# CST STUDIO SUITE® 2019

## GPU Computing Guide

DASSAULT SYSTEMES | The 3DEXPERIENCE® Company

# Contents

# 1 Nomenclature

The following section explains the nomenclature used in this document.

`command` Commands you have to enter either on a command prompt (cmd on MS Windows or your favorite shell on Linux) are typeset using typewriter fonts.

`<...>` Within commands the sections you should replace according to your environment are enclosed in "`<...>`". For example "`<CST_DIR>`" should be replaced by the directory where you have installed CST STUDIO SUITE (e.g. "`c:\Program Files\CST STUDIO SUITE`").

# 2 Supported Solvers and Features

- Transient Solver (T-solver/TLM-solver)
- Integral Equation Solver (direct solver and MLFMM only)
- Multilayer solver (M-solver)
- Particle-In-Cell (PIC-solver)
- Asymptotic Solver (A-solver; for this solver TCC mode is required on Windows!)
- Conjugate Heat Transfer Solver (CHT-solver)

Co-simulation with CST CABLE STUDIO is also supported.

## 2.1 Unsupported Features

The following features are currently not supported by GPU Computing. This list is subject to change in future releases or service packs of CST STUDIO SUITE.

| **Solver** | **Unsupported Features on GPU** |
|---|---|
| Transient Solver | • Subgridding |
| Particle In Cell Solver | • Modulation of External Fields<br>• Open Boundaries |

# 3 Operating System Support

CST STUDIO SUITE is continuously tested on different operating systems. For a list of supported operating systems please refer to

https://updates.cst.com/downloads/CST-OS-Support.pdf

In general, GPU computing can be used on any of the supported operating systems.

# 4 Supported Hardware

CST STUDIO SUITE currently supports up to 16 GPU devices in a single host system, meaning each number of GPU devices between 1 and 16 is supported.[1]

The following tables contain some basic information about the GPU hardware currently supported by the GPU Computing feature of CST STUDIO SUITE, as well as the requirements for the host system equipped with the hardware. To ensure compatibility of GPU hardware and host system please check

https://www.nvidia.com/object/tesla-qualified-servers.html

Please note that a 64 bit computer architecture is required for GPU Computing.
A general hardware recommendation can be found here:

https://www.cst.com/products/csts2/hardwarerecommendation

[1] It is strongly recommended to contact CST before purchasing a system with more than four GPU cards to ensure that the hardware is working properly and is configured correctly for CST STUDIO SUITE.

CST STUDIO SUITE officially supports the Nvidia Tesla and Quadro cards listed in the table below. That means that these GPUs are well tested and validated with CST software and you can contact CST support in case you run into any problems.

**List of supported GPU hardware for CST STUDIO SUITE 2019** [2] [3]

| Card Name | Series | Platform | Min. CST Version |
|---|---|---|---|
| Quadro RTX 8000 [4] | Turing | Workstations | 2019 SP6 |
| Quadro RTX 6000 [4] | Turing | Workstations | 2019 SP6 |
| Quadro RTX 5000 [4] | Turing | Workstations | 2019 SP6 |
| Quadro RTX 4000 [4] | Turing | Workstations | 2019 SP6 |
| Quadro GV100 | Volta | Workstations | 2018 SP6 |
| Tesla V100-SXM2-32GB (Chip) | Volta | Servers | 2018 SP6 |
| Tesla V100-PCIE-32GB | Volta | Servers | 2018 SP6 |
| Tesla V100-SXM2-16GB (Chip) | Volta | Servers | 2018 SP1 |
| Tesla V100-PCIE-16GB | Volta | Servers | 2018 SP1 |
| Tesla P100-SXM2 (Chip) | Pascal | Servers | 2017 release |
| Tesla P100-PCIE-16GB | Pascal | Servers | 2017 release |
| Tesla P100 16GB | Pascal | Servers | 2017 release |
| Tesla P100-PCIE-12GB | Pascal | Servers | 2017 SP2 |
| Quadro P6000 [4] | Pascal | Workstations | 2017 SP 2 |
| Quadro GP100 | Pascal | Workstations | 2017 SP2 |
| Tesla P40 [4] | Pascal | Servers | 2018 release |
| Tesla P4 [4] | Pascal | Servers | 2018 release |
| Tesla M60 [4] | Maxwell | Servers/Workst. | 2016 SP4 |
| Tesla M40 [4] | Maxwell | Servers | 2016 SP4 |
| Quadro M6000 24GB [4] | Maxwell | Workstations | 2016 SP4 |
| Quadro M6000 [4] | Maxwell | Workstations | 2015 SP4 |
| Tesla K80 | Kepler | Servers | 2014 SP6 |
| Tesla K40 m/c/s/st/d/t | Kepler | Servers/Workst. | 2013 SP5 |
| Quadro K6000 | Kepler | Workstations | 2013 SP4 |
| Tesla K20X | Kepler | Servers | 2013 release |
| Tesla K20m/K20c/K20s | Kepler | Servers/Workst. | 2013 release |
| Tesla K10 [4] | Kepler | Servers | 2013 release |

[2] Please note that cards of different series (e.g. ”Maxwell” and ”Pascal”) can’t be combined in a single host system for GPU Computing.

[3] Platform = Servers: These GPUs are only available with a passive cooling system which only provides sufficient cooling if it’s used in combination with additional fans. These fans are usually available for server chassis only!
Platform = Workstations: These GPUs provide active cooling, so they are suitable for workstation computer chassis as well.

[4] **Important:** The double precision performance of this GPU device is poor, thus, it is recommended for T-solver simulations only.

| **Hardware Type** | NVIDIA Tesla P100 Chip<br>(for Servers) | NVIDIA Tesla P100 PCIe[1]<br>(for Servers) |
|---|---|---|
| **Min. CST version required** | **2017 release** | **2017 release** |
| Number of GPUs | 1 | 1 |
| Max. Problem Size<br>(Transient Solver) | approx. 160 million mesh cells | approx. 160 / 120 million mesh cells |
| Form Factor | Chip<br>Passive Cooling | Dual-Slot PCI-Express<br>Passive Cooling |
| Memory | 16 GB CoWoS HBM2 | 16 / 12 GB CoWoS HBM2 |
| Bandwidth | 732 GB/s | 732 GB/s / 549 GB/s |
| Single Precision Performance [2] | 10.6 TFlops | 9.3 TFlops |
| Double Precision Performance [2] | 5.3 TFlops | 4.7 TFlops |
| Power Consumption | 300 W (max.) | 250 W (max.) |
| System interface | NVIDIA NVLink | 1x PCIe Gen 3 (x16 electrically) |
| Power Supply of Host System | min. 750 W | min. 750 W |
| Min. RAM of Host System [3] | 64 GB | 64 GB |

[1] **The 12 GB version has about 25 percent less performance compared to the 16 GB version.**

[2] Measured with BOOST enabled

[3] The host system requires approximately 4 times as much memory as is available on the GPU cards. Although it is technically possible to use less memory than this recommendation, the simulation performance of larger models will suffer.

