Emacs ElementaryX: Elementary Emacs configuration coupled with Guix

Fork of Emacs Bedrock: stepping stones to a better Emacs experience

compose development team

July 24, 2025

Contents

1 Why you should *not* use `emacs elementaryx` 2
- 1.1 But use the original `bedrock` . . . 2
- 1.2 But use other starter kits . . . 3
- 1.3 Because you are not on `linux` or you do not want to use the `guix` package manager . . . 3

2 How to start `elementaryx` with `guix` (recommended)? 3
- 2.1 In a nutshell (recommended) . . . 3
- 2.2 Required dependency: `guix` . . . 5
  - 2.2.1 Pre-built binaries . . . 5
  - 2.2.2 Check up . . . 5
- 2.3 Preliminary step: folder and file creations . . . 6
- 2.4 Start `elementaryx` . . . 7
  - 2.4.1 Execution within an isolated container from both the home directory and the rest of the system . . . 7
  - 2.4.2 Execution in an isolated container from the rest of the system but writable in the home directory . . . 9
  - 2.4.3 Non isolated execution in a pure environment . . . 9
- 2.5 Starting `elementaryx` with non standard keybindings or on a remote machine . . . 10
  - 2.5.1 `vim` users . . . 10
  - 2.5.2 `windows`-like copy-paste users (`C-c C-v`) . . . 11
  - 2.5.3 `elementaryx` on a remote machine with mouse-tracking support . . . 11

3 How to start `elementaryx` with `docker` (alternative if you don't have `guix`)? 12
3.1 Retrieve or generate the `docker` image 12
3.1.1 Option 1: Retrieve it from `docker hub` (recommended) 12
3.1.2 Option 2: Generate it with `guix` (for information, and reproducibility) 12
3.2 Create, start and execute the container 13
3.3 Start `emacs elementaryx` 13
3.4 Support of `X11` and `tty` modes 13
3.4.1 `X11` mode 13
3.4.2 `tty` mode 13

4 First steps with `elementaryx` 13
4.1 Graphical user interface (GUI) 14

This page may be viewed online in html or pdf, while its sources are located on gitlab.inria.fr.

This project is a fork of the 1.2 version of the original emacs bedrock by ashton314. As mentioned in its synopsis, the original `emacs bedrock` was "intended to be copied once and then modified as the user grows in knowledge and power." We believe that the original version of `bedrock` is an excellent starting kit to learn about modern `emacs` (`emacs` 29.1 or later) and we strongly encourage `emacs` users to play with it.

1 Why you should *not* use `emacs elementaryx`

1.1 But use the original `bedrock`

`elementaryx` aims at achieving some specific workflows. In a way, it may be viewed as an extension of the original emacs bedrock by ashton314 to achieve them. On the other hand, it *breaks some of the very nice features* of the original `bedrock`. For instance, the original `bedrock` is delivered through a single `git` repository and depends only on one external package by default, which makes it very nice to play with. On the contrary, the present revisited version – `elementaryx` – is split into multiple repositories and may have many dependencies, which makes it hard to use. In addition, the focus on the considered workflows of the present version is certainly useless or suboptimum for many (most) `emacs` users.

1.2 But use other starter kits

While, according to its synopsis, the original `bedrock` is an "*extremely* minimal `emacs` starter kit", there exist much more fully-featured starter kits such as spacemacs or doomemacs. These starter kits provide an excellent user-experience, are developed and maintained by hundreds of contributors. The `elementaryx` suite does not – and will not – achieve 1% of their capacity, nor ergonomics.

Though, as shown by `asthon314` with the original `bedrock`, it is possible to design a pleasant set up of `emacs` with a modern `emacs` (version 29.1 or more), and we hope that this `revisited` version may also be pleasant to use for performing a few selected workflows.

1.3 Because you are not on `linux` or you do not want to use the `guix` package manager

As discussed in Section section 2, we rely on the guix package manager to use this `elementaryx`. It can be installed on any `linux` distribution (see below) but:

- you may not be on `linux`;
- or you may not want to install yet another package manager within your `linux` distribution.

In these cases, we do not recommend either to use `elementaryx`. Note that it is nevertheless possible to use it with `docker` as explained below.

