# LinX Software Suite DevEnv 4

Quick Start Guide

# Revision history

| Rev | Date | Author | Comments |
|---|---|---|---|
| 1 | 2019-04-23 | Anders Svedberg | Initial document |
| 2 | 2019-06-26 | Anders Svedberg | Updates based on version 4.0.3 |

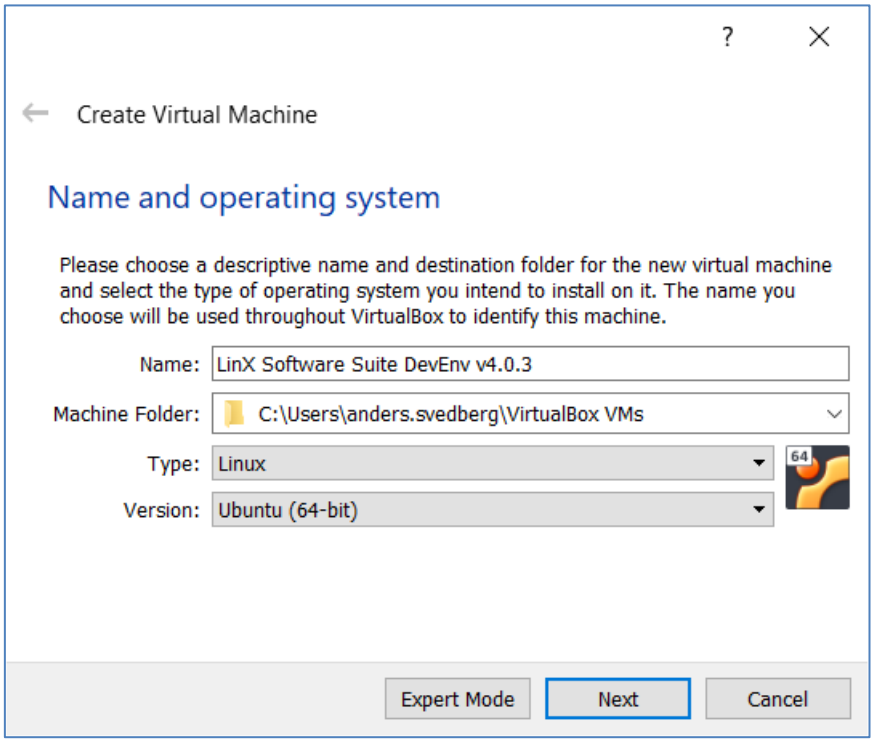
# Contents

**Revision history** ..... **2**
**1. Initial setup** ..... **3**
1.1. Install Guest additions ..... 5

# 1. Initial setup

To setup LinX Software Suite DevEnv v4.0.3 you need to create a new virtual machine in VirtualBox. We recommend to use VirtualBox 6.0 or higher. CrossControl provides the virtual disk image but you need to configure the machine yourself according to your hardware specifications.

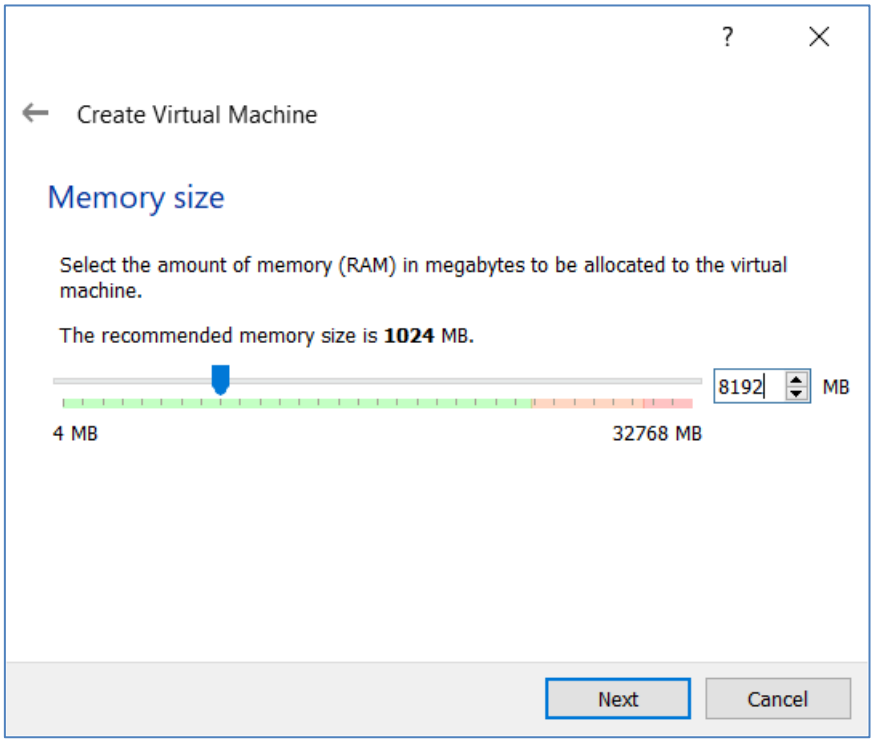
Step 1. Open VirtualBox and select [Machine] -> [New...]