**CST assumes no liability for any problems caused by this information.**

| **Hardware Type** | NVIDIA Quadro GP 100<br>(for Workstations) | NVIDIA Quadro P6000[1]<br>(for Workstations) |
|---|---|---|
| **Min. CST version required** | **2017 SP 2** | **2017 SP 2** |
| Number of GPUs | 1 | 1 |
| Max. Problem Size<br>(Transient Solver) | approx. 160 million mesh cells | approx. 240 million mesh cells |
| Form Factor | Dual-Slot PCI-Express | Dual-Slot PCI-Express |
| Memory | 16 GB HBM2 | 24 GB GDDR5X |
| Bandwidth | 720 GB/s | 432 GB/s |
| Single Precision Performance [2] | 10.3 TFlops | 12.0 TFlops |
| Double Precision Performance [2] | 5.2 TFlops | 0.2 TFlops |
| Power Consumption | 300 W (max.) | 300 W (max.) |
| System interface | 1x PCIe Gen 3 (x16 electrically) | 1x PCIe Gen 3 (x16 electrically) |
| Power Supply of Host System | min. 750 W | min. 750 W |
| Min. RAM of Host System [3] | 64 GB | 96 GB |

[1] **The double precision performance of this GPU device is poor, thus, it is recommended for T-solver simulations only.**

[2] Measured with BOOST enabled

[3] The host system requires approximately 4 times as much memory as is available on the GPU cards. Although it is technically possible to use less memory than this recommendation, the simulation performance of larger models will suffer.

**CST assumes no liability for any problems caused by this information.**

| **Hardware Type** | NVIDIA Tesla V100 SXM 16GB | NVIDIA Tesla V100 PCIe 16GB (for Servers) |
|---|---|---|
| **Min. CST version required** | **2018 SP 1** | **2018 SP 1** |
| Number of GPUs | 1 | 1 |
| Max. Problem Size (Transient Solver) | approx. 160 million mesh cells | approx. 160 million mesh cells |
| Form Factor | Chip<br>Passive Cooling | Dual-Slot PCI-Express<br>Passive Cooling |
| Memory | 16 GB CoWoS HBM2 | 16 GB CoWoS HBM2 |
| Bandwidth | 900 GB/s | 900 GB/s |
| Single Precision Performance [1] | 15 TFlops | 14 TFlops |
| Double Precision Performance [1] | 7.5 TFlops | 7 TFlops |
| Power Consumption | 300 W (max.) | 250 W (max.) |
| System interface | NVIDIA NVLink | 1x PCIe Gen 3 (x16 electrically) |
| Power Supply of Host System | min. 750 W | min. 750 W |
| Min. RAM of Host System [2] | 64 GB | 64 GB |

[1] Measured with BOOST enabled

[2] The host system requires approximately 4 times as much memory as is available on the GPU cards. Although it is technically possible to use less memory than this recommendation, the simulation performance of larger models will suffer.

**CST assumes no liability for any problems caused by this information.**

| **Hardware Type** | NVIDIA Tesla V100 SXM 32GB | NVIDIA Tesla V100 PCIe 32GB (for Servers) |
|---|---|---|
| **Min. CST version required** | **2018 SP 6** | **2018 SP 6** |
| Number of GPUs | 1 | 1 |
| Max. Problem Size (Transient Solver) | approx. 320 million mesh cells | approx. 320 million mesh cells |
| Form Factor | Chip<br>Passive Cooling | Dual-Slot PCI-Express<br>Passive Cooling |
| Memory | 32 GB CoWoS HBM2 | 32 GB CoWoS HBM2 |
| Bandwidth | 900 GB/s | 900 GB/s |
| Single Precision Performance [1] | 15 TFlops | 14 TFlops |
| Double Precision Performance [1] | 7.5 TFlops | 7 TFlops |
| Power Consumption | 300 W (max.) | 250 W (max.) |
| System interface | NVIDIA NVLink | 1x PCIe Gen 3 (x16 electrically) |
| Power Supply of Host System | min. 750 W | min. 750 W |
| Min. RAM of Host System [2] | 128 GB | 128 GB |

[1] Measured with BOOST enabled

[2] The host system requires approximately 4 times as much memory as is available on the GPU cards. Although it is technically possible to use less memory than this recommendation, the simulation performance of larger models will suffer.

**CST assumes no liability for any problems caused by this information.**

| **Hardware Type** | NVIDIA Quadro RTX 8000[1]<br>(for Workstations) | NVIDIA Quadro RTX 6000[1]<br>(for Workstations) |
|---|---|---|
| **Min. CST version required** | **2019 SP 6** | **2019 SP 6** |
| Number of GPUs | 1 | 1 |
| Max. Problem Size<br>(Transient Solver) | approx. 480 million mesh cells | approx. 240 million mesh cells |
| Form Factor | Dual-Slot PCI-Express | Dual Slot PCI-Express |
| Memory | 48 GB GDDR6 with ECC | 24 GB GDDR6 with ECC |
| Bandwidth | 672 GB/s | 672 GB/s |
| Single Precision Performance | 16.3 TFlops | 16.3 TFlops |
| Double Precision Performance | 0.5 TFlops | 0.5 TFlops |
| Power Consumption | 295 W (max.) | 295 W (max.) |
| PCI Express Requirements | 1x PCIe Gen 3 (x16 electrically) | 1x PCIe Gen 3 (x16 electrically) |
| Power Supply of Host System | min. 750 W | min. 750 W |
| Min. RAM of Host System [2] | 192 GB | 96 GB |

[1] **The double precision performance of this GPU device is poor, thus, it is recommended for T-solver simulations only.**

[2] The host system requires approximately 4 times as much memory as is available on the GPU cards. Although it is technically possible to use less memory than this recommendation, the simulation performance of larger models will suffer.

**CST assumes no liability for any problems caused by this information.**

| **Hardware Type** | NVIDIA Quadro RTX 5000[1]<br>(for Workstations) | NVIDIA Quadro RTX 4000[1]<br>(for Workstations) |
|---|---|---|
| **Min. CST version required** | **2019 SP 6** | **2019 SP 6** |
| Number of GPUs | 1 | 1 |
| Max. Problem Size<br>(Transient Solver) | approx. 160 million mesh cells | approx. 80 million mesh cells |
| Form Factor | Dual-Slot PCI-Express | Single Slot PCI-Express |
| Memory | 16 GB GDDR6 | 8 GB GDDR6 |
| Bandwidth | 448 GB/s | 416 GB/s |
| Single Precision Performance | 11.2 TFlops | 7.1 TFlops |
| Double Precision Performance | 0.35 TFlops | 0.2 TFlops |
| Power Consumption | 265 W (max.) | 160 W (max.) |
| PCI Express Requirements | 1x PCIe Gen 3 (x16 electrically) | 1x PCIe Gen 3 (x16 electrically) |
| Power Supply of Host System | min. 750 W | min. 750 W |
| Min. RAM of Host System [2] | 64 GB | 32 GB |

[1] **The double precision performance of this GPU device is poor, thus, it is recommended for T-solver simulations only.**

[2] The host system requires approximately 4 times as much memory as is available on the GPU cards. Although it is technically possible to use less memory than this recommendation, the simulation performance of larger models will suffer.