2 How to start `elementaryx` with `guix` (recommended)?

2.1 In a nutshell (recommended)

In a nutshell, assuming we have `guix` available (see below in Section section 2.2), we can launch `elementaryx` in an isolated shell with guix time-machine using these guix-hpc fixed channels as follows:

```
mkdir -p "$HOME/org-roam"
mkdir -p "$HOME/references"
mkdir -p "/tmp/$(whoami)/elementaryx-test/.emacs.d"
wget https://gitlab.inria.fr/elementaryx/elementaryx.gitlabpag⌋
↪  es.inria.fr/-/raw/main/dotfiles/bash-elementaryx/.bashrc.e⌋
↪  lementaryx -O
↪  "/tmp/$(whoami)/elementaryx-test/.bashrc.elementaryx"
```

```
wget https://gitlab.inria.fr/elementaryx/elementaryx.gitlabpag⌋
↪  es.inria.fr/-/raw/main/channels-fixed.scm -O
↪  "/tmp/$(whoami)/elementaryx-test/channels-fixed.scm"
cd "/tmp/$(whoami)/elementaryx-test/"
SHELL="bash" guix time-machine --channels=/tmp/$(whoami)/eleme⌋
↪  ntaryx-test/channels-fixed.scm -- shell
↪  --share=/tmp/$(whoami)/elementaryx-test/
↪  --share="$HOME/org-roam" --share="$HOME/references"
↪  --expose=/tmp/$(whoami)/elementaryx-test/.bashrc.elementar⌋
↪  yx=$HOME/.bashrc --expose=$HOME/.config/guix/current=/guix
↪  --expose=/tmp/.X11-unix --expose=$XAUTHORITY
↪  --expose=/etc/ssl/certs
↪  --preserve="^SHELL|^DISPLAY$|^XAUTHORITY$|^TERM$" -CNW
↪  icecat emacs-elementaryx-as-default -- bash -c
↪  'GUIX_PROFILE="/guix" ; . "$GUIX_PROFILE/etc/profile" ;
↪  emacs'
```

The rest of the section explains these latter instructions and propose alternative execution modes. The above proposed execution mode corresponds to the concatenation of the preliminary setup from Section section 2.3 and the isolated execution from item 1. If you are happy with this execution mode you can skip the rest of the section and go to Section section 4 to play with `elementaryx`.

We just mention before that, that, although we recommend it, it is not necessary to create and share the `$HOME/org-roam` and `$HOME/references` as suggested above. In case you do not want to create and share them, we can create them in `"/tmp/$(whoami)/elementaryx-test/"` instead. In this case, proceed as follows:

```
mkdir -p "/tmp/$(whoami)/elementaryx-test"
mkdir -p "/tmp/$(whoami)/elementaryx-test/org-roam"
mkdir -p "/tmp/$(whoami)/elementaryx-test/references"
mkdir -p "/tmp/$(whoami)/elementaryx-test/.emacs.d"
wget https://gitlab.inria.fr/elementaryx/elementaryx.gitlabpag⌋
↪  es.inria.fr/-/raw/main/dotfiles/bash-elementaryx/.bashrc.e⌋
↪  lementaryx -O
↪  "/tmp/$(whoami)/elementaryx-test/.bashrc.elementaryx"
wget https://gitlab.inria.fr/elementaryx/elementaryx.gitlabpag⌋
↪  es.inria.fr/-/raw/main/channels-fixed.scm -O
↪  "/tmp/$(whoami)/elementaryx-test/channels-fixed.scm"
```

```
cd "/tmp/$(whoami)/elementaryx-test/"
SHELL="bash" guix time-machine --channels=/tmp/$(whoami)/eleme
↪  ntaryx-test/channels-fixed.scm -- shell
↪  --share=/tmp/$(whoami)/elementaryx-test/ --share="/tmp/$(w
↪  hoami)/elementaryx-test/"="$HOME/org-roam"
↪  --share="/tmp/$(whoami)/elementaryx-test/"="$HOME/referenc
↪  es"
↪  --expose=/tmp/$(whoami)/elementaryx-test/.bashrc.elementar
↪  yx=$HOME/.bashrc --expose=$HOME/.config/guix/current=/guix
↪  --expose=/tmp/.X11-unix --expose=$XAUTHORITY
↪  --expose=/etc/ssl/certs
↪  --preserve="^SHELL|^DISPLAY$|^XAUTHORITY$|^TERM$" -CNW
↪  icecat emacs-elementaryx-as-default -- bash -c
↪  'GUIX_PROFILE="/guix" ; . "$GUIX_PROFILE/etc/profile" ;
↪  emacs'
```

