT Artificial Intelligence Laboratory. Though the core of the program is written in C, for machine efficiency, it is extended in Lisp, for human efficiency. It is this extensibility that sets Emacs apart. Important extensions, known as modes, add features that understand the syntax of the particular file or files being edited. For example, there is a C++ mode that understands the syntax of the C++ programming language. There is an SGML/XML mode that understands not only the syntax of SGML or XML in general but, if the document starts with a reference to a Document Type Definition (DTD), it understands the syntax of that particular markup language. For example, the SGML/XML mode enables Emacs to understand the syntax of Docbook or the syntax of Xhtml. Because of the availability of many preprogrammed modes, Emacs should be preferred over Vi for editing documents which must adhere to complex syntax rules.

Which Emacs?

Going back to the 1970s, there have been a number of versions of Emacs. From about 1987 until the late 1990s, however, GNU Emacs was without a doubt the Emacs of choice. This editor was free, available on a wide variety of platforms, and was the product of Richard Stallman, the creator of the original version of Emacs.

Until the late 90s, the Emacs user interface was entirely driven by commands expressed as keystroke sequences starting with control codes. This design was very efficient for experience users, but could be difficult to learn and difficult to remember for occasional users. The GNU Emacs developers were on their way to creating a menu-driven, more user-friendly version of Emacs, but were proceeding very deliberately.

All too, deliberately, in the opinion of some. A new, independent open source project, called XEmacs was born with the goal of producing an Emacs with the long overdue user-friendly features. The XEmacs project put pressure on GNU Emacs, forcing it to catch up with the times.

Today we have a choice of two user-friendly versions of Emacs. Although we owe the XEmacs developers a debt of gratitude for lighting a fire under the feet of their GNU Emacs counterparts, this tutorial recommends the use of GNU Emacs for the reasons of guaranteed compatibility and respect for Richard Stallman and the GNU tradition. The rest of this document will concentrate on a process for obtaining and installing the latest version of GNU Emacs.

Installing Emacs in a Linux Environment

Emacs is one of the usual components in a Linux system. It is packaged with many optional modes already in place, including the PSGML mode and the C programming mode.

Install Emacs

If your Linux computer was configured for software development, you should already have GNU Emacs installed. If not, you can install the emacs packages from your Linux installation cdrom.

Install the .emacs File

When emacs starts, it reads a configuration file, if one is available. This file can provide a large amount of customization. For instance, it can provide a list of colors to be used for syntax coloring; it can specify a default DTD; it can specify many options. Because it can be very complex and usually contains lisp code, a detailed description is outside the scope of this document. Instead, the installation of a pre-made configuration file will be described.

1. Open up a shell window in the desktop.
2. Copy the file `/opt/download/dot.emacs` from whitechuck , then concatenate it to the `.emacs` file in your home directory. If there was no `.emacs` to start with, this will create one.

```
scp whitechuck:/opt/dot.emacs .
cat dot.emacs >> .emacs
```

Installing Emacs in a Cygwin Environment

The process that follows assumes that the user had already installed Cygwin. If that is not the case, the Cygwin Installation[1] HOWTO will help you to accomplish this.

Install Emacs

In the following procedure, it is assumed, for sake of this example, that the latest version of GNU Emacs for Windows is version emacs-21.2.

1. Decide in which directory you want to install Emacs. This tutorial assumes that you will use the directory `/usr/local` within your Cygwin environment.
2. Go to the website http://ftp.gnu.org/gnu/windows/emacs/latest which contains links for downloading the binary distribution of the latest version of GNU Emacs for Windows.
3. While depressing the **Shift** key, left-click the link to `emacs-21.2-fullbin-i386.tar.gz`
4. A file window will appear. Select the directory into which you want the file to be downloaded, and click on the **Save** button.
5. After the download has completed, open a Cygwin shell window and use the **cd** command to go to the directory which contains the new file.
6. Decompress and expand the file by issuing the command

   ```
   tar xzf emacs-21.2-fullbin-i386.tar.gz
   ```

7. Run the Emacs installation program:

   ```
   emacs-21.2/bin/runemacs
   ```

8. Add Emacs to your execution path. In this example we assume that you installed Emacs in the directory `/usr/local` within your Cygwin environment.

