LinX Software Suite

Getting Started
Contents
Revision history ...........................................................................................................2
Glossary ......................................................................................................................2
1. Brief Introduction .................................................................................................3
2. Components and Installation ..............................................................................3
   2.1. Virtual Box ....................................................................................................3
   2.2. Virtual Development Machine ....................................................................3
   2.3. Set-up of the virtual machine ......................................................................5
3. Inside the Development Environment ...............................................................6
   3.1. Basic Information .........................................................................................6
   3.2. User and Password .....................................................................................6
   3.3. Applications and files provided in the virtual machine ...............................7
       3.3.1. Pre-installed applications ......................................................................7
       3.3.2. Pre-installed files and resources ............................................................7
4. Creating the first Qt5 application ......................................................................8
   4.1. Create an empty project ................................................................................8
   4.2. Connecting to a CCpilot VC .........................................................................11
5. Deploy and run on an CCpilot VC ....................................................................13
6. Autostart settings on CCpilot device ...............................................................14
   6.1. Modify startup scripts ..................................................................................14
   6.2. Change order of startup ..............................................................................15
7. Further reading ..................................................................................................16
8. Appendix: Installing the LinX Base runtime on a CCpilot ..............................17
   8.1. Download the LinX Base runtime for your display ......................................17
   8.2. Transfer the runtime installation file to the CCpilot ..................................17
   8.3. Install the runtime on the CCpilot ..............................................................17

Revision history

<table>
<thead>
<tr>
<th>Rev</th>
<th>Date</th>
<th>Author</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2014-12-10</td>
<td>J. Wendebaur</td>
<td>First Version</td>
</tr>
<tr>
<td>1.1</td>
<td>2014-12-20</td>
<td>J. Wendebaur</td>
<td>Small adjustments and corrections</td>
</tr>
<tr>
<td>1.2</td>
<td>2015-03-16</td>
<td>J. Wendebaur</td>
<td>Updated for new Beta version of VM</td>
</tr>
<tr>
<td>1.3</td>
<td>2015-03-18</td>
<td>J. Melin</td>
<td>Small adjustments and corrections</td>
</tr>
<tr>
<td>1.4</td>
<td>2015-05-20</td>
<td>K. Lindfors</td>
<td>Updated for LinX Software Suite 2.0.2</td>
</tr>
<tr>
<td>1.5</td>
<td>2016-12-14</td>
<td>D.Nisses-Gagnér</td>
<td>Updated company name, password info</td>
</tr>
<tr>
<td>1.6</td>
<td>2017-09-08</td>
<td>D.Nisses-Gagnér</td>
<td>Updated broken link to support site</td>
</tr>
</tbody>
</table>

Glossary

<table>
<thead>
<tr>
<th>Word/Abreviation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CrossTecc</td>
<td>A set of applications and components...</td>
</tr>
<tr>
<td>crc</td>
<td>Cyclic redundancy check</td>
</tr>
<tr>
<td>SAP</td>
<td>Software Application Platform</td>
</tr>
<tr>
<td>Qt</td>
<td>Development framework qt.digia.com</td>
</tr>
<tr>
<td>RTE</td>
<td>RunTime Environment</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>VDM</td>
<td>Virtual Development Machine</td>
</tr>
</tbody>
</table>

www.crosscontrol.com
1. Brief Introduction

The LinX Software Suite Development Environment is a development tool chain running under Linux running within a virtual machine. This guide provides a guide for installing the virtual machine and getting a first QtQuick application running on a maximatecc CCpilot device.

This guide is based on LinX Software Suite Development Environment v2.0.2 or later.

2. Components and Installation

2.1. Virtual Box

In order to be able to execute the virtual machine, the application “Virtual Box” version 4.3 or higher has to be installed on your local computer. Virtual Box is available at https://www.virtualbox.org. Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and is licensed as Open Source under the GNU General Public License V2 (http://www.gnu.org/licenses/gpl-2.0.html).

The full documentation about Virtual Box is available online at https://www.virtualbox.org/manual/ch01.html.

Please refer to the extensive official troubleshooting guide or our knowledge base if you have trouble when installing or running Virtual Box,

2.2. Virtual Development Machine

The virtual development machine (VDM) called “LinX Software Suite Development Environment” can be downloaded from the download section at http://support.crosscontrol.com/downloads/LinX_Software_Suite (you need to register on the site in order to access the downloads).