2.2 Required dependency: `guix`

As mentioned in Section section 1.1, `elementaryx` is composed of multiple (tiny) `git` repositories and has many dependencies (a few `emacs` dependencies, and many *non* ***`emacs`*** dependencies). As a consequence, it is not straightforward (at all) to use. Therefore we rely on a robust deployment tool – namely guix – to set it up. Note that the `guix` package manager can be installed on any `linux` distribution. We refer to guix manual for installing `guix` and assume `guix` is available in the following.

Note that `elementaryx` is not packaged in the main channel of `guix` but in the guix-hpc channel, which must therefore be enabled (see guix-hpc). This is why the channels we use include the `guix-hpc` channel.

The interested reader might read the guix definition of the present `elementaryx` suite but this is not necessary at all for proceeding.

2.2.1 Pre-built binaries

We refer to guix-hpc reference for setting up pre-built binaries.

2.2.2 Check up

In the following, we assume that `guix` and `guix-hpc` are set up.

2.3 Preliminary step: folder and file creations

We proceed with a few folder and file creations:

```
mkdir -p "$HOME/org-roam"
mkdir -p "$HOME/references"
mkdir -p "/tmp/$(whoami)/elementaryx-test"
mkdir -p "/tmp/$(whoami)/elementaryx-test/.emacs.d"
wget https://gitlab.inria.fr/elementaryx/elementaryx.gitlabpag⌋
↪  es.inria.fr/-/raw/main/dotfiles/bash-elementaryx/.bashrc.e⌋
↪  lementaryx -O
↪  "/tmp/$(whoami)/elementaryx-test/.bashrc.elementaryx"
wget https://gitlab.inria.fr/elementaryx/elementaryx.gitlabpag⌋
↪  es.inria.fr/-/raw/main/channels-fixed.scm -O
↪  "/tmp/$(whoami)/elementaryx-test/channels-fixed.scm"
```

- `$HOME/org-roam` will be used for storing non-hierarchical org-roam notes and TODOs;
- `$HOME/references` will be used for storing (`bibtex`) bibliography;
- `/tmp/$(whoami)/elementaryx-test/` will be used for testing purpose;
- `/tmp/$(whoami)/elementaryx-test/.emacs.d/` will be used as an `emacs` init directory (not necessary in an isolated container);
- `/tmp/$(whoami)/elementaryx-test/.bashrc.elementaryx` will be used as a minimalist `bashrc` set up (enable direnv, ...);
- `/tmp/$(whoami)/elementaryx-test/channels-fixed.scm` will be used as channels for `guix time-machine` or `guix pull`.

Note that an even more minimalistic `/.bashrc` setup can be used as follows (rather than the `wget` above command):

```
echo  "eval \"\$(direnv hook bash)\"" >
↪  "/tmp/$(whoami)/elementaryx-test/.bashrc.elementaryx"
```

Note also that in the long run, `$HOME/org-roam` and `$HOME/references` might instead be symbolic links to `git` repositories.

2.4 Start `elementaryx`

Assuming `guix` is available (see Section section 2.2) and the preliminary folder and file creations are done (Section section 2.3), multiple options are then available to start `elementaryx` and play with it.

The most simple way is to use the `emacs-elementaryx-as-default guix` package. This package sets up a minimalist `default.el emacs init file` (see emacs init file reference), basically consisting of `(use-package elementaryx-full)` (see source) for requesting the `emacs elementaryx` package. We also use the `icecat` package for convenient check out of `html` exports (see below for the corresponding workflow).

Here are multiple options for starting `elementaryx`, from the most conservative to the most flexible.