**CST assumes no liability for any problems caused by this information.**

| **Hardware Type** | NVIDIA Quadro GV100 |
|---|---|
| **Min. CST version required** | **2018 SP 6** |
| Number of GPUs | 1 |
| Max. Problem Size<br>(Transient Solver) | approx. 320 million mesh cells |
| Form Factor | Dual-Slot PCI-Express<br>Active Cooling |
| Memory | 32 GB CoWoS HBM2 |
| Bandwidth | 900 GB/s |
| Single Precision Performance [1] | 14 TFlops |
| Double Precision Performance [1] | 7 TFlops |
| Power Consumption | 250 W (max.) |
| System interface | PCIe Gen 3 (x16 electrically) |
| Power Supply of Host System | min. 750 W |
| Min. RAM of Host System [2] | 128 GB |

[1] Measured with BOOST enabled

[2] The host system requires approximately 4 times as much memory as is available on the GPU cards. Although it is technically possible to use less memory than this recommendation, the simulation performance of larger models will suffer.

# 5 Unsupported Hardware

If you have an Nvidia GPU card which is not supported (please see the list of supported cards in the previous section), but fulfills the requirements below, you may enable it for simulations by following these instructions. Note that using unsupported and untested hardware is not recommended. CST will not provide any support for any problems resulting from using unsupported GPU cards and CST will not run any tests on these GPUs.
The Nvidia GPUs that fulfill the following requirements may be enabled for simulations:

- capable of running CUDA 9.2 code
- at least 4 GB of free memory

To enable such GPUs, set the environment variable `CST_HWACC_ALLOW_UNVERIFIED_HARDWARE = 1`. Then all eligible installed and visible Nvidia GPUs will be considered for computation. You can limit the visible devices by using the environment variable `CUDA_VISIBLE_DEVICES`.

# 6 NVIDIA Drivers Download and Installation

An appropriate driver is required in order to use the GPU hardware. Please download the driver appropriate to your GPU hardware and operating system from the NVIDIA website. The driver versions listed below are verified for use with our software. Other driver versions provided by NVIDIA might also work but it is highly recommended to use the versions verified by CST.
We recommend the following driver versions for all supported GPU cards:

| | |
|---|---|
| **Windows:** | **Version 425.25** |
| **Linux:** | **Version 418.67** |

The recommended drivers will work for all supported Tesla and Quadro GPUs. For Quadro GPUs: Please check the Nvidia Driver website for a similar Tesla GPU (same generation, e.g. Pascal or Volta) to be able to find the recommended driver.

## 6.1 GPU Driver Installation

### 6.1.1 Installation on Windows

After you have downloaded the installer executable please start the installation procedure by double clicking on the installer executable. After a quick series of pop-up windows, the NVIDIA InstallShield Wizard will appear. Press the "Next" button and driver installation will begin (The screen may turn black momentarily.). You may receive a message indicating that the hardware has not passed Windows logo testing. In case you get this warning select "Continue Anyway".
If you are updating from a previously installed NVIDIA driver, it's recommended to select "clean installation" in the NVIDIA Installshield Wizard. This will remove the current

driver prior to installing the new driver.
The ”Wizard Complete” window will appear as soon as the installation has finished. Select ”Yes, I want to restart my computer now” and click the ”Finish” button.
**It is recommended that you run the HWAccDiagnostics tool after the installation to confirm that the driver has been successfully installed. Please use HWAccDiagnostics_AMD64.exe which can be found in the AMD64 directory of the installation folder.**

### 6.1.2 Installation on Linux

1. Login on the Linux machine as root.

2. Make sure that the adapter has been recognized by the system using the command
   `/sbin/lspci | grep -i nvidia`
   If you do not see any settings try to update the PCI hardware database of your system using the command
   `/sbin/update-pciids`

3. Stop the X-Server by running in a terminal the command (You may skip this step if you are working on a system without X-server)

   ```
   systemctl isolate multi-user.target
   ```
   (on systems using Systemd)

   ```
   init 3
   ```
   (on systems using SysVinit)

4. Install the NVIDIA graphics driver. Follow the instructions of the setup script. In most cases the installer needs to compile a specific kernel module. If this is the case the gcc compiler and Linux kernel headers need to be available on the machine.

5. Restart the X-server by running the command (You may skip this step if you are working on a system without X-server)

   ```
   systemctl isolate graphical.target
   ```
   (on systems using Systemd)

   ```
   init 5
   ```
   (on systems using SysVinit)

   **Note:** In case you're using the CST Distributed Computing system and a DC Solver Server is running on the machine where you just installed the driver you need to restart the DC Solver Server as otherwise the GPUs cannot be detected properly.

   **Note:** The OpenGL libraries should not be installed on a system which has no rendering capabilities (like a pure DC Solver Server or a pure cluster node). This can be accomplished by starting the NVIDIA installer using the option `"--no-opengl-files"`.

6. You may skip this step if a X-server is installed on your system **and** you are using a NVIDIA graphics adapter (in addition to the GPU Computing devices) in your system. If no X-server is installed on your machine or you don't have an additional NVIDIA graphics adapter, the NVIDIA kernel module will not be loaded automatically. Additionally, the device files for the GPUs will not be generated automatically. The following commands will perform the necessary steps to use the hardware for GPU Computing. It is recommended to append this code to your `rc.local` file such that it is executed automatically during system start.

```
# Load nvidia kernel module
modprobe nvidia

if [ "$?" -eq 0 ]; then

  # Count the number of NVIDIA controllers found.
  N3D=$(/sbin/lspci | grep -i nvidia | grep "3D controller" | wc -l)
  NVGA=$(/sbin/lspci | grep -i nvidia | grep "VGA compatible controller" | wc -l)

  N=$(expr $N3D + $NVGA - 1)
  for i in $(seq 0 $N); do
    mknod -m 666 /dev/nvidia$i c 195 $i;
  done

  mknod -m 666 /dev/nvidiactl c 195 255

fi
```

**Please note:**

- If you encounter problems during restart of the X-server please check chapter 8 ”Common Problems” in the file README.txt located at `/usr/share/doc/NVIDIA_GLX-1.0`. Please also consider removing existing sound cards or deactivating onboard sound in the BIOS. Furthermore, make sure you are running the latest BIOS version.