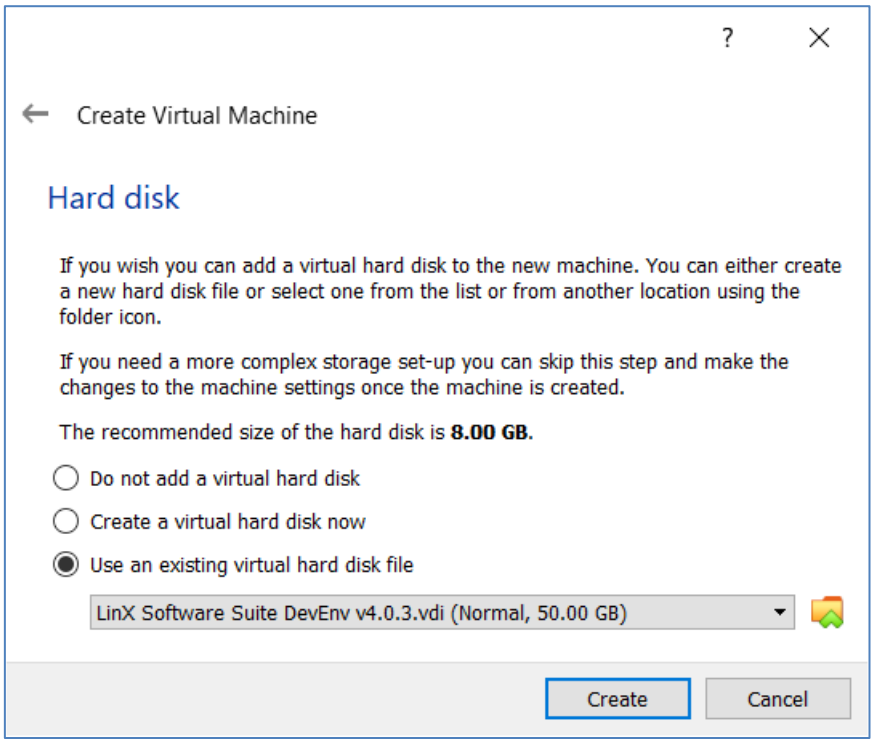

Select:

- Enter a Name
- Type: Linux
- Version Ubuntu (64-bit)

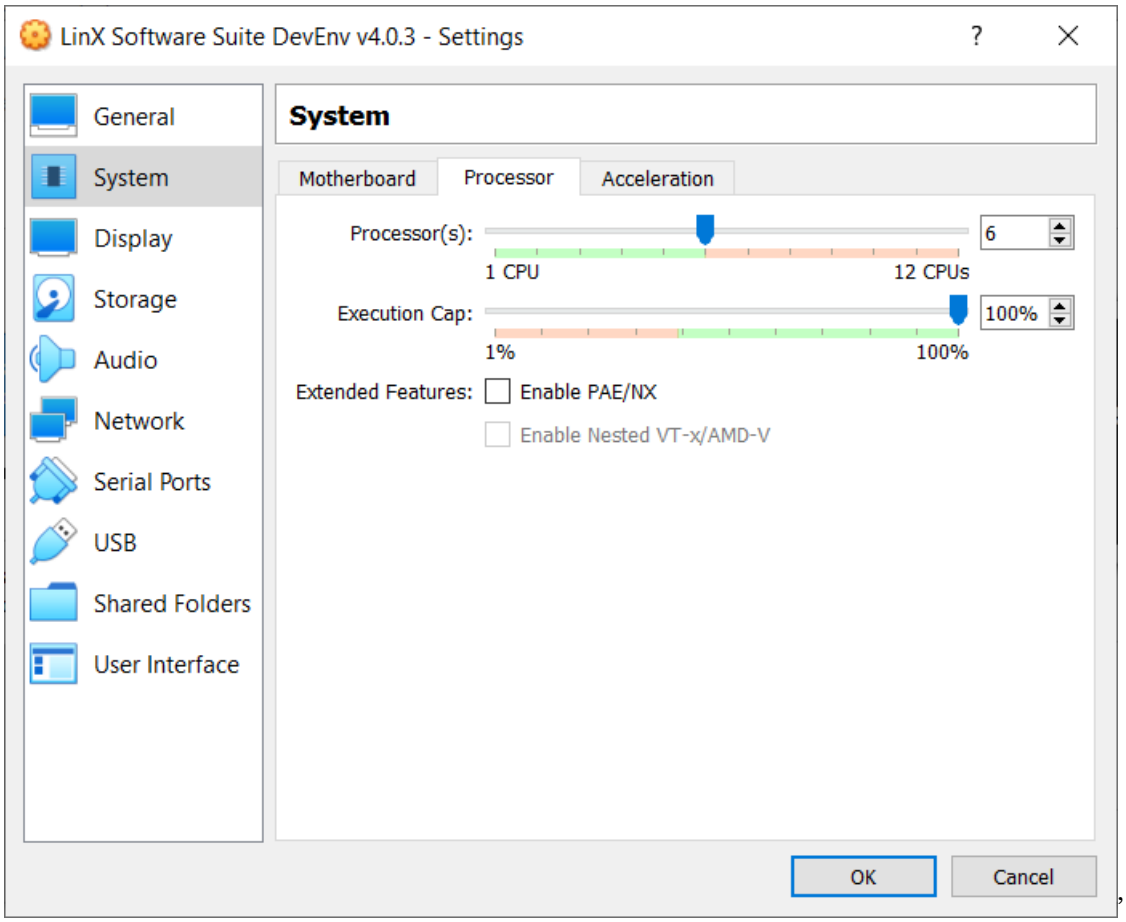
Step 2. Select the amount of memory you want to assign to your virtual machine.




Step 3. Select [Use an existing virtual hard disk file] and press the yellow icon to point out the location of the extracted version of "LinX Software Suite DevEnv v4.0.3.vdi"



Step 4: Complete the wizard and select the new virtual machine. Press *"System"* tab and set the numbers of processors you want to assign. More CPUs will speed up compiling.




Step 5: boot up the virtual machine.

Login information:

