

QDOS development tools docker container

☆ 2 stars 🍴 0 forks

☆ Star

👁 Watch

<> Code

🕒 Issues

🔗 Pull requests

🎬 Actions

📁 Projects

🛡 Security

📈 Insights

🔗 main ▾

...



xXorAa README.md : more fixes/clarifications ...

on Jan 10 ⌚ 13

[View code](#)

☰ README.md

🔗 qdos-devel docker image

As setting up old versions of gcc gets harder as the future goes on here is a Dockerfile that sets up qdos-gcc in a i386 debian buster (10.10) container.

It also sets up xtc68 as well from the latest source from github.

Also included is make/qlzip/unzip/vim so actual development can be carried out in the container.

Building

on an x86_64/x86_32 machine if you have an older version of docker

```
DOCKER_BUILDKIT=1 docker build -t <user>/<label> .
```

If you have a docker that support buildx multi platform, you can build multiple platforms at once as a cross compile.

```
docker buildx build --tag <user>/<label>:<version> --platform  
linux/amd64,linux/386,linux/arm/v7 --load .
```

Obviously you can fill in your own details above, or remove the --push if you don't want to push to docker hub.

Using

For example using the docker image to build the wander game for qdos

```
git clone https://github.com/SinclairQL/wander  
cd wander  
git checkout qdos-port  
cd ..  
docker run -v `pwd`/wander:qdos/wander -it <user>/<label> bash
```

You should now be in the container in the directory /qdos/

```
cd wander  
make CC=qdos-gcc  
make deploy
```

This should make a Wander.zip which can be used in an emulator.

qdos-gcc note

According to the documents C produced object files are not compatible between c68 and gcc. So this container separates the libs into different directories. This means gcc is missing some libraries that c68 has.

You can cheat and use

```
-L/usr/local/share/qdos/lib
```

in the link instruction line to use c68 libs, but you are on your own with compatibility.

Releases

No releases published

Packages

No packages published

Languages

● Dockerfile 51.6% ● Shell 48.4%