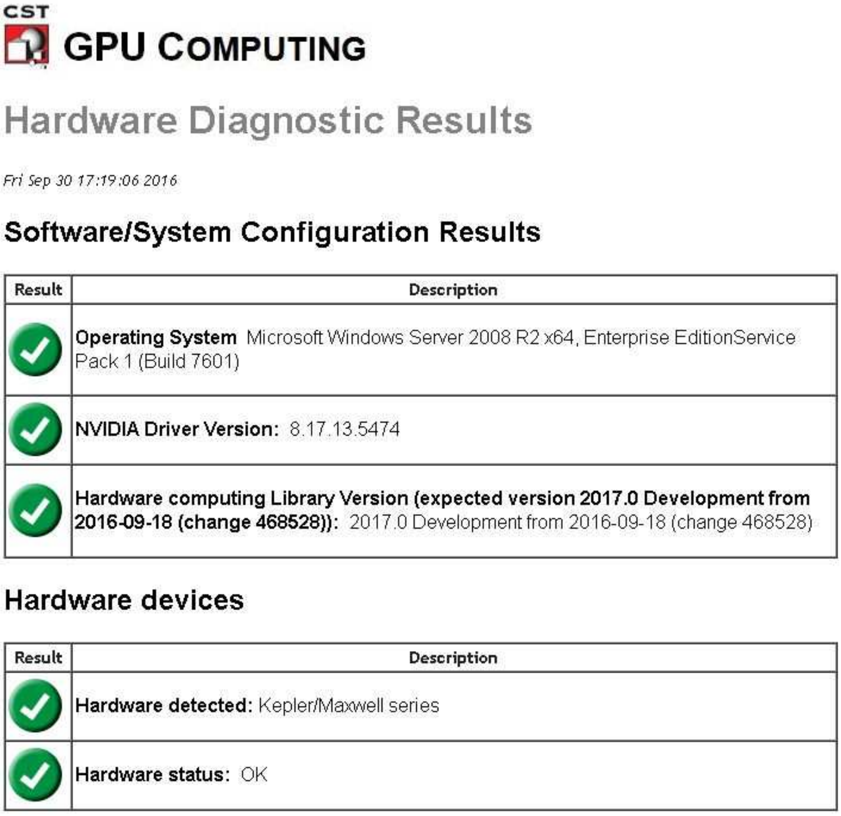
- After installation, if the X system reports an error like `no screen found`, please check Xorg log files in `/var/log`. Open the log files in an editor and search for `"PCI"`. According to the number of hardware cards in your system you will find entries of the following form: `PCI: (0@7:0:0)`. In `/etc/X11`, open the file `xorg.conf` in an editor and search for `"nvidia"`. After the line `BoardName "Quadro M6000"` (or whatever card you are using) insert a new line that reads `BusID "PCI:7:0:0"` according to the entries found in the log files before. Save and close the `xorg.conf` file and type `startx`. If X still refuses to start, try the other entries found in the Xorg log files.

- You need the installation script to uninstall the driver. Thus, if you want to be able to uninstall the NVIDIA software you need to keep the installer script.

- Be aware of the fact that you need to reinstall the NVIDIA drivers if your kernel is updated as the installer needs to compile a new kernel module in this case.

## 6.2 Verifying Correct Installation of GPU Hardware and Drivers

As a final test to verify that the GPU hardware has been correctly installed, the following test can be executed: Log in to the machine and execute the HWAccDiagnostics_AMD64 program found in the AMD64 subfolder of your CST installation (Windows) or in the folder LinuxAMD64 on a Linux system. The macro ”Check GPU Computing Setup” in the Solver macros performs exactly this check.The output of the tool should look similar to the following picture if the installation was successful.



CST GPU COMPUTING

Hardware Diagnostic Results

*Fri Sep 30 17:19:06 2016*

**Software/System Configuration Results**

| Result | Description |
|---|---|
| ✓ | **Operating System** Microsoft Windows Server 2008 R2 x64, Enterprise EditionService Pack 1 (Build 7601) |
| ✓ | **NVIDIA Driver Version:** 8.17.13.5474 |
| ✓ | **Hardware computing Library Version (expected version 2017.0 Development from 2016-09-18 (change 468528)):** 2017.0 Development from 2016-09-18 (change 468528) |

**Hardware devices**

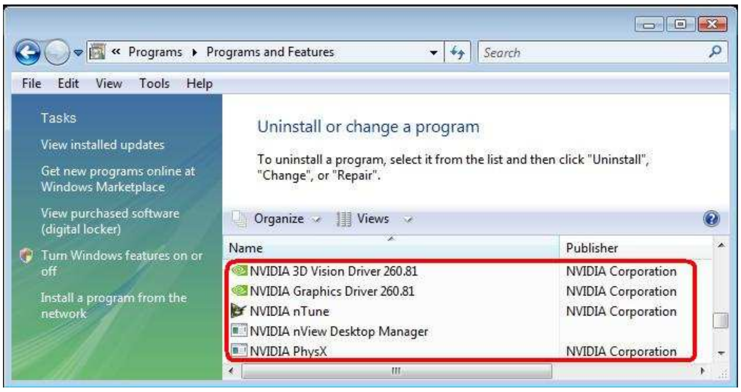
| Result | Description |
|---|---|
| ✓ | **Hardware detected:** Kepler/Maxwell series |
| ✓ | **Hardware status:** OK |

**Figure 1:** Output of HWAccDiagnostics_AMD64.exe tool.

## 6.3 Uninstalling NVIDIA Drivers

### 6.3.1 Uninstall Procedure on MS Windows

To uninstall NVIDIA drivers, select "NVIDIA Drivers" from the "Add or Remove Programs" list and press the "Change/Remove" button (see fig. 2). After the uninstall process has finished you will be prompted to reboot.



« Programs ▸ Programs and Features
Search
File Edit View Tools Help
Tasks
View installed updates
Get new programs online at Windows Marketplace
View purchased software (digital locker)
Turn Windows features on or off
Install a program from the network
Organize
Views

Uninstall or change a program

To uninstall a program, select it from the list and then click "Uninstall", "Change", or "Repair".

| Name | Publisher |
|---|---|
| NVIDIA 3D Vision Driver 260.81 | NVIDIA Corporation |
| NVIDIA Graphics Driver 260.81 | NVIDIA Corporation |
| NVIDIA nTune | NVIDIA Corporation |
| NVIDIA nView Desktop Manager | |
| NVIDIA PhysX | NVIDIA Corporation |

**Figure 2:** "Add or Remove Programs" dialog on Windows

### 6.3.2 Uninstall Procedure on Linux

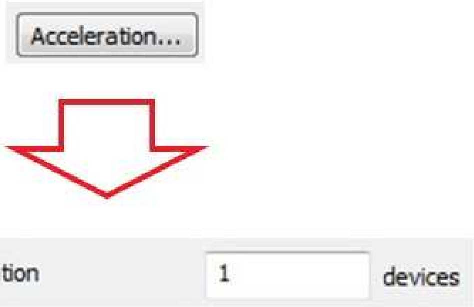
Start the installer with the "`--uninstall`" option. This requires root permissions.