- User: *ccs*
- Password: *default*

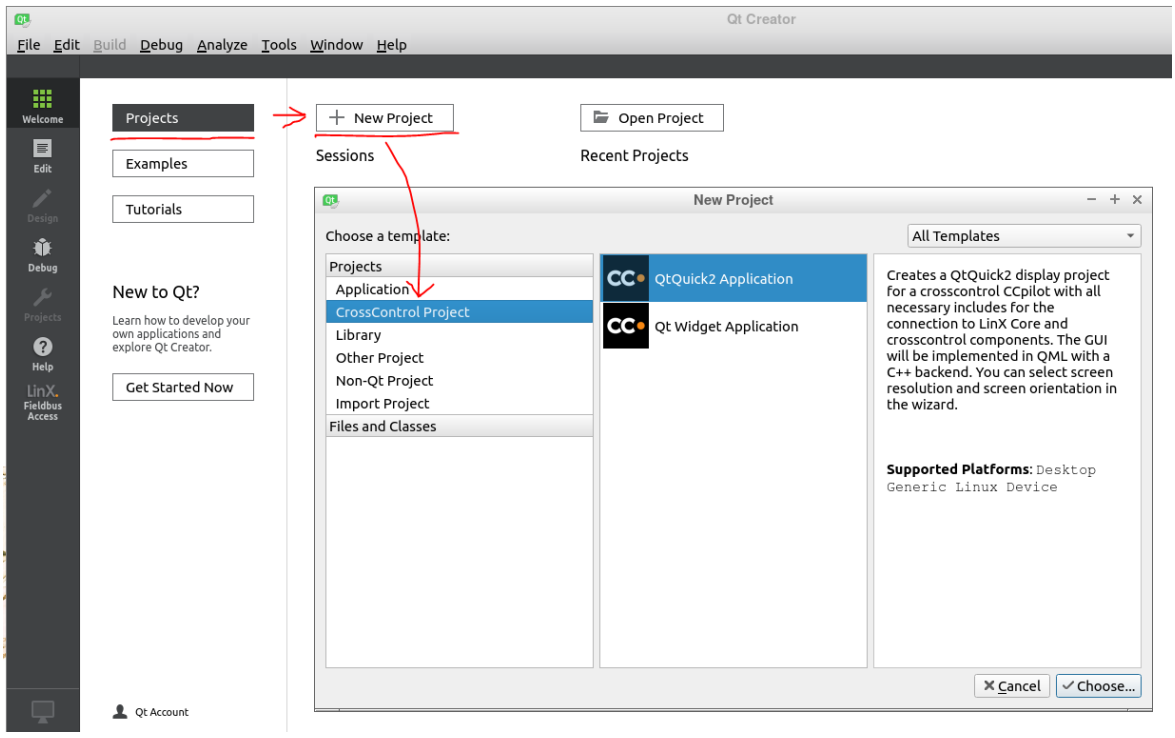
## 1.1. Install Guest additions

It is recommended to install the [VirtualBox Guest additions] in the new virtual machine.
[Devices] -> [Insert Guest Additions CD image…]
The virtual disk image will be mounted at /media/ccs folder. It must be executed as superuser:
*ccs@ubuntu:/media/ccs/VBox_GAs_6.0.4$ sudo ./VBoxLinuxAdditions.run*
*[sudo] password for ccs: default*

