_(/)_

busybox (/r/_/busybox/) ☆

Last pushed: 24 days ago

Repo Info (/_/busybox/)

Short Description

Busybox base image.

Full Description

Supported tags and respective `Dockerfile` links

- `1.29.3-uclibc`, `1.29-uclibc`, `1-uclibc`, `uclibc` (_uclibc/Dockerfile_) (https://github.com/docker-library/busybox/blob/0b299265b6140985a88913133261b93d5a2c2c19/uclibc/Dockerfile)
- `1.29.3-glibc`, `1.29-glibc`, `1-glibc`, `glibc` (_glibc/Dockerfile_) (https://github.com/docker-library/busybox/blob/0b299265b6140985a88913133261b93d5a2c2c19/glibc/Dockerfile)
- `1.29.3-musl`, `1.29-musl`, `1-musl`, `musl` (_musl/Dockerfile_) (https://github.com/docker-library/busybox/blob/0b299265b6140985a88913133261b93d5a2c2c19/musl/Dockerfile)
- `1.29.3`, `1.29`, `1`, `latest` (_uclibc/Dockerfile_) (https://github.com/docker-library/busybox/blob/0b299265b6140985a88913133261b93d5a2c2c19/uclibc/Dockerfile)

Quick reference

- **Where to get help**:
  the Docker Community Forums (https://forums.docker.com/), the Docker Community Slack (https://blog.docker.com/2016/11/introducing-docker-community-directory-docker-community-slack/), or Stack Overflow (https://stackoverflow.com/search?tab=newest&q=docker)

- **Where to file issues**:
  https://github.com/docker-library/busybox/issues (https://github.com/docker-library/busybox/issues)

- **Maintained by**:
  the Docker Community (https://github.com/docker-library/busybox)

- **Supported architectures**: (more info (https://github.com/docker-library/official-images#architectures-other-than-amd64))
  `amd64` (https://hub.docker.com/r/amd64/busybox/), `arm32v5` (https://hub.docker.com/r/arm32v5/busybox/), `arm32v6` (https://hub.docker.com/r/arm32v6/busybox/), `arm32v7` (https://hub.docker.com/r/arm32v7/busybox/), `arm64v8` (https://hub.docker.com/r/arm64v8/busybox/), `i386` (https://hub.docker.com/r/i386/busybox/), `ppc64le` (https://hub.docker.com/r/ppc64le/busybox/), `s390x` (https://hub.docker.com/r/s390x/busybox/)

- **Published image artifact details**:
  repo-info repo's `repos/busybox/` directory (https://github.com/docker-library/repo-info/blob/master/repos/busybox) (history (https://github.com/docker-library/repo-info/commits/master/repos/busybox))
  (image metadata, transfer size, etc)

- **Image updates**:
  official-images PRs with label `library/busybox` (https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fbusybox)
  official-images repo's `library/busybox` file (https://github.com/docker-library/official-images/blob/master/library/busybox) (history (https://github.com/docker-library/official-images/commits/master/library/busybox))

- **Source of this description**:
  docs repo's `busybox/` directory (https://github.com/docker-library/docs/tree/master/busybox) (history (https://github.com/docker-library/docs/commits/master/busybox))

- **Supported Docker versions**:
  the latest release (https://github.com/docker/docker-ce/releases/latest) (down to 1.6 on a best-effort basis)

What is BusyBox? The Swiss Army Knife of Embedded Linux

Coming in somewhere between 1 and 5 Mb in on-disk size (depending on the variant), BusyBox (http://www.busybox.net/) is a very good ingredient to craft space-efficient distributions.

BusyBox combines tiny versions of many common UNIX utilities into a single small executable. It provides replacements for most of the utilities you usually find in GNU fileutils, shellutils, etc. The utilities in BusyBox generally have fewer options than their full-featured GNU cousins; however, the options that are included provide the expected functionality and behave very much like their GNU counterparts. BusyBox provides a fairly complete environment for any small or embedded system.

wikipedia.org/wiki/BusyBox (https://en.wikipedia.org/wiki/BusyBox)
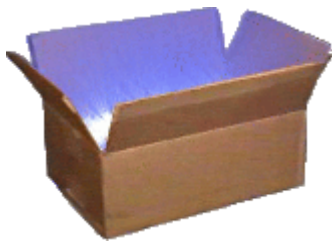
How to use this image

Run BusyBox shell

```
$ docker run -it --rm busybox
```

This will drop you into an `sh` shell to allow you to do what you want inside a BusyBox system.

Create a `Dockerfile` for a binary

```
FROM busybox
COPY ./my-static-binary /my-static-binary
CMD ["/my-static-binary"]
```

This `Dockerfile` will allow you to create a minimal image for your statically compiled binary. You will have to compile the binary in some other place like another container. For a simpler alternative that's similarly tiny but easier to extend, see `alpine` (https://hub.docker.com/_/alpine/).

Image Variants

The `busybox` images contain BusyBox built against various "libc" variants (for a comparison of "libc" variants, Eta Labs has a very nice chart (http://www.etalabs.net/compare_libcs.html) which lists many similarities and differences).

For more information about the specific particulars of the build process for each variant, see `Dockerfile.builder` in the same directory as each variant's `Dockerfile` (see links above).

busybox:uclibc

- uClibc (https://uclibc.org) via Buildroot (https://buildroot.org) (statically compiled)

busybox:glibc

- glibc from Debian (https://packages.debian.org/search?searchon=names&exact=1&suite=all&section=all&keywords=libc6) (which is then included in the image)

busybox:musl

- musl from Alpine (https://pkgs.alpinelinux.org/packages?name=musl) (statically compiled)

License

View license information (http://www.busybox.net/license.html) for the software contained in this image.

As with all Docker images, these likely also contain other software which may be under other licenses (such as Bash, etc from the base distribution, along with any direct or indirect dependencies of the primary software being contained).

Some additional license information which was able to be auto-detected might be found in the `repo-info` repository's `busybox/` directory (https://github.com/docker-library/repo-info/tree/master/repos/busybox).

As for any pre-built image usage, it is the image user's responsibility to ensure that any use of this image complies with any relevant licenses for all software contained within.

Docker Pull Command

```
docker pull busybox
```