   Add the following lines to the `.bash_profile` configuration file in your home directory

   ```
   EMACSBIN=/usr/local/emacs-21.2/bin
   PATH=$EMACSBIN:$PATH
   ```

   Test this by first closing all Cygwin shell windows, then opening a shell window and typing

   ```
   emacs &
   ```

   An emacs window should open up, and run independently of the Cygwin window.

9. When you are convinced that everything is working, you can clean up by removing the download file, `emacs-21.2-fullbin-i386.tar.gz`

Install the PSGML Enhancement

Psgml provides the SGML/XML major mode. Once installed, whenever you open a file which has the `.xml` filename suffix, Emacs will be transformed into a specialized XML editor.

Emacs will understand the syntax of a *well-formed* XML document. Essentially this means that elements are delimited by a start-tag and an end-tag; that elements cannot overlap; and that there is a single top-level element, which is the document itself. Emacs will color tags and provide automatic indentation in order to emphasize this structure and to make it obvious if structure rules are violated.

If in addition to the document being well-formed, the preamble of the document contains a reference to a file called a *Document Type Definition* (DTD), Emacs will read that file, whether it resides on the local machine or somewhere on the Internet, and use the information to provide additional syntax enforcement as well as labor-saving features.

One of the things that a DTD (such as the Docbook DTD) does is specify all of the elements that are allowed, what tag pairs delimit each element, and which elements can be contained in which other elements. Based on the position of the cursor within the document, the **Markup** menu provides a list of elements allowed at that point. When you click on one of the elements named in the pull-down list, a pair of tags is generated automatically. For speed, you can also get tag generation via auto-completion, after typing a key-stroke combination and the first few letters of an element name.

To obtain and install Psgml:

1. Go to the PSGML Project[3] page at SourceForge, and follow the link to the download page for the latest release. There you can download the source file for the latest version, in compressed tar (`.tar.gz`) format into a temporary directory.
2. In a Cygwin shell window, use **cd** to go to the temporary directory. Use **tar** to expand the file. Suppose that the file is named `psgml-1.2.5.tar.gz` ; then the command would be

   ```
   tar xzf psgml-1.2.5.tar.gz
   ```

3. Use the **configure** shell script to create a tailored `Makefile`:

   ```
   cd psgml-1.2.5
   ./configure
   ```

4. Use **make** to have Emacs byte-compile the lisp programs:

   ```
   make
   ```

5. Use **make** to create the directory `/usr/local/share/emacs/site-lisp` if it does not already exist, and to copy necessary files to it:

   ```
   make install
   ```

Install the Cygwin Mount Enhancement

When used in a Cygwin environment, the Windows version of GNU Emacs needs a little help in properly handling Cygwin pathnames and logical links. The enhancement consists of lisp code which you can copy from the RFPK server called *muir*.

1. Using two **My Computer** windows, copy the file with Windows pathname

   ```
   \\muir\SoftwareEngineers\Download\cygwin-mount.el
   ```

   to the directory

```
\cygwin\usr\local\share\emacs\site-lisp
```

on the drive that contains your Cygwin environment.

2. Byte-compile `cygwin-mount.el` so that it will execute more efficiently.

   Open a Cygwin shell window and input the following:

   ```
   cd /usr/local/share/emacs/site-lisp
   emacs &
   ```

   This should start an Emacs window. Move screen focus to that window (normally accomplished by moving the mouse pointer to it) and then perform the following:

   a. Type ALT-x (hold down the ALT key while pressing the x key). The cursor should jump to the mini-buffer at the bottom of the Emacs window.

   b. Type

      ```
      byte-compile-file
      ```

      In the minibuffer, Emacs will show the pathname of the current directory, so that you can append to it the name of the file you want compiled.

   c. Type

      ```
      cygwin-mount.el
      ```

      followed by the Enter keystroke and Emacs should compile the file, informing you of success or failure.