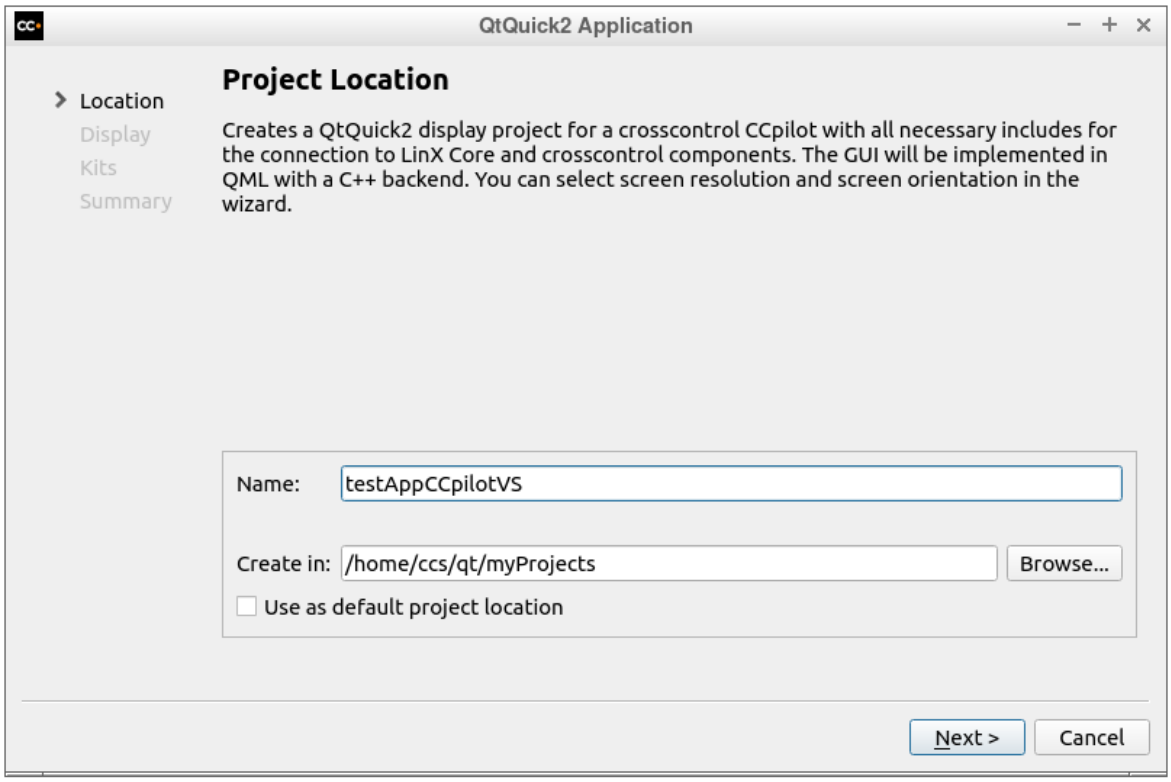
## 2. Getting started with Qt

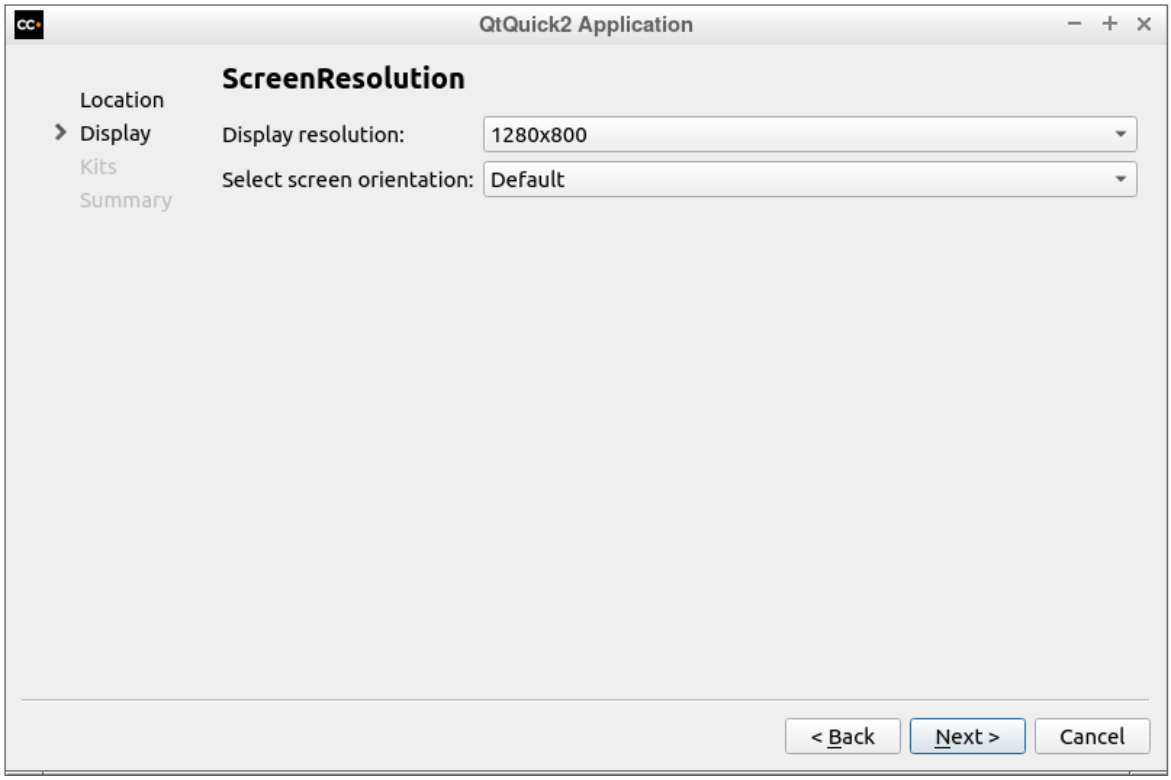
A standard installation of Qt-5.12.0 from the Qt Company is installed in the virtual machine. CrossControl provides pre-compiled Qt runtimes for all Linux-based CCpilot devices in this virtual machine. QtCreator comes with build kits for each device.

CrossControl also provides template applications for our display devices, and it is recommended to use these templates when targeting a CCpilot device. The template projects set up dependencies and include paths for the different targets.



