Please participate in the 2020 Elixir Ecosystem Survey!

Installing Elixir

The quickest way to install Elixir is through a distribution or using one of the available installers. If not available, then we recommend the precompiled packages or compiling it.

Note that Elixir v1.10 requires Erlang 21.0 or later. Many of the instructions below will automatically install Erlang for you. In case they do not, read the "Installing Erlang" section below.

Distributions

The preferred option for installing Elixir. Choose your operating system and tool.

If your distribution contains an old Elixir/Erlang version, see the sections below for installing Elixir/Erlang from version managers or from source.

macOS

- Homebrew
  - Update your homebrew to latest: `brew update`
  - Run: `brew install elixir`
- Macports
  - Run: `sudo port install elixir`

News: Elixir v1.10 released

Search...

UPCOMING CONFERENCES

Online | Conference 18-19 June | Training 15-17 June

Warsaw, Poland | Conference 7-8 October | Training 6 October

OFFICIAL CHANNELS

- Source code & Issues tracker
- #elixir-lang on freenode IRC
- @elixirlang on Twitter

Watch the Elixir mini-documentary!

JOIN THE COMMUNITY

- Hex.pm package manager
- Elixir Forum
- Elixir on Slack
- Elixir on Discord
- Meetups around the world
- Wiki with events and resources maintained by the community

Unix (and Unix-like)

- Arch Linux (Community repo)
    - Run: `pacman -S elixir`
- openSUSE (and SLES)
    - add Elixir/Erlang repo: `zypper ar -f obs://devel:languages:erlang/Elixir-Factory`
    - Run: `zypper in elixir`
    - optional: if you want to use the latest Erlang, you can use this repo: `zypper ar -f obs://devel:languages:erlang:Factory Erlang-Factory`
- Gentoo
    - Run: `emerge --ask dev-lang/elixir`
- GNU Guix
    - Run: `guix package -i elixir`
- Fedora 21 (and older)
    - Run: `yum install elixir`
- Fedora 22 (and newer)
    - Run `dnf install elixir`
- FreeBSD
    - From ports: `cd /usr/ports/lang/elixir && make install clean`
    - From pkg: `pkg install elixir`
- Solus
    - Run: `eopkg install elixir`
- Ubuntu 14.04/16.04/17.04/18.04/19.04 or Debian 7/8/9/10
    - Add Erlang Solutions repo: `wget https://packages.erlang-solutions.com/erlang-solutions_2.0_all.deb && sudo dpkg -i erlang-solutions_2.0_all.deb`
    - Run: `sudo apt-get update`
    - Install the Erlang/OTP platform and all of its applications: `sudo apt-get install esl-erlang`
    - Install Elixir: `sudo apt-get install elixir`
- Slackware
    - Using `sbopkg`: `sbopkg -ki "erlang-otp elixir"`

      Or

      Manually download/build/install from SlackBuilds.org: `erlang-otp`, `elixir`
- OpenBSD
    - Run: `pkg_add elixir`

Windows

- Web installer
    - Download the installer
    - Click next, next, …, finish
- Chocolatey
    - `cinst elixir`

Raspberry Pi

If necessary, replace “buster” with the name of your Raspbian release.

- The Erlang Solutions repository has a prebuilt package for armhf. This saves a significant amount of time in comparison to recompiling natively
- Get Erlang key
    - `echo "deb https://packages.erlang-solutions.com/debian buster contrib" | sudo tee /etc/apt/sources.list.d/erlang-solutions.list`
    - Run: `wget https://packages.erlang-solutions.com/debian/erlang_solutions.asc`
    - Add to keychain: `sudo apt-key add erlang_solutions.asc`
- Install Elixir
    - Update apt to latest: `sudo apt update`
    - Run: `sudo apt install elixir`

Docker

If you are familiar with Docker you can use the official Docker image to get started quickly with Elixir.

- Enter interactive mode
    - Run: `docker run -it --rm elixir`
- Enter bash within container with installed `elixir`
    - Run: `docker run -it --rm elixir bash`

Those distributions will likely install Erlang automatically for you too. In case they don’t, check the Installing Erlang section below.

If you need to programmatically fetch the list of Elixir precompiled packages alongside their checksums, access https://elixir-lang.org/elixir.csv.

Precompiled package

Elixir provides a precompiled package for every release. First install Erlang and then download and unzip the Precompiled.zip file for the latest release.

Once the release is unpacked, you are ready to run the `elixir` and `iex` commands from the `bin` directory, but we recommend you to add Elixir’s bin path to your PATH environment variable to ease development.

Compiling with version managers

There are many tools that allow developers to install and manage multiple Erlang and Elixir versions. They are useful if you can't install Erlang or Elixir as mentioned above or if your package manager is simply outdated. Here are some of those tools:

- asdf - install and manage different Elixir and Erlang versions
- exenv - install and manage different Elixir versions
- kiex - install and manage different Elixir versions
- kerl - install and manage different Erlang versions

Keep in mind that each Elixir version supports specific Erlang/OTP versions. Check the compatibility table if you have questions or run into issues.

If you would prefer to compile from source manually, don't worry, we got your back too.

Compiling from source (Unix and MinGW)

You can download and compile Elixir in few steps. The first one is to install Erlang.

Next you should download source code (.zip, .tar.gz) of the latest release, unpack it and then run `make` inside the unpacked directory (note: if you are running on Windows, read this page on setting up your environment for compiling Elixir).

After compiling, you are ready to run the elixir and `iex` commands from the bin directory. It is recommended that you add Elixir's bin path to your PATH environment variable to ease development.

In case you are feeling a bit more adventurous, you can also compile from master:

```
$ git clone https://github.com/elixir-lang/elixir.git
$ cd elixir
$ make clean test
```

If the tests pass, you are ready to go. Otherwise, feel free to open an issue in the issues tracker on Github.

Installing Erlang

The only prerequisite for Elixir is Erlang, version 21.0 or later. When installing Elixir, Erlang is generally installed automatically for you. However, if you want to install Erlang manually, you might check:

- Source code distribution and Windows installers from Erlang's official website

- Precompiled packages for some Unix-like installations
- A general list of installation methods from the Riak documentation.

After Erlang is installed, you should be able to open up the command line (or command prompt) and check the Erlang version by typing `erl`. You will see some information similar to:

```
Erlang/OTP 21.0 [64-bit] [smp:2:2] [...]
```

Notice that depending on how you installed Erlang, Erlang binaries might not be available in your PATH. Be sure to have Erlang binaries in your PATH, otherwise Elixir won't work!

Setting PATH environment variable

It is highly recommended to add Elixir's bin path to your PATH environment variable to ease development.

On **Windows**, there are instructions for different versions explaining the process.

On **Unix systems**, you need to find your shell profile file, and then add to the end of this file the following line reflecting the path to your Elixir installation:

```
export PATH="$PATH:/path/to/elixir/bin"
```

Checking the installed version of Elixir

Once you have Elixir installed, you can check its version by running `elixir --version`.