The complete virtual machine is over 2 GB large. Therefore it can be downloaded either as

- a single Zip-file, unpack it to any convenient place on your hard drive (about 12 GB is needed)
- a split Zip-file consisting of many smaller files (7-Zip is needed, and about 12 GB is needed)

---

1 The previous versions of the maximatecc virtual development environments used the VMWare Player which was depreciated with this version. However, if you want to continue to use the older versions of the virtual machines, you need to use VMWare, as the Virtual Box software is not able to run those older environments.

www.crosscontrol.com
2.3. Set-up of the virtual machine

Start the Virtual Box application:

Figure 1: Virtual Box GUI

Under the “Machine” menu press on “Add” and open the file “LinX Software Suite Development Environment v2.0.x” from the VDM archive. The GUI should look like this:

Figure 2: Virtual Box with Virtual Development Machine loaded

Review the settings for the VDM and adapt them to your needs.

Press on ▶️ to boot up the virtual machine.

During the first boot up, you will need to read and accept the software license in order to use the LinX Software Suite Development Environment.
3. Inside the Development Environment

After booting the development environment you get to the standard Linux desktop:

![Development environment desktop](image)

**Figure 3: Development environment desktop**

3.1. Basic Information

The virtual machine is running a standard kubuntu 14.04. You can use all the standard features of Kubuntu, there are no limitations from maximatecc to this environment.

You can get further help about kubuntu at [kubuntu.org](http://kubuntu.org).

Sooner or later you will need to use the Linux shell through a terminal window. Open it by clicking on the terminal button in the lower left corner:

![Terminal icon](image)

**Figure 4: Terminal icon**

A guide for using the shell can be found at [wikibooks.org](http://wikibooks.org).

You can change the keyboard layout by pressing the language-switch button in the lower right corner of the screen (should automatically be set to host language, otherwise default is “se”):

![Keyboard layout switcher](image)

**Figure 5: Keyboard layout switcher**

3.2. User and Password

In the virtual machine you will login with the default user account “ccs”. This account is the default root account (used for sudo access)

- User: **ccs**
- Password: **default**
### 3.3. Applications and files provided in the virtual machine

Open your personal “Home” folder by choosing the “File Manager” in the start menu:

![Figure 6: “Places” menu](image)

Inside your “Home” folder, you can find a folder `qt` which contains a folder `MyProjects` which is the default location for new development projects.

#### 3.3.1. Pre-installed applications

- **Qt Framework** 4 and 5
- **Qt Creator** 3.3.0 (Qt5)
- **CoDeSys** run-time 3.5.4
- maximattecc **DataEngineServer** 2.0.9
- **Firebird** Database Server

#### 3.3.2. Pre-Installed files and resources

The folder `/opt` contains further resources for development and deployment and other additional files:

- `/opt/bin`: pre-compiled SAP binaries
- `/opt/codesourcery`: cross-compiler resources for compilation of applications for ARM based products
- `/opt/CoDeSysControl`: the CoDeSys-Run-time for running inside the Virtual Machine
- `/opt/crosscontrol`: a link to the folder mtcc for compatibility reasons
- `/opt/etc`: custom configuration files for the virtual machine
- `/opt/lib`: all libraries needed for compiling applications for the virtual machine and the x86-based products
- `/opt/mtcc/db`: Firebird database storage folder
- `/opt/mtcc/images`: Image resources for Qt components and widgets
- `/opt/mtcc/include`: all include files for compiling Qt applications
- `/opt/mtcc/symbols`: vector based, partly ISO-conform symbols
- `/opt/XA`: contains all binaries compiled for the ARM platform
4. Creating the first Qt5 application

In this chapter you can find a step-by-step guide for creating a simple Qt5-application and download and run it on a CCpilot VC.

4.1. Create an empty project

Start the Qt Creator by either double-clicking the “Qt Creator” icon on the desktop or by selecting “Qt Creator” in the “Programming” sub-menu under “Applications”.

![Figure 7: Qt Creator start menu item](image)

After the Qt Creator has started, press either Ctrl-N or choose “New File or Project...” from the File menu. In the appearing dialog, choose “maximatecc projects” on the left side and “QtQuick2” in the right list.

![Figure 8: “New Project” dialogue](image)

Press “Choose...”, and enter a name for the project in the next step and press on “Next >”. 

www.crosscontrol.com
The next windows will present you with all the Kits available in the virtual machine. They contain pre-configured settings for the different platforms.

As we are going to run a Qt5-application on a CCpilot VC, choose both the “CCpilot ARM Qt5 Development” and the “Virtual Development Qt5” and press “Next >”

![Kit Selection](image)

**Figure 9: Kit Selection**

On the next page, you can select the target’s screen resolution and screen orientation.

![Application Information](image)

**Figure 10: Application Information**

Press “Next >” and “Finish”.

