

Docker Developer Guide

Document ID: RK-KF-YF-334

Release Version: V1.0.1

Release Date: 2020-02-18

E-mail: wxt@rock-chips.com

Security Class: ☐Top-Secret ☐Secret ☐Internal ☒Public

DISCLAIMER

THIS DOCUMENT IS PROVIDED "AS IS". FUZHOU ROCKCHIP ELECTRONICS CO., LTD. ("ROCKCHIP") DOES NOT PROVIDE ANY WARRANTY OF ANY KIND, EXPRESSED, IMPLIED OR OTHERWISE, WITH RESPECT TO THE ACCURACY, RELIABILITY, COMPLETENESS, MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT OF ANY REPRESENTATION, INFORMATION AND CONTENT IN THIS DOCUMENT. THIS DOCUMENT IS FOR REFERENCE ONLY. THIS DOCUMENT MAY BE UPDATED OR CHANGED WITHOUT ANY NOTICE AT ANY TIME DUE TO THE UPGRADES OF THE PRODUCT OR ANY OTHER REASONS.

Trademark Statement

"Rockchip", "瑞芯微", "瑞芯" shall be Rockchip's registered trademarks and owned by Rockchip. All the other trademarks or registered trademarks mentioned in this document shall be owned by their respective owners.

All rights reserved. ©2019. Fuzhou Rockchip Electronics Co., Ltd.

Beyond the scope of fair use, neither any entity nor individual shall extract, copy, or distribute this document in any form in whole or in part without the written approval of Rockchip.

Fuzhou Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian, PRC

Website: www.rock-chips.com

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: fae@rock-chips.com

Preface

Overview

This document introduces how to build a Docker environment to compile libmali, gstreamer, mpp, qt, xserver, libdrm and other deb packages on a PC with Ubuntu system, that for installation on the Debian platform.

Product ID

Chipset Name	OS Version
All chipset	Debian 9

Application Object

This document (this guide) is intended primarily for the following readers:

Field Application Engineer

Software Development Engineer

Revision History

Date	Version	Author	Change Description
2019-08-27	V1.0.0	Caesar Wang	Initial version
2020-02-18	V1.0.1	Caesar Wang	sync the style with release

Content

Docker Developer Guide

[Rockchip Docker](#)

[OS requirements](#)

[Install Docker](#)

[Build application](#)

[Modify image](#)

[Others](#)

[Examples](#)

Rockchip Docker

[Docker](#) is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package. By doing so, thanks to the container, the developer can rest assured that the application will run on any other Linux machine regardless of any customized settings that machine might have that could differ from the machine used for writing and testing the code.

In a way, Docker is a bit like a virtual machine. But unlike a virtual machine, rather than creating a whole virtual operating system, Docker allows applications to use the same Linux kernel as the system that they're running on and only requires applications be shipped with things not already running on the host computer. This gives a significant performance boost and reduces the size of the application.

And importantly, Docker is [open source](#). This means that anyone can contribute to Docker and extend it to meet their own needs if they need additional features that aren't available out of the box.

The Rockchip Docker open source is on [docker-rockchip](#).

OS requirements

To install Docker, you need the 64-bit version of one of these Ubuntu versions:

- Bionic 18.04 (LTS)
- Xenial 16.04 (LTS)
- Trusty 14.04 (LTS)

Note: the dockerfile is used for arm64 socs by default.

Below is used for arm32 Socs: cp dockerfile-32 dockerfile

Install Docker

- Use this command to install the latest version of Docker(replace docker with docker.io in ubuntu 14.04):

```
sudo apt-get install docker qemu-user-static binfmt-support
```

- Start run Docker daemon:

```
sudo service docker start
```

- Build Docker image by dockerfile:

```
sudo docker build -t rockchip .
```

- Now you get a Docker image named "rockchip" which include a debian multiarch cross-compiling enviroment.

Build application

- Enter docker shell:

```
docker run -it -v <package dir>:/home/rk/packages rockchip /bin/bash
```

- Start build:

for arm 32-bit Socs :

```
cd /home/rk/packages/<package-name>  
DEB_BUILD_OPTIONS=nocheck dpkg-buildpackage -rfakeroot -b -d -uc -us -aarmhf  
ls ../ | grep *.deb
```

for arm 64-bit Socs :

```
cd /home/rk/packages/<package-name>  
DEB_BUILD_OPTIONS=nocheck dpkg-buildpackage -rfakeroot -b -d -uc -us -aarm64  
ls ../ | grep *.deb
```

Modify image

If you want to modify your Docker image, you can open a shell by below command:

```
docker run -it rockchip /bin/bash
```

After exit from container, you should use below command to save your changes.

```
docker commit <container_id> rockchip
```

Others

To get more informations about dockers, please check below link: <https://docs.docker.com>

Examples

- How to get the libmali-rk-midgard-t86x-r14p0_1.7-1_arm64.deb on [libmali](#)

```
wxt@nb:~/work/docker/docker-rockchip$ sudo service docker start
wxt@nb:~/work/docker/docker-rockchip$ sudo docker build -t rockchip .
wxt@nb:~/work/docker/docker-rockchip$ sudo docker run -it -v
/home/wxt/work:/home/rk/packages rockchip /bin/bash
rk@2888134f9c12:/$ cd /home/rk/packages/docker/libmali
rk@2888134f9c12:~/packages/docker/libmali$ DEB_BUILD_OPTIONS=nocheck dpkg-
buildpackage -rfakeroot -b -d -uc -us -aarm64
```

The above steps will get the debs for ~/packages/docker/

- Rockchip had uploaded some source code for building and generating the deb packages

[libdrm](#)

[libmali](#)

[openbox](#)

[xserver](#)

[qt](#)

[gststreamer-rockchip](#)

[mpp](#)