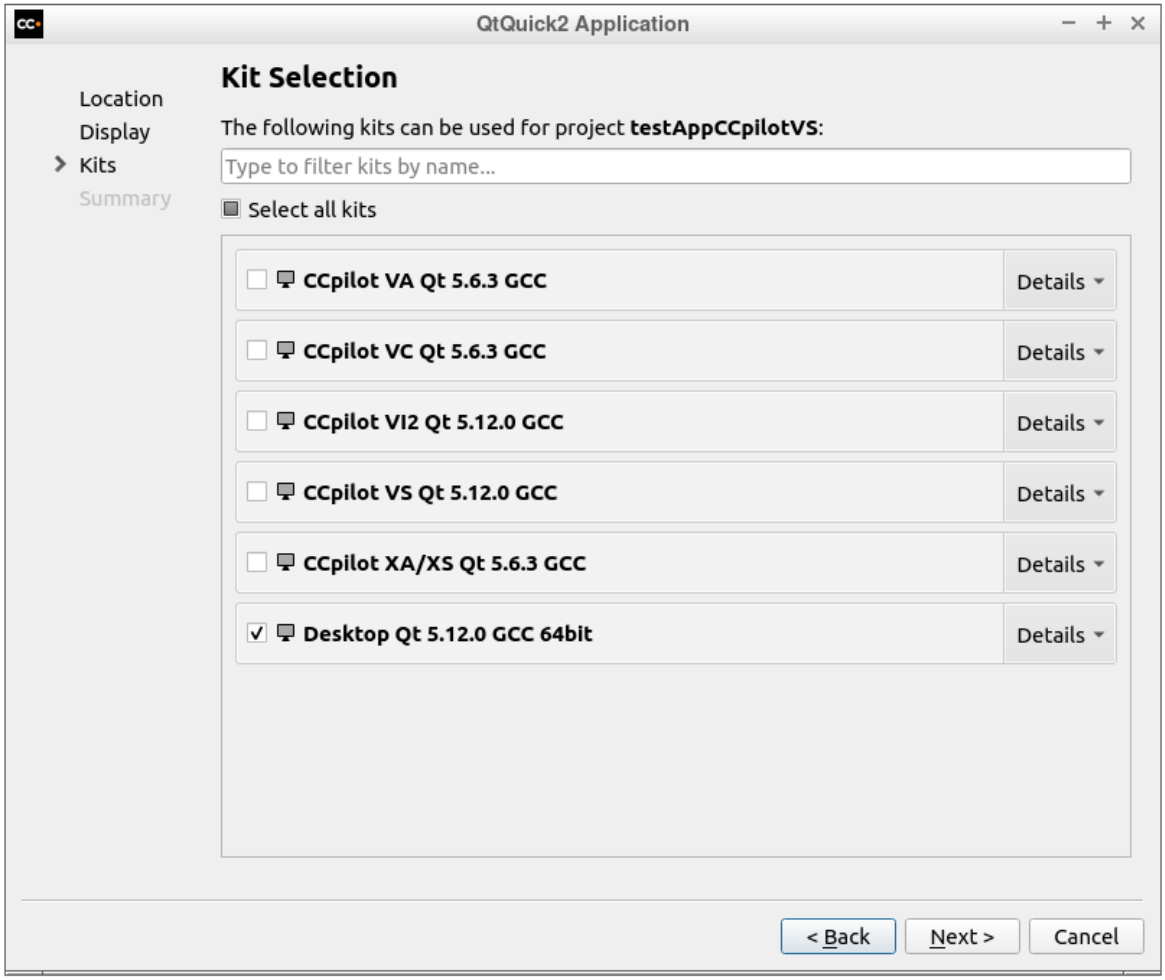

Select QML or Widget based application and press *[Choose...]*



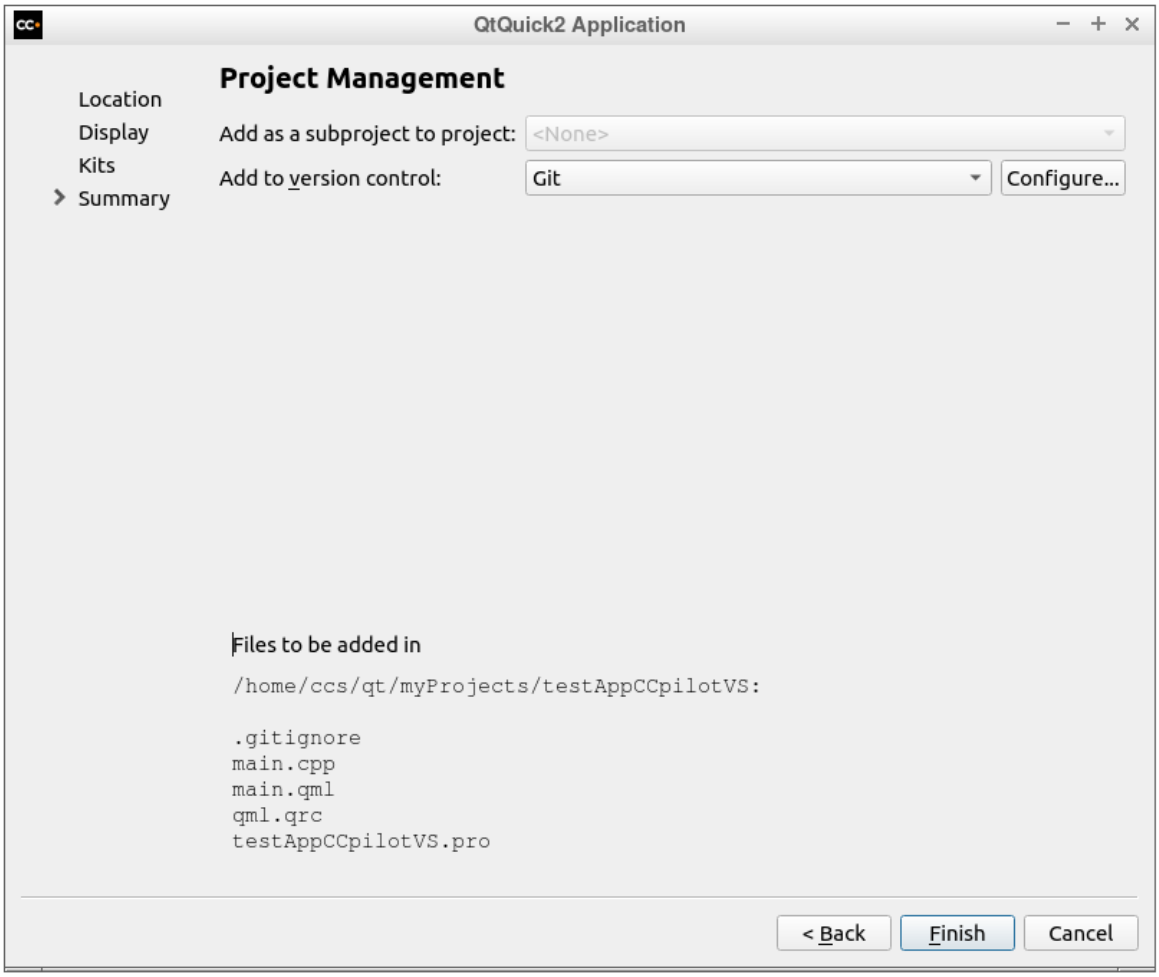




Then select the “Kit”. The Kit is the setup for the Qt runtimes, one kit for each CCpilot device. Several kits can be selected if you want to target several display types. Kits can easily be added later as well.

It is recommended to select the “Desktop” kit as well for running the application in the virtual machine.