A project will be created with a few lines of QML-code:

```
import QtQuick 2.4
import QtQuick.Window 2.2
import QtQuick.Controls 1.3

Window {
    id: base
    visible: true

    // this will provide automatic adaption to screen size and
    // orientation
    property bool orientationPortrait: false
    property bool targetARM: false // do not change this line

    width: (orientationPortrait) ? 480 : 600;
    height: (orientationPortrait) ? 320 : 600;

    // this will provide automatic adaption to screen size
    // and orientation
    Rectangle {
        id: view
        rotation: (targetARM & orientationPortrait) ? 90 : 0;
        anchors.centerIn: parent

        // add your QML code below this line
        text: "Hello"
    }
}
```

**Figure 11: Template project**

www.crosscontrol.com
You can directly execute the application in the Virtual Machine. Click on the “Kit”-button in the bottom left corner of your screen and select “Virtual Development Qt5”.

![Kit selection shortcut](image12.png)

Press on “Run” in the “Build”-menu, F5 or Ctrl-R:

![Run command](image13.png)

or on the green “Run”-Arrow in the lower left corner to start the application.

The application will compile and start up, looking like this:

![Template project running in virtual machine](image14.png)

Congratulations, you have just created your first QtQuick / QML application!
4.2. Connecting to a CCpilot VC

Be sure to have the CCpilot connected to both power and an Ethernet port which is accessible from your PC (the CCpilot uses DHCP by default, connecting the CCpilot to your PC directly will not work). The Standard GUI on the CCpilot will show its IP-Address in the lower right corner:

![CCpilot Startup Launcher GUI](image)

Figure 15: IP address showing on the CCpilot Startup Launcher GUI

Open the options menu by clicking on “Options...” in the “Tools” menu:

![Options… menu item](image)

Figure 16: Options… menu item

To set up the IP-address of your unit, click on “Devices” in the left column, and choose the “CCpilot ARM Device” in the drop-down field at the top of the window.

Enter the IP address of your unit in the “Host name” field:

![“Devices” window of the QtCreator](image)

Figure 17: “Devices” window of the QtCreator
Press on the “Test” button on the right side to the test connection:

![SSH connection test](image)

**Figure 18: SSH connection test**

Close the window by pressing “Ok”

**Troubleshooting tips if you can’t connect:**

1. Make sure that you can connect from Windows to your CCpilot by opening the start menu and run the command “cmd” (just press the windows-key and enter the letters c m d and press enter):

![Windows run window](image)

**Figure 19: Windows run window**

This should bring up a console window. Enter the command “ping” and the ip-address of the CCpilot and press enter. The output should look like this:

![“Ping” in the Windows console window](image)

**Figure 20: “Ping” in the Windows console window**

If not, check the Ethernet connection between your PC and the CCpilot. Ask your local IT support for help if not sure how to do this.

2. If you can connect to the CCpilot from windows, check if you can access it from a console window in the virtual machine by opening a console and entering the same “ping” command as above (press Ctrl-C to cancel):

![“Ping” in the Linux console window](image)

**Figure 21: “Ping” in the Linux console window**

If this does not work, check the Ethernet configuration of the virtual machine. Refer to the “Virtual Box” manual for further help.

3. Make sure you have activated the network connection from the virtual machine. If the network symbol looks like in the picture below. Click on it and activate it.

![If you see this symbol, click it and activate network](image)

**Figure 22: If you see this symbol, click it and activate network**
5. Deploy and run on an CCpilot VC

Click on the “Projects”-tab on the left side of the Qt Creator screen:

The precondition for this process is the availability of the LinX Base runtime on the display. See chapter 8 for instructions on how to install the runtime.

Check the compiler settings by clicking on “Manage Kits...” button next to the Kits tables. The list of the available kits should look like this:

![Figure 23: Tool bar in the QtCreator](image)

Make sure that the compiler settings look like this:

![Figure 24: Build and Run settings](image)

Close the dialogue by pressing “Ok”

Go to the “Run”-Settings for the CCpilot ARM target. At the run-settings, add “-plugin tslib” to the run arguments:

![Figure 26: “Run” settings tab in the project settings](image)
After changing from the X86-compiler to the ARM-compiler, you need to rebuild the project. Choose “Rebuild all” from the “Build” menu:

Press on the run button. The application should compile and start on the CCpilot. You must close the application by pressing the “Exit” button before you can deploy a new version to the display. After closing the application, the application window might stay on the screen because there might be no other application running anymore which updates the screen buffer.

Congratulations, you are running your first QtQuick / QML application on a CCpilot!

