

LinX DevEnv Suite 4 Quick Start Guide Revision 2 2019-06-26

### 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

Re	Revision history			
1	Initial setup	3		
••				
	1.1. Install Guest adaltions	. 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...]

				?	×
← Create Virtua	I Machine				
Name and c	perating system	ı			
Please choose a and select the ty choose will be us	descriptive name and de pe of operating system y red throughout VirtualBo	estination fo /ou intend t x to identify	older for the new o install on it. Th v this machine.	virtual m le name y	nachine /ou
Name:	LinX Software Suite Dev	/Env v4.0.3			
Machine Folder:	C:\Users\anders.sv	edberg\Vir	tualBox VMs		$\sim$
Туре:	Linux			-	64
Version:	Ubuntu (64-bit)			•	
	Exp	ert Mode	Next	Ca	incel

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.

	?	×
← Create Virtual Machine		
Memory size		
Select the amount of memory (RAM) in megabytes to be allocated machine.	l to the virtu	ial
The recommended memory size is <b>1024</b> MB.		
	8192	De MB
4 MB 3276	8 MB	
Next	Ca	incel

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"

	?	$\times$
← Create Virtual Machine		
Hard disk		
If you wish you can add a virtual hard disk to the new machine. You a new hard disk file or select one from the list or from another locat folder icon.	can eithe ion using	er create the
If you need a more complex storage set-up you can skip this step ar changes to the machine settings once the machine is created.	id make t	he
The recommended size of the hard disk is 8.00 GB.		
O not add a virtual hard disk		
Create a virtual hard disk now		
Use an existing virtual hard disk file		
LinX Software Suite DevEnv v4.0.3.vdi (Normal, 50.00 GB)		- 🧔
Create	Ca	incel

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.

😳 LinX Software Suite	DevEnv v4.0.3 - Settings ? ×
General	System
System	Motherboard Processor Acceleration
Display	Processor(s):
Storage	1 CPU 12 CPUs Execution Cap:
🕩 Audio	1% 100%
Network	Enable Nested VT-x/AMD-V
Serial Ports	
<i> </i> USB	
Shared Folders	
User Interface	
	OK Cancel

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 ...]

cc•	QtQuick2 Application	- + ×
<b>N</b> Leasting	Project Location	
> Location Display Kits Summary	Creates a QtQuick2 display project for a crosscontrol CCpilot with all necessary in the connection to LinX Core and crosscontrol components. The GUI will be implem QML with a C++ backend. You can select screen resolution and screen orientation wizard.	cludes for iented in in the
	Name: testAppCCpilotVS	
	Create in: /home/ccs/qt/myProjects	Browse
	Use as default project location	
	<u>N</u> ext >	Cancel
<b>~</b>	QtQuick2 Application	- + ×
	ScreenResolution	
Location Display	Display resolution: 1280x800	-
Kits	Select screen orientation: Default	
Summary		
	< Back Next >	Cancel

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.

CC•	QtQuick2 Application	- + ×
Location	Kit Selection	
Display	The following kits can be used for project <b>testAppCCpilotVS</b> :	
Kits	Type to filter kits by name	
Summary	Select all kits	
	□ 🖵 CCpilot VA Qt 5.6.3 GCC	Details 👻
	□ 🖵 CCpilot VC Qt 5.6.3 GCC	Details 👻
	CCpilot VI2 Qt 5.12.0 GCC	Details 👻
	□ 🖵 CCpilot VS Qt 5.12.0 GCC	Details 👻
	□ 🖵 CCpilot XA/XS Qt 5.6.3 GCC	Details 👻
	✓ 🖵 Desktop Qt 5.12.0 GCC 64bit	Details 👻
	< <u>B</u> ack Nex	t > Cancel

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.

Project Management         Location         Display         Kits         Add as a subproject to project:         Add to version control:         Git	
Display     Add as a subproject to project: <none>       Kits     Add to version control:     Git     Config</none>	
Kits     Add to version control:     Git     Config       > Summary     Summary     Summary     Summary     Summary     Summary	~
	jure
Files to be added in	
/home/ccs/qt/myProjects/testAppCCpilotVS:	
.gitignore main.cop	
main.qml	
qm1.qrc testAppCCpilotVS.pro	
< <u>B</u> ack <u>F</u> inish Ca	ancel

