Standard Toolchain Setup for Linux and macOS

:link_to_translation:`zh_CN:[中文]`

Installation Step by Step

This is a detailed roadmap to walk you through the installation process.

Setting up Development Environment

These are the steps for setting up the ESP-IDF for your {IDF_TARGET_NAME}.

- :ref:`get-started-prerequisites`
- :ref:`get-started-get-esp-idf`
- :ref:`get-started-set-up-tools`
- :ref:`get-started-set-up-env`
- :ref:`get-started-start-a-project`

Step 1. Install Prerequisites

In order to use ESP-IDF with the {IDF_TARGET_NAME}, you need to install some software packages based on your Operating System. This setup guide helps you on getting everything installed on Linux and macOS based systems.

For Linux Users

To compile using ESP-IDF, you need to get the following packages. The command to run depends on which distribution of Linux you are using:

- Ubuntu and Debian:

```
sudo apt-get install git wget flex bison gperf python3 python3-pip python3-venv
```

- CentOS 7 & 8:

```
sudo yum -y update && sudo yum install git wget flex bison gperf python3 cmake
```

CentOS 7 is still supported but CentOS version 8 is recommended for a better user experience.

- Arch:

```
sudo pacman -S --needed gcc git make flex bison gperf python cmake ninja ccache
```

Note

- CMake version 3.16 or newer is required for use with ESP-IDF. Run "tools/idf_tools.py install cmake" to install a suitable version if your OS versions does not have one.
- If you do not see your Linux distribution in the above list then please check its documentation to find out which command to use for package installation.

For macOS Users

ESP-IDF uses the version of Python installed by default on macOS.

- Install CMake & Ninja build:
    - If you have HomeBrew, you can run:

      ```
      brew install cmake ninja dfu-util
      ```

    - If you have MacPorts, you can run:

      ```
      sudo port install cmake ninja dfu-util
      ```

    - Otherwise, consult the CMake and Ninja home pages for macOS installation downloads.
- It is strongly recommended to also install ccache for faster builds. If you have HomeBrew, this can be done via `brew install ccache` or `sudo port install ccache` on MacPorts.

Note

If an error like this is shown during any step:

```
xcrun: error: invalid active developer path (/Library/Developer/CommandLineTools),
```

Then you need to install the XCode command line tools to continue. You can install these by running `xcode-select --install` .

Apple M1 Users

If you use Apple M1 platform and see an error like this:

```
WARNING: directory for tool xtensa-esp32-elf version esp-2021r2-patch3-8.4.0 is pre
ERROR: tool xtensa-esp32-elf has no installed versions. Please run 'install.sh' to
```

or:

```
zsh: bad CPU type in executable: ~/.espressif/tools/xtensa-esp32-elf/esp-2021r2-pat
```

Then you need to install Apple Rosetta 2 by running

```
/usr/sbin/softwareupdate --install-rosetta --agree-to-license
```

Installing Python 3

Based on macOS Catalina 10.15 release notes, use of Python 2.7 is not recommended and Python 2.7 is not included by default in future versions of macOS. Check what Python you currently have:

```
python --version
```

If the output is like `Python 2.7.17` , your default interpreter is Python 2.7. If so, also check if Python 3 is not already installed on your computer:

```
python3 --version
```

If the above command returns an error, it means Python 3 is not installed.

Below is an overview of the steps to install Python 3.

- Installing with HomeBrew can be done as follows:

  ```
  brew install python3
  ```

- If you have MacPorts, you can run:

  ```
  sudo port install python38
  ```

Step 2. Get ESP-IDF

To build applications for the {IDF_TARGET_NAME}, you need the software libraries provided by Espressif in ESP-IDF repository.

To get ESP-IDF, navigate to your installation directory and clone the repository with `git clone` , following instructions below specific to your operating system.

Open Terminal, and run the following commands:

```
.. include-build-file:: inc/git-clone-bash.inc
```

ESP-IDF is downloaded into `~/esp/esp-idf` .

Consult :doc:`/versions` for information about which ESP-IDF version to use in a given situation.