6. Autostart settings on CCpilot device

All CCpilot display devices come preinstalled with an application, CCSSettings, which automatically starts during boot. This is a Qt4 application located in the device folder:

```
/opt/bin/StartupLauncherGui
```

This application holds a few setting and test tools together with current assigned IP Adress.

```
[CC Settings    Touch Calibrator    CC Video ]
[Terminal      Rotate Display      Shutdown]
```

6.1. Modify startup scripts

In Linux the startup procedure looks thru a couple of files to decide what application, drivers and scripts to start and in what order. In following directory of the device we find the scripts,

- cd /opt/etc/init.d
- ls
These files are installed during installation of packages. LinX-Base will install StartupGuiQt5 for example. StartupGui is the default script from the OS image and the one that starts the default CCSettings application “StartupLauncherGui”. **StartupGui** is the script responsible for Qt4 Gui applications and **StartupGuiQt5** is responsible for Qt5 (Widget and Quick/QML) GUI applications. If we would have our QML **Qt5** application starting during boot instead of the default CCSettings follow these steps.

First stop StartupGui from starting (the CCSettings application)

- `chmod -x StartupGui`

Second make the StartupGuiQt5 script “executeable” instead

- `chmod +x StartupGuiQt5`

The StartupGuiQt5 scripts needs to be edited and point to what application to start. Edit the script with an editor, we will use nano in this example to point to “Demo_App” stored in “/opt/DemoApplication”.

- `nano StartupGuiQt5`

Once opened, you need to modify the script. There is a section right at the beginning with a comment " END USER: Replace this with your own application..." followed by two lines where you need to add the directory of your application, and also the name of the application.

Here is an example how it can look like once modified:

```
#!/bin/sh
./etc/init.d/cross-common

# END USER: Replaces this with your own application. Take care using correct Qt flags.
# APPLICATION_PATH="/opt/DemoApplication/"
# APPLICATION="Demo_App"
# Set touch-area (Needed by VC-Qt5 applications)
```

**Snippet of StartupGuiQt5 script**

- Ctrl + O (“WriteOut” to save your file in nano)

- Ctrl + X (Close nano editor)

Restart your device (# reboot) and your application should show up on the display.

If we instead have a **Qt4** application we wouldn’t need to change the execution rights (chmod) but instead just edit the StartupGui script directly.

### 6.2. Change order of startup

The applications starts in a timed order during boot and this is controlled by symbolic links in folder /opt/etc/rc3.d

Follow these steps to change the order of your startup application (if needed).

- `cd /opt/etc/rc3.d`
This is symbolic links pointing to the actual startup-script in `/opt/etc/init.d`

For example if we look at S04StartupGui with

```
$ ls
```

S04StartupGui point to `/usr/local/etc/init.d/StartupGui` (same as `/opt/etc/init.d/StartupGui`)

To change it to start after Data Engine during boot rename the link to S98StartupGui (in this example).

```
$ mv S04StartupGui S98StartupGui
```

Now our application in StartupGui will start after Data Engine (starting at S97).

### 7. Further reading

We recommend the following resources for further reading:

- The official Qt documentation at [qt-project.org](http://qt-project.org)
- *Qt5 Cadaques - A Book about Qt5*
- [Linux Guide at wikibooks.org](http://wikibooks.org)
- And of course the knowledge base at the [maximatecc support site](http://maximatecc.com).
8. Appendix: Installing the LinX Base runtime on a CCpilot

When you order a product with LinX Base (UX Designer and Data Engine) the required runtime will be pre-installed.

If you still need to download this runtime package manually, follow the steps below.

8.1. Download the LinX Base runtime for your display

In the virtual machine, open the Firefox web browser and navigate to the maximat support site. Under “Product Information” > “Software” > “UX Designer” select and download the LinX-Base installation file for your display from the section “LinX-Base runtimes”.

You can see the download progress in the top right corner of the window. Wait for the download to complete. By default, the files get downloaded to the folder “Downloads” in your home folder.

8.2. Transfer the runtime installation file to the CCpilot

In the virtual machine, open a terminal window (see chapter 3.1).

Change to the download folder by entering

```
cd Downloads
```

Make sure that you know the IP address of your CCpilot (see chapter 4.2).

Enter the following command in the terminal window, replacing LinX-Base-XXX.run with the proper filename of the file you just downloaded and #.#.#.# with the IP address of the CCpilot:

```
scp LinX-Base-XXX.run root@#.#.#.#:/opt
```

Write “yes” if you get asked “Are you sure you want to continue connecting (yes/no)?”

When asked for the password, enter the root-password “suseroot”.

8.3. Install the runtime on the CCpilot

While still in the terminal window, enter

```
ssh root@#.#.#.#
```

where #.#.#.# is the IP address of your CCpilot.

When logged in with the password “suseroot”, enter

```
cd /opt
chmod +x ./LinX-Base-XXX.run
```

Don’t forget to replace “LinX-Base-XXX.run” with the filename of the installation file.

The runtime installation takes around one minute and will finish by listing the installed packages.

Reboot the unit by writing

```
reboot
```

---

2 You need to be registered and logged in on the support site in order to access the downloads.

www.crosscontrol.com