#### 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.

<u>F</u> ile <u>E</u> dit	<u>File E</u> dit <u>B</u> uild <u>D</u> ebug <u>A</u> nalyze <u>T</u> ools <u>W</u> indow <u>H</u> elp					
	Projects 🗢 🕈 🖼 🕀 🗄					
	testApp3000	25 else: unix:!android: target.path = /opt/\$\${TARGET}/bin				
Welcome	E testApp3000.pro	<pre>26 !isEmpty(target.path): INSTALLS += target</pre>				
	<ul> <li>Resources</li> </ul>					
Edit	🕨 🦏 QML	28 RESOURCES += dml.qrc				
		29 30 DISTETLES += main gm]				
Decien		31				
Design		32 unix: {				
Ŵ		33 linux-g++ { # 0t5 x86				
Debug		34 message(Compiling for \$\$QT_VERSION OS Version - Linux Virtual Machine)				
لكر		35 DEFINES += TARGET_X86				
Projects		36 LIB_PATH = /opt/lib				
•		37 INCLUDEPATH += /opt/crosscontrol/include				
Help	N	38 }				
LinX.	<b></b>	$39$ linux-arm-papa-gaugabi-gtt { # 0+5 6 2 for imvel				
Fieldbus Access		41 message(Compiling for \$ \$ 000 UP ( ) UP (				
		42 DEFINES += LINUX				
		43 DEFINES += CCAUX				
		44 DEFINES += TARGET_ARM_IMX5				
		45 LIBS += -lcc-aux2				
		47				
		48 linux-imx6-g++ { # Qt5.12.0 LinX version VS/VI2				
		49 DEFINES += LINUX				
		30 DEFINES T- CLAUA 51 DEFINES += TADGET ADM TMYG				
		$51 \qquad \text{JIRS} + = -1 \text{ cr-aux}^2$				
		53 message(Compiling for \$\$0T VERSION LinX Version - Linux ARM iMX6 CCpilot)				
		54 }				
		55 }				
		57				

#### 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  $\rightarrow$  Tools  $\rightarrow$  Options menu, it is possible to configure the IP address to the CCpilot display.

\_

Image: Nils       Image: Control         Image: Provide the state       Image: Provide the state         Image: Provide the state <th>TH Kitc</th> <th></th>	TH Kitc	
■ Text Editor       General         ▲ FakeVim       Name:       CCpilot device         ④ Help       Type:       Generic Linux         ↓ C++       Auto-detected: No       Current state:       Unknown         ✓ Qt Quick       Type Specific         ▲ Build & Run       Type Specific         ▲ Debugger       Machine type:       Physical Device         ▲ Authontication type:       Password ○ Key via ssh-agent         Host name:       192.168.1.2       SSH port:       22 \$         ♥ Version Control       Username:       root       Password       Suseroot         ♥ Devices       Password:       suseroot       Show password       Private key file:         ▲ Testing       GDB server executable:       Leave empty to lo       Create New	Environment	Device: CCpilot device (default for Generic Linux)
Mare:       CCpilot device         Image:       Comparison         Image:       CCpilot device         Image:       Image:         Image:       Unknown         Image:       Unknown         Image:       Physical Device         Image:       Image:         Image:	Text Editor	General
<ul> <li>Pelp</li> <li>Help</li> <li>Type: Generic Linux</li> <li>Auto-detected: No</li> <li>Current state: Unknown</li> <li>Type Specific</li> <li>Type Specific</li> <li>Machine type: Physical Device</li> <li>Authentication type: Physical Device</li> <li>Authentication type: Password ○ Key ○ Key via ssh-agent</li> <li>Host name: 192.168.1.2 SSH port: 22 ‡ Check host key</li> <li>Version Control</li> <li>Username: root</li> <li>Devices</li> <li>Password: suseroot</li> <li>Show password</li> <li>Private key file:</li> <li>Browse Create New</li> </ul>	👫 FakeVim	Name: CCpilot device
<ul> <li>{) C++</li> <li>↓ Qt Quick</li> <li>↓ Build &amp; Run</li> <li>↓ Debugger</li> <li>↓ Designer</li> &lt;</ul>	Help	Type: Generic Linux
✓ Qt Quick       Current state:       Unknown         ✓ Build & Run       Type Specific         Image: Debugger       Machine type:       Physical Device         ✓ Designer       Authentication type:       Password ○ Key ○ Key via ssh-agent         Image: Device       Host name:       192.168.1.2       SSH port:       22 ♀ ○ Check host key         Image: Devices       Free ports:       10000-10100       Timeout:       10s ♀         Image: Devices       Password:       suseroot       ✓ Show password         Image: Devices       Password:       Suseroot       ✓ Show password         Image: Devices       Private key file:       Browse       Create New         Image: Devices       GDB server executable:       Leave empty to lo       Image: Device motion	{} C++	Auto-detected: No
<ul> <li>➢ Build &amp; Run</li> <li>Type Specific</li> <li>Machine type: Physical Device</li> <li>Authentication type: Password <u>K</u>ey Key via ssh-agent</li> <li>Host name: 192.168.1.2 SSH port: 22 ↓ Check host key</li> <li>Version Control</li> <li>Version Control</li> <li>Username: Password: Suseroot</li> <li>Code Pasting</li> <li>Fresting</li> <li>GDB server executable: Leave empty to low</li> </ul>	🖈 Qt Quick	Current state: Unknown
<ul> <li>i Debugger</li> <li>Machine type: Physical Device</li> <li>Authentication type: Physical Device</li> <li>Authentication type: Password ○ Key via ssh-agent</li> <li>i Host name: 192.168.1.2 SSH port: 22 ‡ Check host key</li> <li>i Pree ports: 10000-10100 Timeout: 10s ‡</li> <li>i Devices</li> <li>Password: Suseroot</li> <li>i Show password</li> <li>Private key file: Browse Create New</li> </ul>	➤ Build & Run	Type Specific
✓ Designer       Authentication type:       ● Password ○ Key ○ Key via ssh-agent         Image: Password ○ Key ○ Key via ssh-agent       Host name:       192.168.1.2       SSH port:       22 ‡ ○ Check host key         Image: Pree ports:       192.168.1.2       SSH port:       22 ‡ ○ Check host key         Image: Pree ports:       10000-10100       Timeout:       10s ‡         Image: Password:       Password:       root       Image: Password         Image: Password:       Private key file:       Browse       Create New         Image: Private key file:       Image: Private key file:       Image: Private key file:       Image: Private key file:	Debugger	Machine type: Physical Device
Image: Free ports:       192.168.1.2       SSH port:       22       ↓       Check host key         Image: Version Control       Image: Free ports:       10000-10100       Timeout:       10s       ↓         Image: Version Control	🖊 Designer	Authentication type:   Password <u>Key</u> Key via ssh-agent
Image: Second control       Free ports:       10000-10100       Timeout:       10s         Image: Devices       Username:       root       Image: Devices       Image: Devices         Image: Devices       Password:       suseroot       Image: Devices       Image: Devices         Image: Devices       Private key file:       Image: Devices       Image: Devices       Image: Devices         Image: Devices       Private key file:       Image: Devices       Image: Devices       Image: Devices         Image: Devices       ODB server executable:       Leave empty to low:       Image: Devices       Image: Devices	Analyzer	Host name: 192.168.1.2 SSH port: 22 Check host key
□ Devices       Username:       root         □ Devices       Password:       suseroot         □ Code Pasting       Private key file:       Browse         □ Testing       GDB server executable:       Leave empty to lo	Version Control	Free ports: 10000-10100 Timeout: 10s
Password:       suseroot       ✓ Show password         I Code Pasting       Private key file:       Browse       Create New         I Testing       GDB server executable:       Leave empty to lo	Devices	Username: root
Private key file:     Browse     Create New       Image: Comparison of the second seco	Code Pasting	Password: Show password
GDB server executable: Leave empty to low	A Testing	Private key file: Browse Create New
	<u> </u>	GDB server executable: Leave empty to lo