2.4.1 Execution within an isolated container from both the home directory and the rest of the system

1. With `guix shell` and `guix time-machine` (recommended) `elementaryx` may be executed within an isolated container (`guix shell -C`, see guix shell reference) from both the home directory and the rest of the system with guix time-machine using these guix-hpc fixed channels as follows:

```
cd "/tmp/$(whoami)/elementaryx-test/"
SHELL="bash" guix time-machine --channels=/tmp/$(whoami)/
↪  elementaryx-test/channels-fixed.scm -- shell
↪  --share=/tmp/$(whoami)/elementaryx-test/
↪  --share="$HOME/org-roam" --share="$HOME/references"
↪  --expose=/tmp/$(whoami)/elementaryx-test/.bashrc.elem
↪  entaryx=$HOME/.bashrc
↪  --expose=$HOME/.config/guix/current=/guix
↪  --expose=/tmp/.X11-unix --expose=$XAUTHORITY
↪  --expose=/etc/ssl/certs
↪  --preserve="^SHELL|^DISPLAY$|^XAUTHORITY$|^TERM$"
↪  -CNW icecat emacs-elementaryx-as-default -- bash -c
↪  'GUIX_PROFILE="/guix" ; . "$GUIX_PROFILE/etc/profile"
↪  ; emacs'
```

2. With `guix shell` but not `guix time-machine` (not recommended for a tutorial / testing purpose)

```
cd "/tmp/$(whoami)/elementaryx-test/"
```

```
SHELL="bash" guix shell
↪  --share=/tmp/$(whoami)/elementaryx-test/
↪  --share="$HOME/org-roam" --share="$HOME/references"
↪  --expose=/tmp/$(whoami)/elementaryx-test/.bashrc.elem┘
↪  entaryx=$HOME/.bashrc
↪  --expose=$HOME/.config/guix/current=/guix
↪  --expose=/tmp/.X11-unix --expose=$XAUTHORITY
↪  --expose=/etc/ssl/certs
↪  --preserve="^SHELL|^DISPLAY$|^XAUTHORITY$|^TERM$"
↪  -CNW icecat emacs-elementaryx-as-default -- bash -c
↪  'GUIX_PROFILE="/guix" ; . "$GUIX_PROFILE/etc/profile"
↪  ; emacs'
```

3. Explanations

   This execution mode ensures that the execution of `elementaryx` does not interfere with the rest of the system, nor with the home directory of the user.

   We allow only the `$HOME/org-roam`, `$HOME/references` and `/tmp/elementaryx-test` repositories created in Section section 2.3 to be writable.

   In addition, we allow nested usage of `guix` (`-W` option) and we make sure that all the channels from the main profile are made available (`--expose=$HOME/.config/guix/current` and `GUIX_PROFILE=$HOME/.config/guix/current ; . $GUIX_PROFILE/etc/profile`). We also allow for network usage (`-N` option).

   We define `bash` as the default `shell` (`SHELL="bash"` and `--expose=/tmp/$(whoami)/elementaryx` and `--preserve=^SHELL$`).

   We furthermore allow for graphical (`--expose=/tmp/.X11-unix --expose=$XAUTHORITY --preserve="^DISPLAY$|^XAUTHORITY$"`) and terminal (`--preserve=^TERM$`) execution of `emacs`. Here `emacs` (`emacs`) is started in graphical mode; but it could also be started in terminal mode (with `emacs -nw`).

   The `--expose=/etc/ssl/certs` allows for exposing system X.509 certificates (see guix manual), required for instance to authenticate web servers over `https`.

   We refer to guix shell reference for an exhaustive presentation on how to use `guix shell`.

2.4.2 Execution in an isolated container from the rest of the system but writable in the home directory

This alternative execution mode allows for interacting in a writable mode with the home directory (`--share="$HOME"`) and the `/tmp` directory (`--share=/tmp/`) but still ensures not to interfere with the rest of the system.

1. With `guix shell` and `guix time-machine`

```
SHELL="bash" guix time-machine --channels=/tmp/$(whoami)/
↪  elementaryx-test/channels-fixed.scm -- shell
↪  --share="$HOME" --share=/tmp/
↪  --expose=$HOME/.config/guix/current=/guix
↪  --expose=$XAUTHORITY --expose=/etc/ssl/certs
↪  --preserve="^SHELL|^DISPLAY$|^XAUTHORITY$|^TERM$"
↪  -CNW icecat emacs-elementaryx-as-default -- bash -c
↪  'GUIX_PROFILE="/guix" ; . "$GUIX_PROFILE/etc/profile"
↪  ; emacs
↪  --init-dir=/tmp/$(whoami)/elementaryx-test/.emacs.d/'
```