Install the .emacs File

When emacs starts, it reads a configuration file, if one is available. This file can provide a large amount of customization. For instance, it can provide a list of colors to be used for syntax coloring; it can specify a default DTD; it can specify many options. Because it can be very complex and usually contains lisp code, a detailed description is outside the scope of this document. Instead, the installation of a pre-made configuration file will be described.

By default, the name of the Emacs configuration file is `.emacs` and it is located in your home directory. Note that the file is a *hidden file* under Unix, Linux, or Cygwin, because its name starts with the dot character. To see it in the output of the **ls** command, use the `-a` option.

How does Emacs know where to find your home directory? If you start it from a Cygwin shell window, Emacs will inherit the `HOME` environment variable from the shell. If you start Emacs from a desktop icon, `HOME` may be undefined.

If you ever intend to run Emacs outside of Cygwin, it would be a good idea to define `HOME` as a Windows environment variable. You do this in the **Environment Variables** screen, reached by the following series of menus: **Start**, **Settings**, **Control Panel**, **System**, **Advanced**, **Environment Variables**. In the *User Variables* section of the screen, define the variable `HOME`, and specify the full path of the directory in which your `.emacs` file resides (normally the full Windows path of your Cygwin home directory) as its value.

To use Emacs for XML, you do not have to create or even understand your `.emacs` file. Instead, you can make a copy of a file that someone else has developed for this purpose. A suitable file resides on the RFPK server named *muir*. The windows pathname of the file to copy is

```
\\muir\SoftwareEngineers\Download\dot_emacs
```

You can use the Windows **My Computer** explorer to copy this file to your home directory and then change the name to `.emacs` .

Configure Emacs for C or C++

When Emacs opens a file whose name ends in one of the usual C or C++ suffixes (.c, .h, .cpp, .cc, etc.), the CC mode is loaded. This mode can provide syntax coloring, indentation and parentheses matching, as well as interfacing with **make** and with **gdb**, the GNU debugger. The user does not have to do much to configure this behavior. There are some options, however, that can be specified in the **Options** menu.

If you change options, the **Options** menu will allow you to save them. The new values are automatically appended to your `.emacs` file so that they will be applied the next time you start Emacs.

Learning Emacs

Emacs is not too difficult to learn, especially the menu-driven version. There are a number of freely available documents that can help you.

- The built-in tutorial is a good place to begin. Simply start emacs and choose **Emacs Tutorial** from the **Help** menu.
- The GNU Emacs Reference Card. This is available in `.pdf` format on the RFPK server, *muir*, at pathname

  ```
  \\muir\SoftwareEngineers\Download\EmacsRefcard.pdf
  ```

  Using the Adobe Acrobat Reader, you can print this double-sided, on a single piece of paper, so that it can be folded into a reference card of six panels.

  In the Print window, select **HP LaserJet 5000** for the **Name** field, and **Language Level 2** for the **Print Method** field.

  In the **Properties** menu, select the **Layout** page.

  In the **Orientation** menu select the **Landscape** radio button; and in the **Print on Both Sides (Duplex)** menu select the **Flip on Short Edge** radio button.

  Finally press the **OK** button to print your reference card.

- The PSGML manual. The Emacs mode for editing XML and SGML is called PSGML. The principal developer for this module is Lennart Staflin and he has written a manual, called *Editing SGML with Emacs and PSGML*, which covers the special features that the module has added to Emacs. You can get a copy of this manual in postscript from the RFPK server, muir, at the pathname

```
\\muir\SoftwareEngineers\Download\psgml.ps
```

- The official GNU Emacs manual is available online [4].

Conclusion

Emacs is the most powerful and most configurable text editor ever created. We have only skimmed the surface when discussing a few of the features of this tool. The time and effort needed to acquire, install, and learn to use this unique resource will be an excellent investment for anyone with a need to edit structured text.

Notes

1. http://muir.rfpk.washington.edu:800/HOWTO/CYGWIN-rfpk-HOWTO.html
2. http://ftp.gnu.org/gnu/windows/emacs/latest
3. http://sourceforge.net/projects/psgml/
4. http://www.gnu.org/manual/emacs/