



## Download and Install CCS5 for Linux™

This procedure describes the simplest way to install CCS5 on a single Linux PC (client and server running on the same host), and a single user. If you are setting up a more complex client/server system, with multiple users, under a dedicated contract then you will need access to our CCS Wiki; please request an account from your Terma contact point.

*Even if your ultimate goal is to setup and manage a full CCS, it may be worthwhile for you to try out these instructions first on a private machine; it may help you to understand CCS by starting with a simple installation first.*

The instructions were written for RHEL / Fedora Core / CentOS. They work similarly for other Linux distributions using RPM for software distribution (e.g. SLES, OpenSUSE LEAP), although you will need to take into account small differences between Linux distributions.

*NOTE: detailed instructions - especially screen shots - may go slightly out of date over time. If you find an important inconsistency that we missed, please let us know.*

## Which Linux Distribution?

CCS5 should work on practically any Linux platform with a current kernel and C/C++ runtime libraries. CCS5 policy is to avoid dependency on a specific Linux distribution and version. We also want to avoid a long list of dependent packages that you have to download, with different names on each distribution. For practical and economic reasons, we develop and test using CentOS and OpenSUSE LEAP. We only support other distributions where this falls under a specific contract.

The CCS installation avoids external dependencies by including all its dependent libraries (mainly Qt and Tcl) in the RPM installer. Nevertheless, CCS needs a database back-end for its archive index, and you have to install that separately.

## Platform

A typical starting point is a desktop PC, a virtual machine and an ISO image of the preferred Linux distribution, e.g. you might download a "Linux Live" image, and install a virtualisation tool on your PC, e.g. VMware Workstation or Virtual Box. The live distribution images can be saved to your desktop or to a bootable USB stick. Once booted, these distributions usually give you the option to install an image to your virtual hard drive.

## Document Conventions

We assume you are using `bash`, and shell commands are shown using a fixed font. For many of these instructions, you will need super-user privileges, or be able to access them via `sudo`.

2

“Normal” user commands are indicated with a “\$” prompt e.g.:

```
$ /opt/CCS/bin/ccs.sh
```

“Super” user (as root) commands are indicated with a “#” prompt, e.g.:

```
# systemctl restart SESS
```

Instead of working as root, you can use “`sudo`”, but we don’t repeat this with every instruction e.g.:

```
$ sudo systemctl restart SESS
```

Installation commands here use **yum**: please adjust the commands to the install command on your distribution (e.g. `dnf`, `zypper`). Package names may differ slightly according to the distribution.

## Preparations

### Set Host Name

We recommend choosing a short (<16 characters) and simple (alphanumeric only) host name. This host name (the host name in the system kernel, as opposed to its network host name) should be a simple host name, not including any domain name.

To change the hostname use:

```
# hostnamectl set-hostname <NAME>
```

If you have a static IP address, we recommend putting the (simple) host name in `/etc/hosts`, with the IP address. The IP address should **not** be the loopback address (127.0.0.1). For e.g.

```
# OK (if IP address is static)
```

```
10.1.20.67 myccs
```

```
# NOK!
```

```
#10.1.20.67      myccs.enormous.company.com
```

```
# NOK!
```

```
#127.0.0.1 myccs
```

Obviously, if you have configured your host name and `/etc/hosts` we recommend rebooting and checking that your network is generally working.

## Disable Firewall

For quick installation, we assume that Linux firewall is off. The CCS is usually installed in a secure environment where no direct external connections are possible. The list of ports used by CCS5 is published in the Wiki.

```
# systemctl disable firewalld
# systemctl stop firewalld
```

## Clear Settings... (Optional)

If you previously had an older CCS installation, or if you are switching between projects with different settings, then it is probably best to clear your old CCS settings.

Click on the SET application in your applications menu, or run the script

```
$ /opt/CCS/bin/settings.sh
```

- Hit Clear Settings button
- Check all checkboxes, including “Clear global settings” if it is enabled
- Hit Yes to confirm
- Quit from the CCS settings editor

*NOTE: this will preserve your license.*

*NOTE: before clearing settings, make sure they are written down in your project's InitSettings.tcl, so that they can be restored afterwards.*

## Install Archive RDBMS

MySQL is the most commonly-used database back-end; these short instructions assume this is what you will use. We tested our Linux installations using MariaDB, a free MySQL implementation.

