



Download and Install TSC for Windows™

This procedure describes how to install TSC on Windows.

The instructions were written for Windows 10. They should still work for releases since Windows 7.

NOTE: detailed instructions - especially screen shots - may go slightly out of date over time.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.

NOTE: TSC is also delivered as an RPM installer for use on RHEL, OpenSUSE and related Linux distributions. We don't provide corresponding instructions for Linux, since the instructions are mostly equivalent.

Preparations

When upgrading from an earlier version of TSC...

Make sure you clear settings and uninstall TSC as follows.

Start TSC and go to menu Settings -> Clear. Hit Yes to confirm.

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.

Then navigate to Start -> TestConsole -> Uninstall and follow the wizard.

WARNING: this will delete the TSC installation folder altogether. In case you have patched any data in that folder (generally not recommended), back it up beforehand. Of course, your TESTENV will remain intact.

Check that the TSC installation folder is gone. If there are still some files left, make sure that no application is locking them, and delete them manually.

Install Notepad++

The recommended editor for TOPE sequences on Windows is Notepad++. It supports Tcl syntax highlighting out of the box, and TSC will detect Notepad++ automatically when you configure it.

<https://notepad-plus-plus.org/>

On Linux, it depends on your desktop (gedit, Kate, etc...)

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 is Tortoise SVN.

<https://tortoisesvn.net/>

When running the installer, make sure you enable the “command line tools” option, so that SVN is available from the Windows command shell.

Install TSC

Download the TSC installer from

<https://tgss.terma.com>

Downloads are available from the Licenses tab. We recommend using an officially tested release unless you really need a nightly build. Execute the installer. You may use the default options throughout.

Create a Test Environment

TSC expects a TESTENV directory to be in your “Documents” folder (or on Linux in your home directory), 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 SVN

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 folder to TESTENV in your Documents folder. 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 settings for your project (e.g. spacecraft epoch)

Option 2: Check out from SVN

Alternatively, you can check out from your own SVN 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

Check out the selected test pack into your “Documents” folder. The checkout should be into a single directory with subdirectories, under which you should see TSEQ, TSEQLIB, data/ASCII, QML, SYN, etc.

Re-index sequence directories

Indexing the sequence directories makes sure that TOPE (TCL) can auto-load procedure files when they are called.

In the Test Sequence Files pane **hold down** the Check Tcl Indices button and select **Index All**.

*NOTE: it is important to **hold down** the menu! Then you will see the **Index All** button!*

Run InitSettings.tcl

If TESTENV/TSEQ contains a file InitSettings.tcl, this probably needs to be executed.

- Look in the tab Sequence Files > TSEQ
- Locate the file InitSettings.tcl
- Right-click and select “Exec”

Before continuing....

After configuring TSC, you probably need to restart!

NOTE: restarting is necessary because some projects change configuration settings that are only read when the application starts. These settings will only take effect when TSC is restarted.

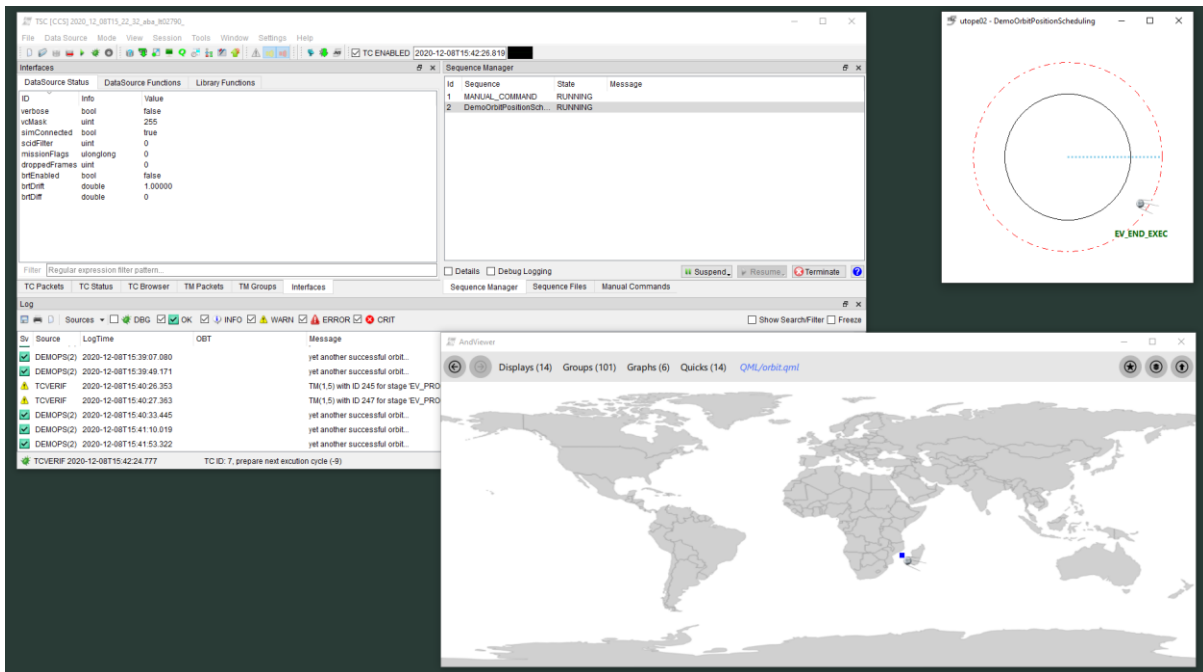
Simple Demonstration

If you are using the TSC 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
- Click View -> AND View in TSC, and in the resulting display click on the “Quicks” tab
- Open the “orbit” view

You should then see something like this:

4



In this demo, TSC 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.

Getting Help

Online help appears when you hit F1.

Licensed customers have access to our CCS5&TSC Wiki at:

http://dewiki.terma.com/CCS5/index.php/CCS5_and_TSC

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