# 7 Switch On GPU Computing

## 7.1 Interactive Simulations

GPU Computing needs to be enabled via the acceleration dialog box before running a simulation. To turn on GPU Computing:

1. Open the dialog of the solver.
2. Click on the ”Acceleration” button.
3. Switch on ”Hardware acceleration” and specify how many GPU devices should be used for this simulation. The specification of the number of devices is per solver (e.g. if DC is used). Please note that the maximum number of GPU devices available for a simulation depends upon the number of tokens in your license.



☑ Hardware acceleration 1 devices

## 7.2 Simulations in Batch Mode

If you start your simulations in batch mode (e.g. via an external job queuing system) there is a command line switch (`-withgpu`) which can be used to switch on the GPU Computing feature. The command line switch can be used as follows:[5]

In Windows:

```
"<CST_INSTALL_DIR>/CST Design Environment.exe" -m -r -withgpu=<NUMBER_OF_GPUs> "<FULL_PATH_TO_CST_FILE>"
```

In Linux:

```
"<CST_INSTALL_DIR>/cst_design_environment" -m -r -withgpu=<NUMBER_OF_GPUs> "<FULL_PATH_TO_CST_FILE>"
```

[5] This example shows the batch mode simulation for the transient solver (`-m -r`). To learn more about the command line switches used by CST STUDIO SUITE please refer to the online help documentation in the section ‘General Features’, subsection ‘Command Line Options’.

# 8 Usage Guidelines

## 8.1 The Error Correction Code (ECC) Feature

ECC can detect and eventually correct problems caused by faulty GPU memory. Such GPU memory errors typically cause unstable simulations. However, this feature deteriorates the performance of older GPU hardware (all cards of the Fermi, Kepler, and Maxwell series are affected). Therefore, we recommend disabling the feature. If simulations running on GPU hardware become unstable it is recommended to enable ECC temporarily as a diagnostic tool to determine whether the problems are caused by a GPU memory defect. Please also refer to section 11.

The latest NVIDIA GPU hardware (Pascal) has native ECC support with no performance overhead. For those GPUs ECC can't be switched off.

The ECC feature can be managed by using either the Nvidia Control Panel or the command line tool `nvidia-smi`. Please note, that on Windows 7, Windows Server 2008 R2, and newer version of Windows the following commands have to be run as administrator.

### 8.1.1 Managing the ECC Feature via Command Line

This procedure works on all supported versions of Windows and on all supported Linux distributions.

The command requires administrator privileges on Windows and root privileges on Linux, respectively.

1. Locate the file nvidia-smi. This file is typically found in "`c:\Program Files\NVIDIA Corporation\NVSMI`" or in `/usr/bin` on Linux.

2. Open up a command prompt/terminal window and navigate to this folder.

3. Execute the following command:
   ```
   nvidia-smi -L
   ```

4. Please note down how many GPUs are found.

5. To disable ECC: Please execute the following command for each of the GPUs:
   ```
   nvidia-smi -i <number_of_the_GPU_card> -e 0
   ```

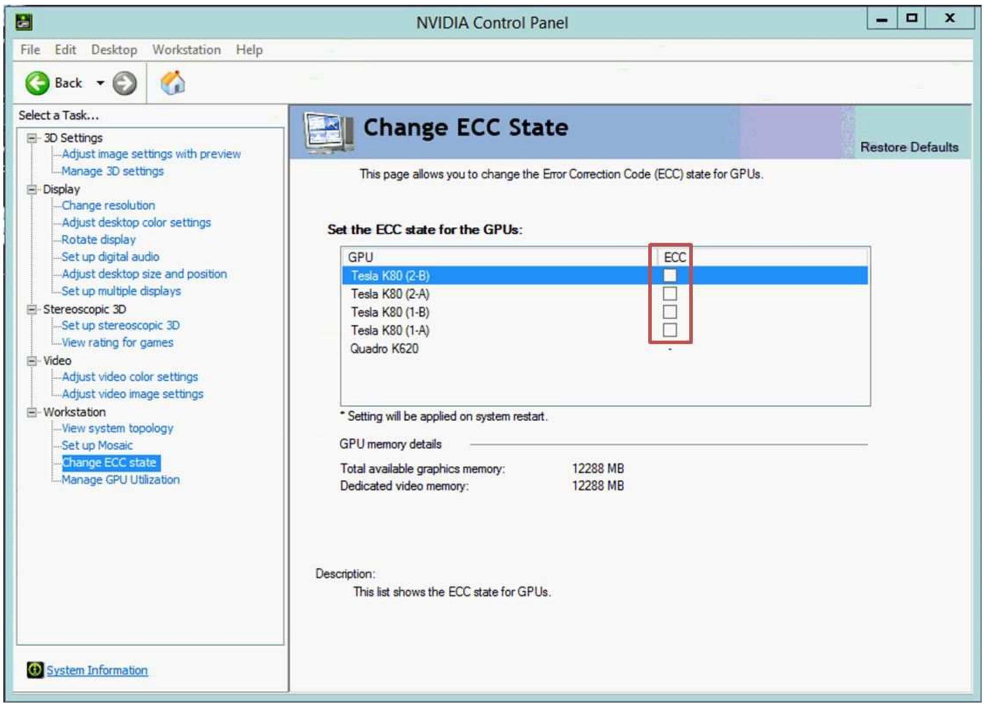
6. To enable ECC: Please execute the following command for each of the GPUs:
   ```
   nvidia-smi -i <number_of_the_GPU_card> -e 1
   ```

7. Reboot.

### 8.1.2 Managing the ECC Feature via NVIDIA Control Panel

This procedure works on all versions of Windows.

1. Start the Control Panel via the Windows start menu.
2. Start the NVIDIA Control Panel.
3. Search for the term ”ECC State” in the navigation tree of the dialog and open the ”ECC State” page of the dialog by clicking on the tree item.
4. Disable or enable the ECC feature for all Tesla devices (see fig. 3).



This page allows you to change the Error Correction Code (ECC) state for GPUs.

Set the ECC state for the GPUs:

| GPU | ECC |
|---|---|
| Tesla K80 (2-B) | □ |
| Tesla K80 (2-A) | □ |
| Tesla K80 (1-B) | □ |
| Tesla K80 (1-A) | □ |
| Quadro K620 | - |

* Setting will be applied on system restart.