*NOTE: CCS5 supports PostgreSQL equally well. If you prefer this database, the setup instructions are equivalent.*

```
# yum install mariadb-server
```

The server needs to be running:

```
# systemctl start mysqld
```

Or, depending on the platform:

```
# systemctl start mariadb
```

## Configure to use Consistent Case

We use lower-case table names in the database back-end; edit the file:

```
/etc/my.cnf.d/server.cnf
```

Find the server section and edit as follows:

```
[server]
```

```
lower_case_table_names=1
```

And then restart the database

```
# systemctl restart mysqld
```

## Configure CCS User in RDBMS

A completely new database installation may lack a root user/password. This can be initialised as follows

```
# mysql_secure_installation
```

Set up a CCS user in the database. It will be used by CCS5 to create archive index databases and tables within them for each CCS session. It is also used when searching for archived commands and telemetry.

The choice of user name needs to match the one used when the CCS is configured (see below), e.g.:

User: **ccsuser**

Password: **leiden1**

If you don't have a database administration tool this can also be done using command line:

```
# mysql -h localhost -u root -p mysql
MariaDB [mysql]> CREATE USER 'ccsuser'@'%' IDENTIFIED BY 'leiden1';
MariaDB [mysql]> GRANT ALL PRIVILEGES ON *.* TO 'ccsuser'@'%';
MariaDB [mysql]> exit
```

*NOTE: Different versions seem to allow either % or \* as a wildcard. You may see an error message about this, adjust the query to use %.% if required.*

## Install LSB

You may need to install the “Linux Standard Base” (LSB) for your operating system. This should ensure that desktop, installation, start-up & configuration scripts function correctly on your system. For example (adapt as required for your system)

```
# yum install redhat-lsb-core
```

## Install Tortoise SVN (Optional)

We recommend keeping your user files (i.e. TESTENV) under configuration control. This can be any tool of your choice (e.g. git, Mercurial etc). The most popular choice in CCS5 projects is SVN, and therefore CCS5 will automatically use some SVN features if the user files came from an SVN repository.

```
# yum install subversion
```

## Install CCS5

Download a CCS5 installer package from

<https://tgss.terma.com>

Navigate the site, following the links offering Linux downloads (from the Licenses tab)

**Daily** builds are the latest from the previous night. They have not been formally tested.

**Tested** builds are formally tested snapshots.

Download RPM and install it as follows:

```
# rpm -i ~/Downloads/CCS-1.0-XXX.x86_64.rpm
```

If you are upgrading from an older CCS version, you can use the -U option instead of -i.

If you had the SESS service running, it will be stopped automatically before upgrading.

## Create a Test Environment

CCS5 expects a TESTENV folder to be in your \$HOME folder, with a MIB database, TOPE scripts and Mimic pictures. If it doesn't exist, an empty one will be created, but it is more useful to have some pre-prepared contents. You have these options:

- Copy existing project files into place
- Check out from configuration control

### Option 1: Copy Test Environment files into place

This option may be possible if you already have some example files from an existing project. The installation directory contains "TestPacks" directory with several example TESTENV for self-test and training.

In this option simply copy and rename the directory to TESTENV in your home directory. The files should follow the standard structure under TESTENV, for example TSEQ, TSEQLIB, SYN, data/ASCII. By convention, the TSEQ folder should contain InitSettings.tcl that will initialise CCS5 settings for your project (e.g. spacecraft epoch).

### Option 2: Check out from configuration control

Alternatively, you can check out from your own SVN/git/etc repository, or if appropriate the TESTENV may be one from Terma. The URL of the Terma SVN server is here:

<https://projects.nl.terma.com/SVN/SYNOPTER/trunk/TestPacks>

A read-only SVN account can be used (you will not be able to check in any changes):

- **User:** TestPacks
- **Password:** TestPacks

You can check out the CcsSelfTest pack as follows:

```
$ cd ~
```

```
$ svn co --username TestPacks --password TestPacks  
https://projects.nl.terma.com/SVN/SYNOPTER/trunk/TestPacks/CcsSelfTest
```

Then rename “CcsSelfTest” directory to “TESTENV” (moving or removing any existing directory)

## Configure CCS5

Select the desktop menu (e.g. in GNOME)...