Step 3. Set up the Tools

Aside from the ESP-IDF, you also need to install the tools used by ESP-IDF, such as the compiler, debugger, Python packages, etc, for projects supporting {IDF_TARGET_NAME}.

```
cd ~/esp/esp-idf
./install.sh {IDF_TARGET_PATH_NAME}
```

or with Fish shell

```
cd ~/esp/esp-idf
./install.fish {IDF_TARGET_PATH_NAME}
```

The above commands install tools for {IDF_TARGET_NAME} only. If you intend to develop projects for more chip targets then you should list all of them and run for example:

```
cd ~/esp/esp-idf
./install.sh esp32,esp32s2
```

or with Fish shell

```
cd ~/esp/esp-idf
./install.fish esp32,esp32s2
```

In order to install tools for all supported targets please run the following command:

```
cd ~/esp/esp-idf
./install.sh all
```

or with Fish shell

```
cd ~/esp/esp-idf
./install.fish all
```

Note

For macOS users, if an error like this is shown during any step:

```
<urlopen error [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: unable t
```

You may run `Install Certificates.command` in the Python folder of your computer to install certificates. For details, see Download Error While Installing ESP-IDF Tools.

Alternative File Downloads

The tools installer downloads a number of files attached to GitHub Releases. If accessing GitHub is slow then it is possible to set an environment variable to prefer Espressif's download server for GitHub asset downloads.

Note

This setting only controls individual tools downloaded from GitHub releases, it does not change the URLs used to access any Git repositories.

To prefer the Espressif download server when installing tools, use the following sequence of commands when running `install.sh` :

```
cd ~/esp/esp-idf
export IDF_GITHUB_ASSETS="dl.espressif.com/github_assets"
./install.sh
```

Note

For users in China, we recommend using our download server located in China for faster download speed.

```
cd ~/esp/esp-idf
export IDF_GITHUB_ASSETS="dl.espressif.cn/github_assets"
./install.sh
```

Customizing the Tools Installation Path

The scripts introduced in this step install compilation tools required by ESP-IDF inside the user home directory: `$HOME/.espressif` on Linux. If you wish to install the tools into a different directory, set the environment variable `IDF_TOOLS_PATH` before running the installation scripts. Make sure that your user account has sufficient permissions to read and write this path.

If changing the `IDF_TOOLS_PATH` , make sure it is set to the same value every time the Install script (`install.bat` , `install.ps1` or `install.sh`) and an Export script (`export.bat` , `export.ps1` or `export.sh`) are executed.

Step 4. Set up the Environment Variables

The installed tools are not yet added to the PATH environment variable. To make the tools usable from the command line, some environment variables must be set. ESP-IDF provides another script which does that.

In the terminal where you are going to use ESP-IDF, run:

```
. $HOME/esp/esp-idf/export.sh
```

or for fish (supported only since fish version 3.0.0):

```
. $HOME/esp/esp-idf/export.fish
```

Note the space between the leading dot and the path!

If you plan to use esp-idf frequently, you can create an alias for executing `export.sh` :

1. Copy and paste the following command to your shell's profile (`.profile` , `.bashrc` , `.zprofile` , etc.)

   ```
   alias get_idf='. $HOME/esp/esp-idf/export.sh'
   ```

2. Refresh the configuration by restarting the terminal session or by running `source [path to profile]` , for example, `source ~/.bashrc` .

Now you can run `get_idf` to set up or refresh the esp-idf environment in any terminal session.

Technically, you can add `export.sh` to your shell's profile directly; however, it is not recommended. Doing so activates IDF virtual environment in every terminal session (including those where IDF is not needed), defeating the purpose of the virtual environment and likely affecting other software.

Step 5. First Steps on ESP-IDF

Tip: Updating ESP-IDF

It is recommended to update ESP-IDF from time to time, as newer versions fix bugs and/or provide new features. Please note that each ESP-IDF major and minor release version has an associated support period, and when one release branch is approaching end of life (EOL), all users are encouraged to upgrade their projects to more recent ESP-IDF releases, to find out more about support periods, see :doc:`ESP-IDF Versions <../versions>`.

The simplest way to do the update is to delete the existing `esp-idf` folder and clone it again, as if performing the initial installation described in :ref:`get-started-get-esp-idf`.

Another solution is to update only what has changed. :ref:`The update procedure depends on the version of ESP-IDF you are using <updating>`.

After updating ESP-IDF, execute the Install script again, in case the new ESP-IDF version requires different versions of tools. See instructions at :ref:`get-started-set-up-tools`.

Once the new tools are installed, update the environment using the Export script. See instructions at :ref:`get-started-set-up-env`.

Related Documents

- :doc:`establish-serial-connection`
- Eclipse Plugin
- VSCode Extension
- :doc:`../api-guides/tools/idf-monitor`

```
.. toctree::
    :hidden:
    :maxdepth: 1

    establish-serial-connection
    flashing-troubleshooting
    ../api-guides/tools/idf-monitor
```