GPU memory details

| | |
|---|---|
| Total available graphics memory: | 12288 MB |
| Dedicated video memory: | 12288 MB |

Description:

This list shows the ECC state for GPUs.

**Figure 3:** Switch off the ECC feature for all Tesla cards.

## 8.2 Tesla Compute Cluster (TCC) Mode (Windows only)

When available, the GPUs have to operate in TCC mode. **Please note that the TCC mode is currently required for A-solver GPU computing.**

### 8.2.1 Enable the TCC Mode

When available, the GPUs have to operate in TCC mode[6]. Please enable the mode, if not yet enabled.
Please note that the following commands require administrator privileges.

1. Locate the file nvidia-smi. This file is typically found in ”`c:\Program Files\NVIDIA Corporation\NVSMI`”.
2. Open up a command prompt and navigate to this folder.
3. Execute the following command:
   ```
   nvidia-smi -L
   ```
4. Please note down how many GPUs are found.
5. For each of the GPUs, please execute the following command:
   ```
   nvidia-smi -i <number_of_the_GPU_card> -dm 1
   ```
6. Reboot.

### 8.2.2 Disabling the TCC Mode

If available, this feature should always be enabled. However, under certain circumstances you may need to disable this mode.
Please note that the following commands require administrator privileges.

1. Locate the file nvidia-smi. This file is typically found in ”`c:\Program Files\NVIDIA Corporation\NVSMI`”.
2. Open up a command prompt and navigate to this folder.
3. Execute the following command:
   ```
   nvidia-smi -L
   ```
4. Please note down how many GPUs are found.
5. For each of the GPUs, please execute the following command:
   ```
   nvidia-smi -i <number_of_the_GPU_card> -dm 0
   ```
6. Reboot.

[6] The TCC Mode is available on all Tesla and on most Quadro cards. This mode is not available for Quadro cards which are connected to a display/monitor.

## 8.3 Disable the Exclusive Mode

This mode has to be disabled in order to use CST STUDIO SUITE.
To test if this mode is switched on, please do the following:

1. Locate the file nvidia-smi. This file is typically found in ”`c:\Program Files\NVIDIA Corporation\NVSMI`” or in `/usr/bin` on Linux.
2. Open up a command prompt and navigate to this folder.
3. Execute the following command:
   `nvidia-smi -q`

Search for the term ”Compute Mode” in the output of the tool. If the setting for ”Compute Mode” is not ”default” or ”0”, then the card is being operated in an exclusive mode. In this case, please execute the following commands in order to disable this mode.
Please note that the following commands require administrator privileges.

1. Locate the file nvidia-smi. This file is typically found in ”`c:\Program Files\NVIDIA Corporation\NVSMI`” or `/usr/bin` on Linux.
2. Open up a command prompt and navigate to this folder.
3. Execute the following command:
   `nvidia-smi -L`
4. Please note down how many GPUs are found.
5. For each of the GPUs, please execute the following command:
   `nvidia-smi -g <number_of_the_GPU_card> -c 0`
6. There is no need to reboot.

## 8.4 Display Link

Some cards of the Tesla series provide a display link to plug in a monitor. Using this display link has the following implications:

- The TCC Mode of the card cannot be used. This deteriorates the performance.
- GPU Computing can’t be used in a remote desktop session.

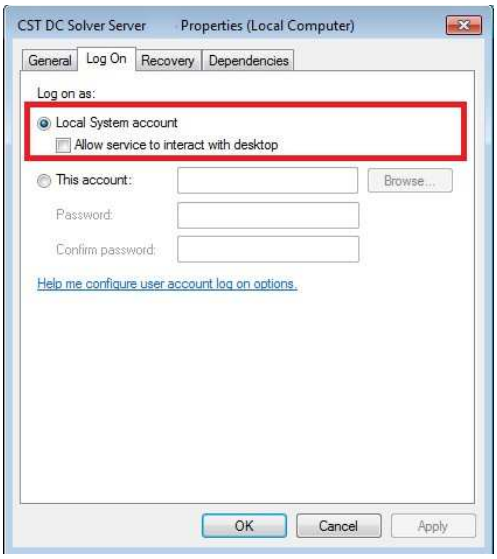
Because of these limitations we recommend using an additional graphics adapter for the graphics output, or if available, an onboard graphics chipset.

## 8.5 Combined MPI Computing and GPU Computing (Windows only)

For combined MPI Computing and GPU Computing the TCC mode of the GPU hardware must be enabled (see 8.2).

## 8.6 Service User (Windows only)

If you are using GPU Computing via the CST Distributed Computing system and your DC Solver Server runs on Windows then the DC Solver Server service must be started using the Local System account (see fig. 4). The CST STUDIO SUITE installer installs the service by default using the correct account.



Log on as:

● Local System account

□ Allow service to interact with desktop

○ This account:

Password:

Confirm password:

Help me configure user account log on options.

**Figure 4:** Local System Account.

## 8.7 GPU Computing using Windows Remote Desktop

For users with a LAN license, GPU Computing using RDP can be used in combination with Tesla or Quadro GPU cards as long as there is no monitor connected to the GPU.

## 8.8 Running Multiple Simulations at the Same Time

Running multiple simulations in parallel on **the same** GPU card will deteriorate the performance. Therefore we recommend to run just one simulation at a time. If you have a system with multiple GPU cards and would like to assign simulations to specific GPU cards please refer to section 8.13.

## 8.9 Video Card Drivers

Please use only the drivers recommended in this document or by the hardware diagnostics tool (See section 6.2). They have been tested for compatibility with CST products.

## 8.10 Operating Conditions

CST recommends that GPU Computing is operated in a well ventilated temperature controlled area. For more information, please contact your hardware vendor.

## 8.11 Latest CST Service Pack

Download and install the latest CST Service Pack prior to running a simulation or HWAccDiagnostics.

## 8.12 GPU Monitoring/Utilization

Locate the file nvidia-smi. This file is typically found in
"`c:\Program Files\NVIDIA Corporation\NVSMI`" on Windows or in `/usr/bin` on Linux. If you start this tool with the command line switch `-l` or `--loop` it will show the utilization and other interesting information such as the temperatures of the GPU cards. The `-l` option makes sure that the tool runs in a loop such that the information gets updated every couple seconds. For more options please run `nvidia-smi -h`. If you want to check the GPU utilization only, you can also run the graphical tool NvGpuUtilization (Windows only). This file is typically found in
"`c:\Program Files\NVIDIA Corporation\Control Panel Client`".

## 8.13 Select Subset of Available GPU Cards

If you have multiple GPU cards supported for GPU computing in the same machine you may want to specify the cards visible to the CST software such that your simulations are only started on a subset of the cards. This can be accomplished in two different ways.