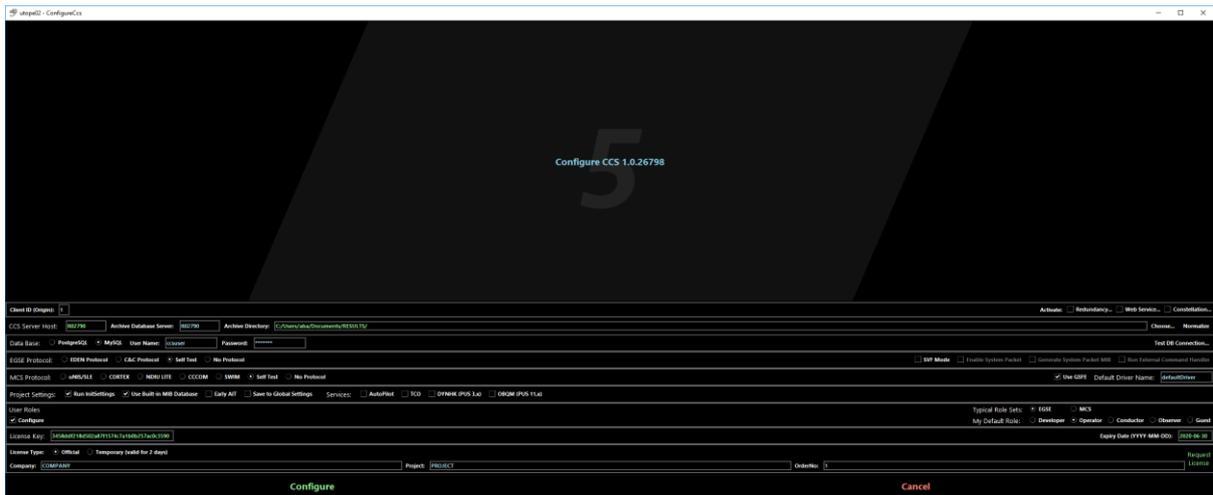


And hit Configure.

Alternatively, use the shell command:

```
$ /opt/CCS/bin/config.sh
```

A window like this should appear:



7

Please fill in the details as follows...

## Network and Archive Settings

**CCS Server Host:** enter the host name of your CCS server. The current hostname is set up by default.

*NOTE: do not enter 'localhost' and do not enter an IP address.*

**Archive Database Server:** enter the host name of your RDBMS if different from the CCS server host.

**Client ID:** This must be a number between 1 and 15. It is a unique identifier for the CCS client application (or workstation in a real deployment) on the CCS network. For a single host setup, the default value 1 is fine.

*NOTE: the CCS server console is client ID 0. In addition to the CCS server you can have up to 15 workstations. A client on the CCS server might be client ID 1, so workstations attached to a CCS might use client numbers 2 to 15.*

**Archive Directory:** this is the 'RESULTS' directory that will contain all the session data for the CCS installation. The default is **\$HOME/RESULTS**. When setting up a distributed CCS system, this should be a shared folder on the server that is accessible to all workstations.

## Data Base Settings

Enter the back-end RDBMS settings, e.g:

**Engine:** select **MySQL** or **PostgreSQL** as appropriate.

**User Name:** ccsuser

**Password:** leiden1

*NOTE: these must match the credentials that have been created in the database above. Click on "Test DB connection" to check that it works.*

## Protocols

Configure the protocols that will be used in your system.

If you are just trying out CCS, leave these as default.

Most projects will use either pure CCS or pure MCS protocols. However, it is permitted to set up a hybrid system in which both EGSE and MCS protocols are active.

**EGSE Protocol:** choose an EGSE (SCOE and Front End) protocol for checkout systems.

*HINT: In a “pure” MCS, this should be “No Protocol”*

**MCS Protocol:** choose a ground station protocol for operation systems.

*HINT: In a “pure” CCS, this should be “No Protocol”*

## Project Settings

**Run InitSettings:** if you have already put in place a TESTENV, this will automatically run the standard “InitSettings.tcl” script to further configure CCS5 for your specific project.

**Use Built-in MIB database:** this database contains some absolutely minimal MIB definitions needed for normal use of CCS5. It is compiled into the program, so you can’t edit it. Uncheck this option, if these definitions conflict with your project’s MIB.