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



- 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  $\rightarrow$  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/touchscreeno FB\_MULTI\_BUFFER=3

These environment parameters can be set from Qt Creator  $\rightarrow$  run environment.

	Qt 5.12.0 CCpilot V12	Use Global Settings Run Environment	Details 👻
	■ Qt 5.6.3 (Qt-5.6.3)         ■ Qt 5.6.3 (Qt-5.6.3)         ■ Qt 5.6.3 (Qt-5.6.3) - temporary         ■ Qt 5.6.3 (Qt-5.6.3) - temporary         ■ Qt 5.6.4 (qt5)         ■ Unnamed         Project Settings	Use System Environment and Set <u>FB_MULT1_BUFFER</u> to 3 Set <u>OT_OPA_EVDEV_TOUCHSCREEN_PARAMETERS</u> to /dev/input/touchscreen Set <u>OT_OPA_PLATFORM</u> to eglfs	10 Details A
		Base environment for this run configuration: System Environment	Fetch Device Environment
			Edit
	Editor	OT OPA EVDEV /dev/input/touchscreen0	Add
	Code Style Dependencies	QT_QPA_PLATFORM eglfs	
	Clang Tools		
	Clang Format		
			<u>B</u> atch Edit
			Open <u>T</u> erminal

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 <a href="mailto:support@crosscontrol.com">support@crosscontrol.com</a> 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:/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 "/".

**CrossControl** Box 83, SE-822 22 ALFTA Fax +46 271 75 76 89 Varmvalsvägen 13 SE-721 30 VÄSTERÅS Fax + 46 21 40 32 10

Deployment		
Method:	Deploy to Remote Linux Host   Add   Remove Rename	
Files to deploy:		
Local File Path		Re
/tmp/demo-fr /tmp/demo-fr /tmp/demo-fr	amework-eglfs/build-MainCluster-CCpilot_VI2_Qt_5_12_0_CCC-Debug/libmaincluster.sc amework-eglfs/build-MainCluster-CCpilot_VI2_Qt_5_12_0_GCC-Debug/libmaincluster.sc amework-edlfs/build-MainCluster-CCpilot_VI2_Qt_5_12_0_GCC-Debug/libmaincluster.sc	0,1.0.0 /oj 0.1.0 /oj 0.1 /oj
Check for free	disk space	Details 🔺
Remote path to	o check for free space: /	
Required disk s	space: 5MB 🗘	
Kill current app	lication instance	
Upload files via	SFTP	Details -
Add Deploy Step	•	
Run		