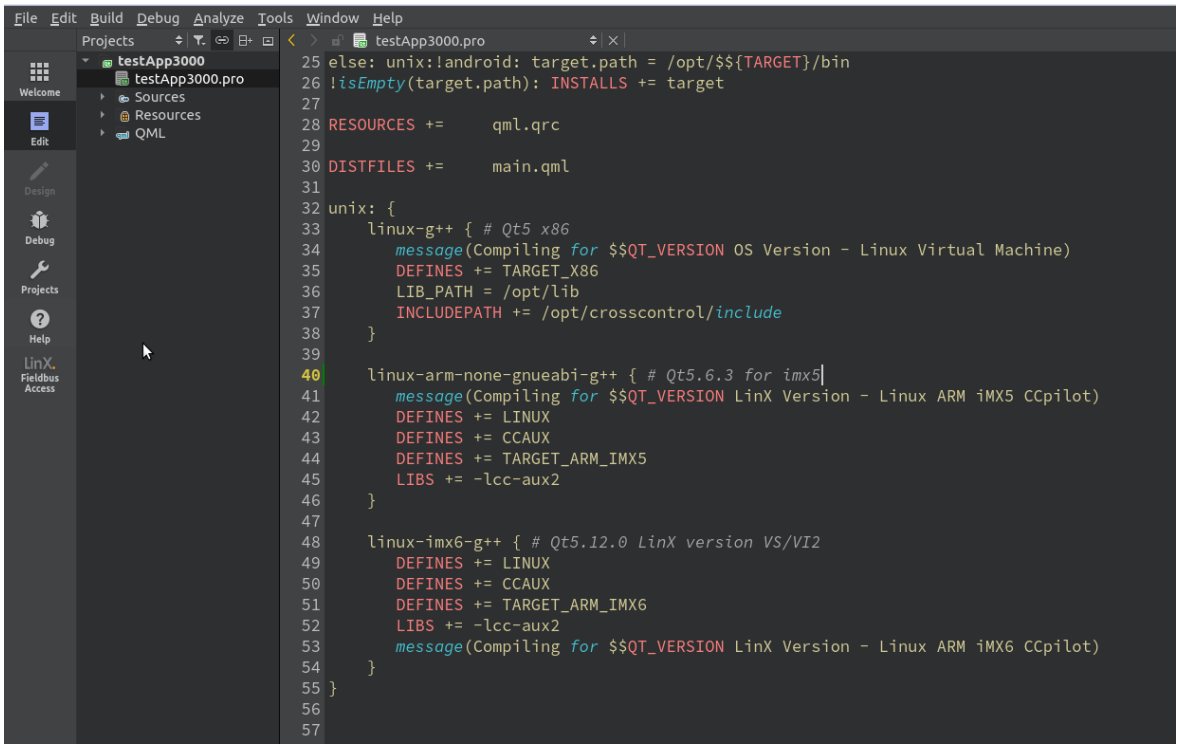

Last page in the wizard is the project management. Git is pre-installed in the virtual machine and useful for version control of your application. Press [*Finish*] to complete the wizard.




## 2.1. CrossControl Template project file

The template project creates a .PRO file with pre-configured search paths for each target device. If you are converting another Qt project, please look at the template file in VM4 to see how you should setup the project file to work with the new runtimes / virtual machine.

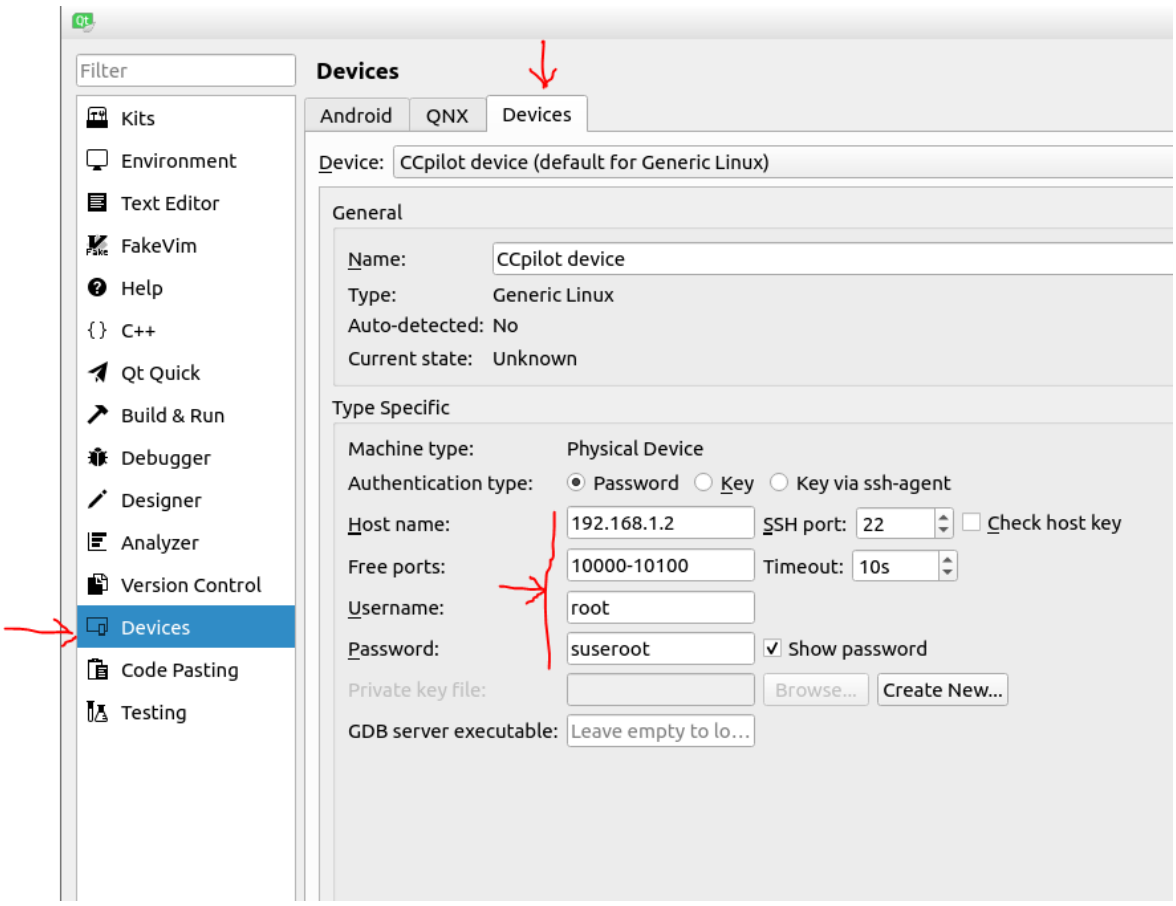
In the picture below, there are several blocks of code, one for each build kit. Make sure to add additional include paths and/or dependencies in the correct block.




## 2.2. Setup a target device

To start developing an application with LinX software suite 4, it is recommended to connect the development machine and the CCpilot device on the same network and use Ethernet to deploy and remote debug.

In QtCreator → Tools → Options menu, it is possible to configure the IP address to the CCpilot display.




It is recommended to do development as root user to be able to access all parts of the OS. The application will also be executed as root by the autostart script.

## 2.3. Enable root SSH login to CCpilot VS/VI2 devices (development machines only)

The application will be executed as root when auto started via the auto start script. For security reasons, root access via SSH is by default disabled on these devices, but during development it is good to turn this on to be able to access all functionality on the device.