**Early AIT:** when checked, this option disables excessive logging of warnings that are common at early AIT stages.

## License Key

If you don’t have a license key yet, you can acquire it from our website [ccs.terma.com](https://ccs.terma.com). Log in using your personal credentials and allocate a new license. You will need to specify a MAC address of your computer. If it has multiple network interfaces, we advise choosing the most stable and “real” NIC. The loopback address 00:00:00:00:00:00 will not be accepted. If your MAC address changes later, you will need to deallocate the old license key and allocate a new one.

If the license has been accepted, the license key and corresponding end date will show, and both fields will turn green.

*HINT: If you are just trying out CCS5 you can opt for a “Temporary license”.*

The remaining options of “Configure CCS” are described in detail in the CCS5 user manual. For now, you can leave them in default.

Click the ‘Configure’ button to save the configurations.

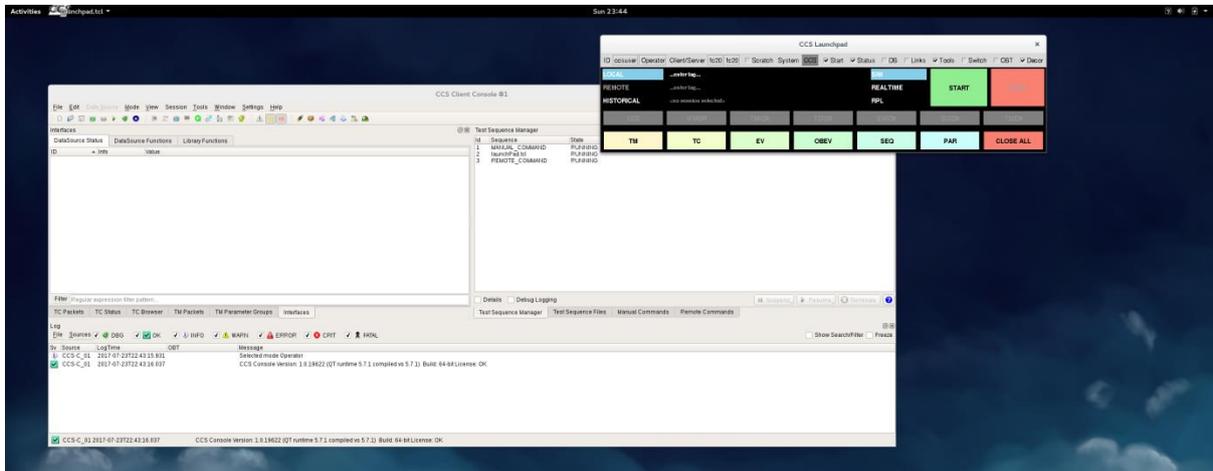
*NOTE: the configuration is saved only for the current user. If you login as another user, you will need to re-run “Configure CCS”.*

You can re-run this tool whenever needed but remember to restart all CCS applications (including SESS\_SERVER) afterwards, so that they read updated settings.

## Starting the CCS Client

If you now click the CCS icon on your desktop, the CCS Client Console will appear, and the CCS Launch Pad will also pop up:

*HINT: the Launch Pad appearance may be slightly different: you can check / uncheck boxes in the top panel show or hide different panels. We want to see the START button.*