### 8.13.1 Environment Variable CUDA_VISIBLE_DEVICES

The environment variable `CUDA_VISIBLE_DEVICES` which contains a comma separated list of GPU IDs will force a process (such as a CST solver) to use the specified subset of GPU cards only).[7] If this variable is set in the environment of the CST software or globally on your system the simulation will be started on the cards listed in the `CUDA_VISIBLE_DEVICES` list only.
Example: Open a shell (cmd on Windows, or bash on Linux) and enter

- `set CUDA_VISIBLE_DEVICES=0`

on Windows or

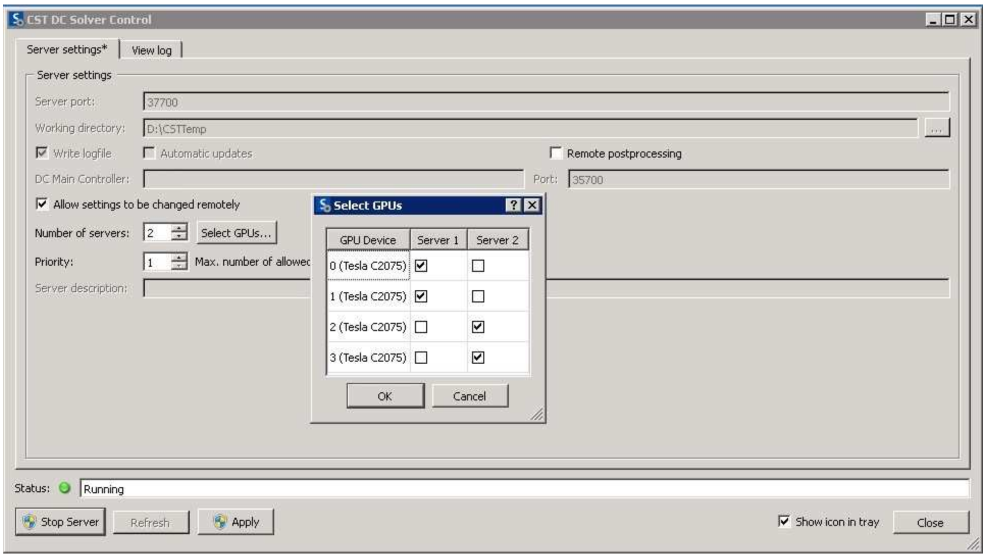
- `export CUDA_VISIBLE_DEVICES=0`

on Linux to bind all CST solver processes started from this shell to the GPU with ID 0. To make the setting persistent for all CST instances started on the system you may add the variable to the global system environment variables.

### 8.13.2 Distributed Computing

The CST Distributed Computing (DC) system can be used to assign the GPU cards of a multi-GPU system to different DC Solver Servers. The solver processes executed by a certain DC Solver Server will only be able to access the GPU cards assigned to this Solver Server (see fig. 5). Please refer to the online help documents of CST STUDIO SUITE (section ”Simulation Acceleration”, subsection ”Distributed Computing”) to learn more about the setup and configuration of the DC system.

[7] Execute the command `nvidia-smi -L` to get the GPU IDs of your cards.



CST DC Solver Control

Server settings* | View log

Server settings

Server port: 37700

Working directory: D:\CSTTemp

☑ Write logfile ☐ Automatic updates ☐ Remote postprocessing

DC Main Controller: Port: 35700

☑ Allow settings to be changed remotely

Number of servers: 2

Priority: 1 Max. number of allowec

Server description:

Select GPUs

| GPU Device | Server 1 | Server 2 |
|---|---|---|
| 0 (Tesla C2075) | ☑ | ☐ |
| 1 (Tesla C2075) | ☑ | ☐ |
| 2 (Tesla C2075) | ☐ | ☑ |
| 3 (Tesla C2075) | ☐ | ☑ |

Status: Running

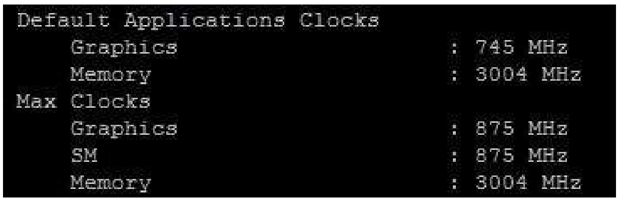
☑ Show icon in tray

**Figure 5:** Assignment of GPUs to specific DC Solver Servers.

# 9 NVIDIA GPU Boost

NVIDIA GPU Boost$^{\text{TM}}$is a feature available on the recent NVIDIA Tesla products.
This feature takes advantage of any power and thermal headroom in order to boost performance by increasing the GPU core and memory clock rates. The Tesla GPUs are designed with a specific Thermal Design Power (TDP). Frequently HPC workloads do not come close to reaching this power limit, and therefore have power headroom. A performance improvement can be expected when using the GPU Boost feature on the CST solvers. The Tesla GPUs come with a ”Base clock” and several ”Boost Clocks” which may be manually selected for compute intensive workloads with available power headroom. The Tesla GPUs give full control to end-users to select one of the core clock frequencies via the NVIDIA System Management Interface (nvidia-smi). For the K40 card, figuring out the right boost clock setting may require some experimentation to see what boost clock works best for a specific workload. NVIDIA GPU Boost on the Tesla K80 is enabled by default and dynamically selects the appropriate GPU clock based on the power headroom. The GPU Boost feature can be employed by using either the NVIDIA Control Panel or the command line tool nvidia-smi. The nvidia-smi file is typically found in
”`c:\Program Files\NVIDIA Corporation\NVSMI`” in Microsoft Windows or `/usr/bin` in Linux. The following are common commands for setting the GPU Boost feature and checking GPU performance. To display the current application clock in use execute the following command:

```
nvidia-smi -q -d CLOCK
```



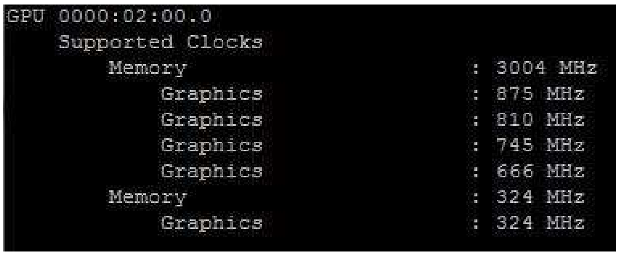
```
Default Applications Clocks
    Graphics                     : 745 MHz
    Memory                       : 3004 MHz
Max Clocks
    Graphics                     : 875 MHz
    SM                           : 875 MHz
    Memory                       : 3004 MHz
```

Before making any changes to the clocks, the GPU should be set to Persistence Mode. Persistence mode ensures that the driver stays loaded and does not revert back to the default clock once the application is complete and no CUDA or X applications are running on the GPU. To enable persistence mode use the following command:

```
nvidia-smi -pm 1
```

To view the clocks that are supported by the Tesla board:

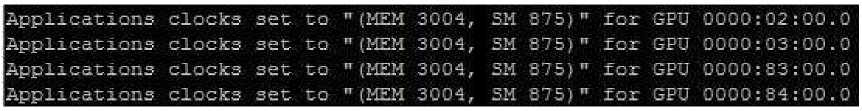
```
nvidia-smi -q -d SUPPORTED_CLOCKS
```



```
GPU 0000:02:00.0
    Supported Clocks
        Memory                       : 3004 MHz
            Graphics                 : 875 MHz
            Graphics                 : 810 MHz
            Graphics                 : 745 MHz
            Graphics                 : 666 MHz
        Memory                       : 324 MHz
            Graphics                 : 324 MHz
```

Please note that the supported graphics clock rates are tied to a specific memory clock rate so when setting application clocks you must set both the memory clock and the graphics clock[8]. Do this using the `-ac <MEM clock, Graphics clock>` command line option.

```
nvidia-smi -ac 3004,875
```



```
Applications clocks set to "(MEM 3004, SM 875)" for GPU 0000:02:00.0
Applications clocks set to "(MEM 3004, SM 875)" for GPU 0000:03:00.0
Applications clocks set to "(MEM 3004, SM 875)" for GPU 0000:83:00.0
Applications clocks set to "(MEM 3004, SM 875)" for GPU 0000:84:00.0
```

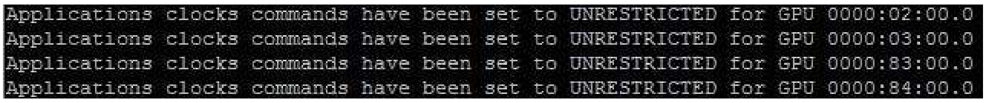
Execute the following command to reset the application clocks back to default settings.

```
nvidia-smi -rac
```

Changing application clocks requires administrative privileges. However, a system administrator can remove this requirement to allow non-admin users to change application clocks by setting the application clock permissions to ’UNRESTRICTED’ using the following command:

```
nvidia-smi -acp UNRESTRICTED
```

[8] The memory clock should remain at 3 GHz for the Tesla K40.



```
Applications clocks commands have been set to UNRESTRICTED for GPU 0000:02:00.0
Applications clocks commands have been set to UNRESTRICTED for GPU 0000:03:00.0
Applications clocks commands have been set to UNRESTRICTED for GPU 0000:83:00.0
Applications clocks commands have been set to UNRESTRICTED for GPU 0000:84:00.0
```

Please be aware that the application clock setting is a recommendation. If the GPU cannot safely run at the selected clocks, for example due to thermal or power reasons, it will automatically lower the clocks to a safe clock frequency. You can check whether this has occurred by typing the following command while the GPU is active:

```
nvidia-smi -a -d PERFORMANCE
```

# 10 Licensing

The GPU Computing feature is licensed by so called "Acceleration Tokens", i.e. your CST license must contain at least one "Acceleration Token" if you want to accelerate your simulations using a GPU. The CST Licensing Guide, which can be downloaded from the CST homepage, contains more information about the licensing of the GPU Computing feature. Please open the link https://www.cst.com/Company/Terms in your browser to find the recent version of this document.

# 11 Troubleshooting Tips

The following troubleshooting tips may help if you experience problems.

- If you experience problems during the installation of the NVIDIA driver on the Windows operating system please try to boot Windows in "safe mode" and retry the driver installation.

- If you have a multi-GPU setup (4 or 8 GPUs) and you encounter an "out-of-memory" problem please set the environment variable
  `CUDA_DEVICE_MAX_CONNECTIONS=1`.
  In case your host system has at least 512 GB of RAM please also check out the following website: GPU addressing capabilities.

- Please note that CST STUDIO SUITE cannot run on GPU devices when they are in "exclusive mode". Please refer to section 8.3 on how to disable this mode.

- If you are using an external GPU device ensure that the PCI connector cable is securely fastened to the host interface card.

- Uninstall video drivers for your existing graphics adapter prior to installing the new graphics adapter.

- Make sure the latest motherboard BIOS is installed on your machine. Please contact your hardware vendor support for more information about the correct BIOS version for your machine.

- Use the HWAccDiagnostics tool to find out whether your GPU hardware and your driver is correctly recognized.

- GPU temperatures are crucial for the performance and overheating of GPU devices can lead to hardware failures. Please refer to section 8.12 for details.

- A faulty GPU device can be responsible for seemingly random solver crashes. To ensure that your GPU is working properly please run tests provided by the HWAccDiagnostics tool found in the CST installation. Examples of usage:

  - `HWAccDiagnostics_AMD64 --runstresstest -duration=2000 -percentage=99` will run a memory test on all GPUs one by one (helps to identify GPU hardware problems usually related to a specific GPU).
  - `HWAccDiagnostics_AMD64 --runstresstest2 -duration=2000 -percentage=99` will run a simulation on all GPUs concurrently first, followed by running the same simulation on each GPU seperately (helps to identify thermal issues).
  - `HWAccDiagnostics_AMD64 --runstresstest2 -duration=2000 -percentage=99 --deviedID=0` will run the same simulation as before on device ID 0 only (helps to verify problems usually related to a specific GPU).
  - Please run `HWAccDiagnostics_AMD64 --h` to see all possible options.

- In case simulations are getting unstable when running on the GPU it's recommended to check the GPU memory by switching on the ECC feature on the GPU (see 8.1).

- If a GPU is not recognized during the installation please check if `Memory Mapped I/O above 4GB` is enabled in your bios settings.

- CUDA error 11: If you encounter an error message similar to
  `CUDA error 11:  invalid argument, dev:  1`
  you can usually fix this problem by using CUDA_VISIBLE_DEVICES so that only devices are visble for CUDA which will be used for simulations. Please refer to section 8.13 for details.

- TCC mode: GPUs running in WDDM instead of TCC mode might eventually fail during memory allocations. It is highly recommended to put all GPUs in TCC mode to avoid these kind of problems (see 8.2).

- Please execute the nvidia-smi tool found in
  "`c:\Program Files\NVIDIA Corporation\NVSMI`" on Windows and in "`/usr/bin`" on Linux in order to find out whether the GPUs are correctly recognized by the GPU driver.

If you have tried the points above with no success please contact CST technical support (info@cst.com).

## 12 History of Changes

The following changes have been applied to the document in the past.

| Date | Description |
|---|---|
| Aug. 06 2018 | first version of this document |
| Aug. 17 2018 | remove Fermi GPUs from list of supported hardware, add section about unsupported hardware, update drivers, add information about Quadro drivers |
| Aug. 21 2018 | fix first date since when Quadro P40 and Quadro P4 are supported |
| Jan. 15 2019 | mention requirement of TCC for A-solver |
| Jul. 8 2019 | add support for RTX GPUs, update recommeded driver |