2. With `guix shell` but not `guix time-machine`

```
SHELL="bash" guix shell  --share="$HOME" --share=/tmp/
↪  --expose=$HOME/.config/guix/current=/guix
↪  --expose=$XAUTHORITY --expose=/etc/ssl/certs
↪  --preserve="^SHELL|^DISPLAY$|^XAUTHORITY$|^TERM$"
↪  -CNW icecat emacs-elementaryx-as-default -- bash -c
↪  'GUIX_PROFILE="/guix" ; . "$GUIX_PROFILE/etc/profile"
↪  ; emacs
↪  --init-dir=/tmp/$(whoami)/elementaryx-test/.emacs.d/'
```

2.4.3 Non isolated execution in a pure environment

The `--container` `guix shell` option requires kernel support. If this support is not available, an alternative is to use the `--pure` `guix shell` option. It does not ensure isolation but clears environment variable definitions to reduce (but not guarantee) unattended interaction with the rest of the system.

1. With `guix shell` and `guix time-machine`

```
SHELL="bash" guix time-machine --channels=/tmp/$(whoami)/⌋
↪ elementaryx-test/channels-fixed.scm -- shell --pure
↪ --preserve="^SHELL|^TERM$" icecat
↪ emacs-elementaryx-as-default -- bash -c
↪ 'GUIX_PROFILE="$HOME/.config/guix/current" ; .
↪ "$GUIX_PROFILE/etc/profile" ; emacs
↪ --init-dir=/tmp/$(whoami)/elementaryx-test/.emacs.d/'
```

2. With `guix shell` but not `guix time-machine`

```
SHELL="bash" guix shell --pure --preserve="^SHELL|^TERM$"
↪ icecat emacs-elementaryx-as-default -- bash -c
↪ 'GUIX_PROFILE="$HOME/.config/guix/current" ; .
↪ "$GUIX_PROFILE/etc/profile" ; emacs
↪ --init-dir=/tmp/$(whoami)/elementaryx-test/.emacs.d/'
```

2.5 Starting `elementaryx` with non standard keybindings or on a remote machine

The philosophy is to follow as much as possible the reference `emacs` 29.1. When additional packages are used, we follow as much as possible the proposed default suggestions both in terms of enabled configuration and key bindings.

2.5.1 `vim` users

There is no advanced support for `vim` but `vim` users might want to start `emacs` with the `-f evil-mode` option to enable basic `vim` key binding support:

```
emacs -f evil-mode
```

For instance, the suggested execution in Section section 2.1 may be appended as follows:

```
SHELL="bash" guix time-machine --channels=/tmp/$(whoami)/eleme⌋
↪  ntaryx-test/channels-fixed.scm -- shell
↪  --share=/tmp/$(whoami)/elementaryx-test/
↪  --share="$HOME/org-roam" --share="$HOME/references"
↪  --expose=$HOME/.config/guix/current=guix
↪  --expose=/tmp/.X11-unix --expose=$XAUTHORITY
↪  --preserve="^SHELL|^DISPLAY$|^XAUTHORITY$|^TERM$" -CNW
↪  icecat emacs-elementaryx-as-default -- bash -c
↪  'GUIX_PROFILE="/guix" ; . "$GUIX_PROFILE/etc/profile" ;
↪  emacs -f evil-mode'
```

2.5.2 `windows`-like copy-paste users (`C-c C-v`)

Users used to copy-paste with `C-c C-v` may want to start start `emacs` with `-f cua-mode`:

```
emacs -f cua-mode
```

For instance, the suggested execution in Section section 2.1 may be appended as follows:

```
SHELL="bash" guix time-machine --channels=/tmp/$(whoami)/eleme⌋
↪  ntaryx-test/channels-fixed.scm -- shell
↪  --share=/tmp/$(whoami)/elementaryx-test/
↪  --share="$HOME/org-roam" --share="$HOME/references"
↪  --expose=$HOME/.config/guix/current=/guix
↪  --expose=/tmp/.X11-unix --expose=$XAUTHORITY
↪  --preserve="^SHELL|^DISPLAY$|^XAUTHORITY$|^TERM$" -CNW
↪  icecat emacs-elementaryx-as-default -- bash -c
↪  'GUIX_PROFILE="/guix" ; . "$GUIX_PROFILE/etc/profile" ;
↪  emacs -f cua-mode'
```