Alternatively type the shell command

```
$ /opt/CCS/bin/ccs.sh
```

## Starting CCS

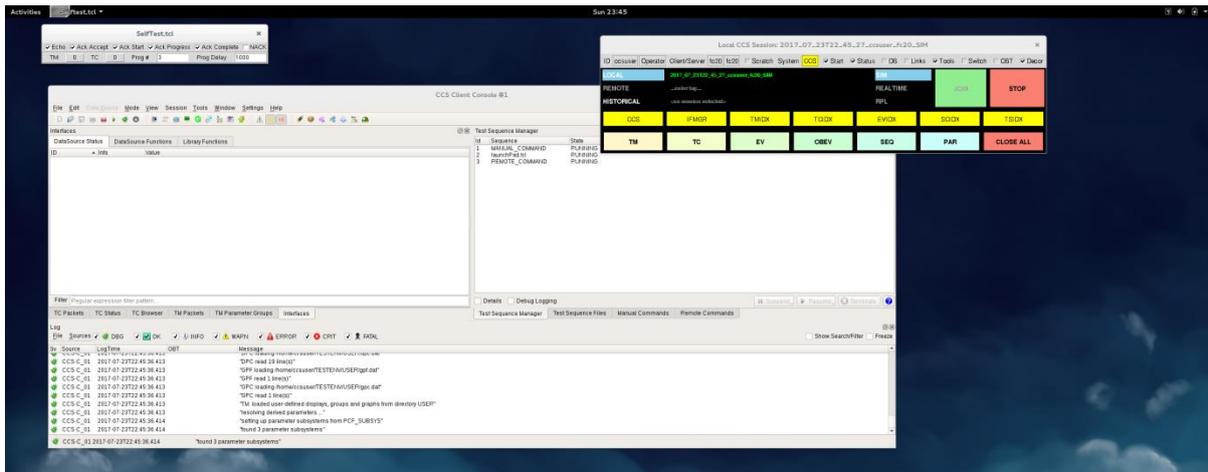
Now that you have configured the CCS and installed a test environment you should be able to start a useful CCS session. When the CCS client starts, the CCS Launch Pad opens automatically.

When you are ready, it looks like nothing is running, and it is waiting for you to start a local simulation session).

*HINT: if you do not see all of the panels shown above, click the check buttons to “reveal” them. Launch Pad “remembers” which panels have been hidden or exposed.*

Hit the big green START button.... A session name is automatically set and CCS starts.

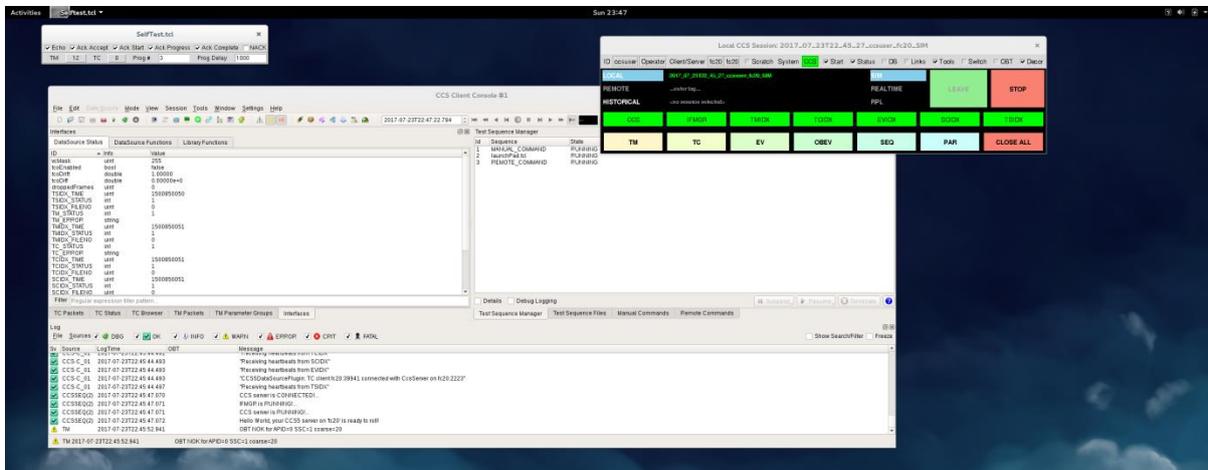
While the CCS is starting (a few seconds), the Launch Pad looks like this:



During session start-up:

- The SESS\_SERVER creates a new session directory in the RESULTS directory
- The client and server will try to connect to RDBMS
- The MIB database(s) will be loaded
- The client will listen for the heartbeat signals from the server
- The Launch Pad window should display all processes in green

When CCS is running, the display looks like this:



The different Launch Pad panels can be hidden and revealed to show more information or to save space.

Position the launch pad in a convenient place on your desktop, and it will stay there.