To turn on root SSH access, follow these instructions:

1. Log in to the device as user ccs / default.
2. Switch to superuser with command: *sudo su*. Enter password *default*.
3. Mount file system writable with the command: *mount –o remount,rw /*
4. Edit the SSH config file: *nano /etc/ssh/sshd_config*

5. Find the line #PermitRootLogin yes and remove the # in front of the line

```
# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

6. Make file system read only again with command: *sudo mount –o remount,ro /*
7. Restart the device with command: *reboot*
8. Now it should be possible to login and deploy applications with username: *root* password: *suseroot*

# 3. Running a Qt application using 5.12.0 runtime on CCpilot VS/VI2

## 3.1. Using platform Wayland (default)

The Qt 5.12.0 runtime for VS/VI2 uses the Wayland platform as default platform since the CCpilot VS/VI2 runs wayland with Weston as a window manager as default configuration.

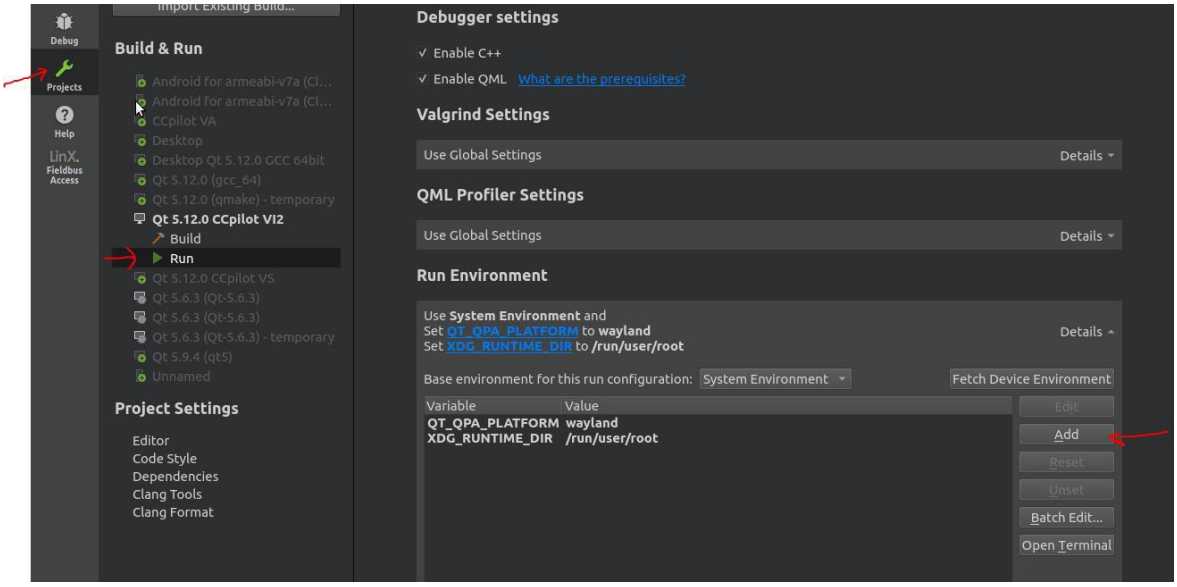
Weston makes it possible to run several applications side by side, but with some graphical performance penalty. Animations and framerate is higher with platform EGLFS so it is a tradeoff between functionality and performance and the setup is different for each customer.

To run a Qt application in Weston (default) the following environment flags must be set:

*XDG_RUNTIME_DIR=/run/user/root*

*QT_QPA_PLATFORM=wayland*

These environment parameters can be set from Qt Creator → run environment




If the application is started from a bash script, the environment variables must also be set first before starting the Qt application.

```
root@vs:/opt/testApp3000/bin# export XDG_RUNTIME_DIR=/run/user/root
root@vs:/opt/testApp3000/bin# export QT_QPA_PLATFORM=wayland
root@vs:/opt/testApp3000/bin# ./testApp3000
QML debugging is enabled. Only use this in a safe environment.
Using Wayland-EGL
Using the 'xdg-shell-v6' shell integration
```

## 3.2. Using platform EGLFS

If multi window functionality is not needed, then EGLFS platform can be used. Then Weston window manager should be stopped on the CCpilot VS/VI2 device. For testing, this can be done with the following command:

*sudo /etc/init.d/weston stop*

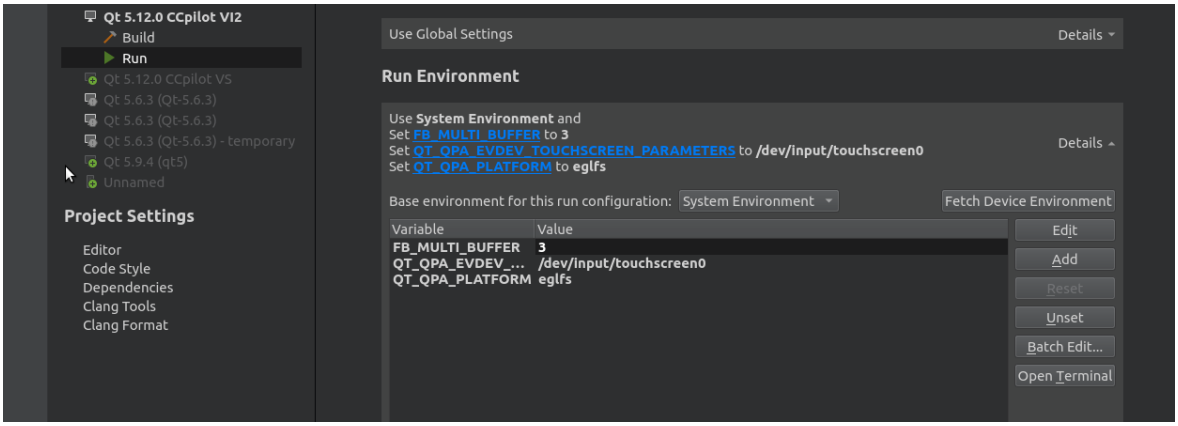
The following environment variables should be set before running a Qt application with platform EGLFS:

*QT_QPA_PLATFORM=eglfs*

*QT_QPA_EVDEV_TOUCHSCREEN_PARAMETERS=/dev/input/touchscreen0*

*FB_MULTI_BUFFER=3*

These environment parameters can be set from Qt Creator → run environment.




If the application is started from a bash script, the environment variables must also be set first before starting the Qt application.