2.5.3 `elementaryx` on a remote machine with mouse-tracking support

You may want to start `emacs elementaryx` on a remote machine through `ssh`. In this case, assuming `ssh` was performed with X11 forwarding (typically via `ssh -X` or `ssh -Y`), `emacs` may be started with no X (`-nw`) for performance reason, while still enabling mouse-tracking support (`-f xterm-mouse-mode`):

```
emacs -nw -f xterm-mouse-mode
```

For instance, assuming a remote machine with no the suggested execution in Section item 1 (with `--pure` option) may be appended as follows:

```
SHELL="bash" guix time-machine --channels=/tmp/$(whoami)/eleme
↪  ntaryx-test/channels-fixed.scm -- shell --pure
↪  --preserve="^SHELL|^TERM$" icecat
↪  emacs-elementaryx-as-default -- bash -c
↪  'GUIX_PROFILE="$HOME/.config/guix/current" ; .
↪  "$GUIX_PROFILE/etc/profile" ; emacs
↪  --init-dir=/tmp/$(whoami)/elementaryx-test/.emacs.d/ -nw
↪  -f xterm-mouse-mode'
```

This execution mode may be a typical usage on a remote computer where container support would not be available.

3 How to start `elementaryx` with `docker` (alternative if you don't have `guix`)?

3.1 Retrieve or generate the `docker` image

3.1.1 Option 1: Retrieve it from `docker hub` (recommended)

```
docker pull emmanuelagullo/guix-hpc-elementaryx:latest
```

3.1.2 Option 2: Generate it with `guix` (for information, and reproducibility)

```
guix_hpc_elementaryx_docker_tarball=$(guix time-machine --chan
↪  nels=/tmp/$(whoami)/elementaryx-test/channels-fixed.scm --
↪  system image --image-type=docker
↪  elementaryx-docker-image.scm)
docker load < ${guix_hpc_elementaryx_docker_tarball}
docker tag guix:latest
↪  emmanuelagullo/guix-hpc-elementaryx:latest
```

Note that this is how the image was generated before being pushed to `docker hub` with:

```
# We assume: `docker login` has been performed to pursue
docker push emmanuelagullo/guix-hpc-elementaryx:latest # image
↪  pushed online: https://hub.docker.com/r/emmanuelagullo/gui
↪  x-hpc-elementaryx
```

3.2 Create, start and execute the container

```
guix_hpc_elementaryx_docker_container=$(docker create -v
↪  /tmp/.X11-unix:/tmp/.X11-unix -e DISPLAY=$DISPLAY -e
↪  XAUTHORITY=$XAUTHORITY -e TERM=$TERM --platform
↪  linux/amd64 emmanuelagullo/guix-hpc-elementaryx:latest)
docker start $guix_hpc_elementaryx_docker_container
# As root:
# docker exec -ti $guix_hpc_elementaryx_docker_container
↪  /run/current-system/profile/bin/bash --login
# As user "panoramix":
docker exec -u panoramix -ti
↪  $guix_hpc_elementaryx_docker_container
↪  /run/current-system/profile/bin/bash --login
```

3.3 Start `emacs elementaryx`

We are now logged onto the image and can start `emacs elementaryx` from there:

```
emacs
```

3.4 Support of `X11` and `tty` modes

3.4.1 `X11` mode

The `X11` support may require some additional set up, such as `xhost +local:docker` (with `guix`: `guix shell xhost -- xhost +local:docker`). We warn the reader about potential security issues and refer to `docker` reference for further information on that topic.

3.4.2 `tty` mode

It is also possible to run in terminal mode with mouse support. We refer the reader to `docker` reference for possible additional `docker` setup. Regarding `emacs`, it can be started with:

```
emacs -nw -f xterm-mouse-mode
```

4 First steps with `elementaryx`

We assume to be all set up to play with `elementaryx`, may the user has started it with guix or docker, as discussed above.

The `elementaryx` software suite aims TO BE CONTINUED.

4.1 Graphical user interface (GUI)

As with most editors, most common operations can be performed using the graphical user interface (GUI), an important feature for discoverability.

In particular, the menu-bar allows one to