*HINT: if you accidentally quit Launch Pad and want to restart it, you can find it in the toolbar on the CCS client console (look for the green "rocket" icon).*

**Congratulations, CCS is now running!**

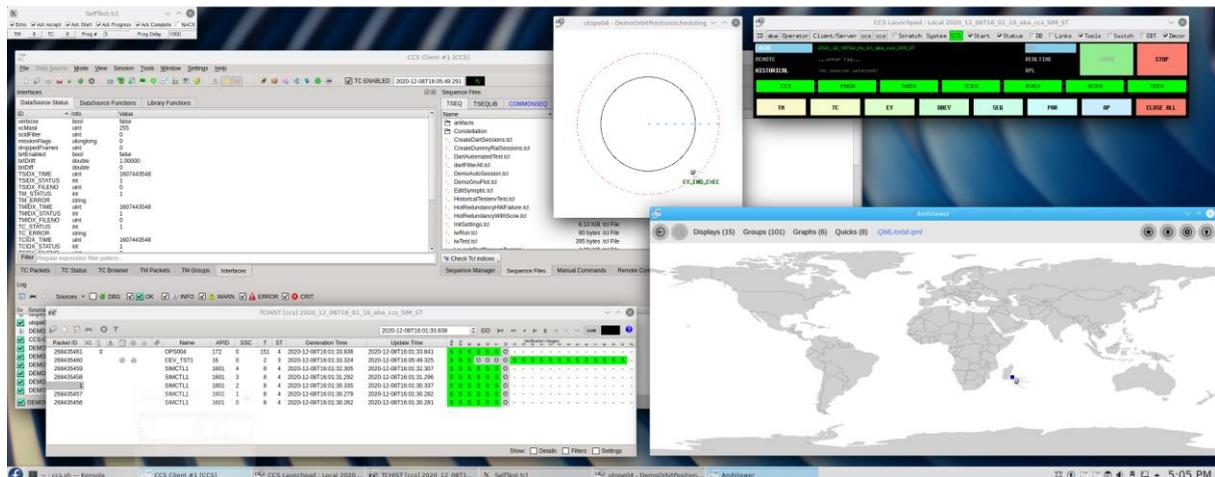
## Simple Demonstration

11

If you are using the CCS SelfTest environment, and want to see something interesting, try this

- Click Sequence Files (top-right panel of CCS Console) and switch to the COMMONSEQ tab
- Right-click on DemoOrbitPositionScheduling.tcl and hit Exec
- Go to LaunchPad and hit the TC button to open a TC history
- Click AND View in the desktop, and in the resulting display click the “Quicks” tab
- Open the “orbit” view

You should then see something like this:



In this demo, the CCS sends a cyclic TC (CEV\_TST1) as a parameter to another (OPS004). The CEV\_TST1 command is supposed to execute on each orbit cycle. The script calculates a simple simulated orbit, while at the same time simulating the verification of a TC as it reaches a different orbit position. It continues forever.

Try navigating back to the list of “Quicks” and opening “demo”: this should give you an idea of your platform graphical capabilities and performance.

## Stopping a Session

In the CCS Launch Pad, hit the STOP button and wait a few seconds.

*NOTE: if you simply quit the CCS Console, the CCS session will keep running. If you start the CCS Console again, this would be indicated by Launch Pad offering you to JOIN the current session.*

## Getting Help

Online help appears when you hit F1

## Troubleshooting

The most common problems are:

- The system firewall is blocking one or more of the ports used by CCS. Turn it off.
- You left a CCS (partially) running before de-installing and re-installing. Kill it.
- Mistake in the host name, or hostname incompatible with database engine back-end
- The network configuration changed since the CCS was configured (maybe VPN)
- Mistake in the configuration of RDBMS users or passwords
- Empty MIB database (nothing in your TESTENV)
- You may need to re-index your TOPE directories

You should be able to solve these problems by repeating the above procedure and looking closely for incorrect input.

In some Linux distributions, the dependency from the Tcl MySQL library to MySQL expects a slightly different version of the shared library. This can be fixed by creating a symbolic link. If you installed MariaDB instead of MySQL, you will find that the MySQL shared library is already a symbolic link, simply create a new symbolic link in the expected version to the MariaDB shared library.

To re-index all TOPE sequences, go to the Sequence Files pane, **hold down** the Check Tcl Indices button and select **Index All**.

Licensed customers have access to our CCS wiki at:

[http://dewiki.terma.com/CCS5/index.php/CCS5\\_and\\_TSC](http://dewiki.terma.com/CCS5/index.php/CCS5_and_TSC)

This gives all kinds of advice about system configuration and troubleshooting more obscure issues.

### Killing a Dangling or Zombie CCS Session

On Linux, this script kills all CCS processes:

```
# /opt/CCS/bin/abort.sh
```