```
root@vs:/opt# export QT_QPA_PLATFORM=eglfs
root@vs:/opt# export QT_QPA_EVDEV_TOUCHSCREEN_PARAMETERS=/dev/input/touchscreen0
root@vs:/opt# export FB_MULTI_BUFFER=3
root@vs:/opt# /opt/testApp3000/bin/testApp3000
QML debugging is enabled. Only use this in a safe environment.
```

See https://doc.qt.io/qt-5/embedded-linux.html for more details if needed. If you want to disable Weston window manager permanently at boot, please contact support@crosscontrol.com for detailed instructions.

# 4. Copy Qt runtime libraries to target device

CrossControl provides Qt runtime libraries for each device in a package called LinX-base. These are available from our support web page: support.crosscontrol.com

It is also possible to copy the Qt runtime libraries from virtual machine sysroot folders to the CCpilot device each sysroot folder (for example /opt/VA/Qt-5.6.3) to /opt/Qt-5.6.3 at the target device.

For iMX6 devices, use the following command to copy (change to correct IP):

*rsync -av /opt/VS/sysroots/cortexa9hf-neon-poky-linux-gnueabi/opt/Qt-5.12.0 ccs@192.168.1.2:/opt*

# 5. Known issues

## 5.1. QtMultimedia is not working with Qt 5.12.0 runtime on CCpilot VS / VI2 on OS 1.4.1.0

A library is missing in OS version 1.4.1.0 for the CCLinux devices. To be able to use QtMultimedia, this library must be manually copied from the virtual machine to the device.

To be able to copy the library to the correct place on the device, the read only file system needs to be unmounted.

1. Log in to the display using username: *ccs* password: *default*

2. Mount file system writable with the command: *sudo mount –o remount,rw /*

3. Copy the missing library from the virtual machine to the device with this command: *rsync -av /opt/VS/sysroots/cortexa9hf-neon-poky-linux-gnueabi/usr/lib/libpulse-mainloop-glib* root@x.x.x.x:/usr/lib*

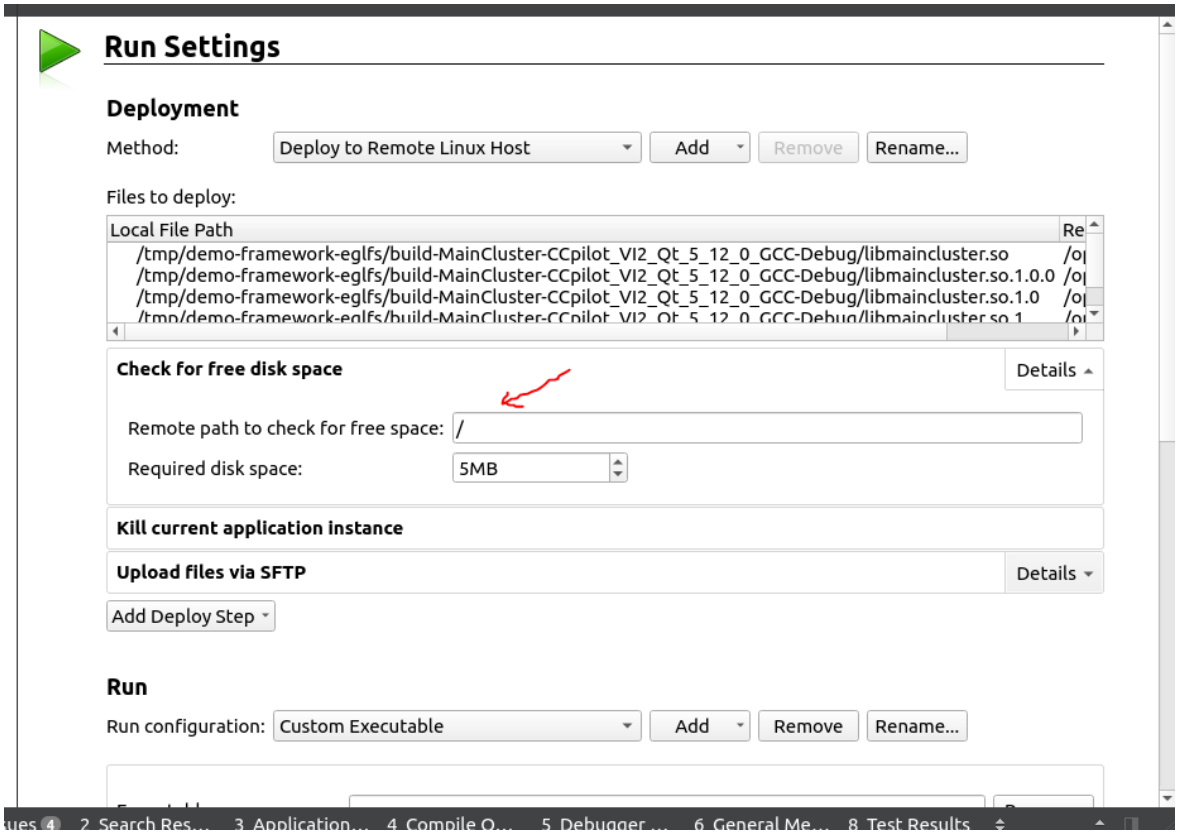
4. Make file system read only again with command: *sudo mount –o remount,ro /*

This library will be added in next OS release and this step is only needed for OS version 1.4.1.0.

## 5.2. Qt Creator – check for free disk space

Qt Creator has a built-in check for free disk space when deploying an application to an embedded Linux device. The CCpilot devices has the read only root partition at / and the writable partition mounted at /opt instead.

The remote path to check for free disk space should therefore be changed to *"/opt"* instead of *"/"*.


Run Settings
Deployment
Method:
Deploy to Remote Linux Host
Add
Remove
Rename...
Files to deploy:
Local File Path
/tmp/demo-framework-eglfs/build-MainCluster-CCpilot_VI2_Qt_5_12_0_GCC-Debug/libmaincluster.so
/tmp/demo-framework-eglfs/build-MainCluster-CCpilot_VI2_Qt_5_12_0_GCC-Debug/libmaincluster.so.1.0.0
/tmp/demo-framework-eglfs/build-MainCluster-CCpilot_VI2_Qt_5_12_0_GCC-Debug/libmaincluster.so.1.0
Check for free disk space
Details
Remote path to check for free space:
/
Required disk space:
5MB
Kill current application instance
Upload files via SFTP
Details
Add Deploy Step
Run
Run configuration:
Custom Executable
Add
Remove
Rename...