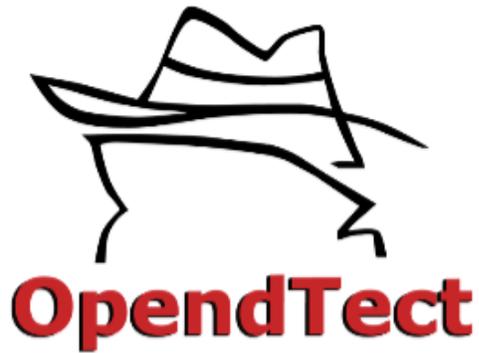


OpendTect Administrator's Manual - 6.6



Created by



Copyright © 2002-2021 by dGB Beheer B.V.

All rights reserved. No part of this publication may be reproduced and/or published by print, photo print, microfilm or any other means without the written consent of dGB Beheer B.V.

Under the terms and conditions of any of the following three license types, license holders are permitted to make hard copies for internal use:

- [GNU GPL](#)
- [OpendTect Pro License](#)
- [Academic License](#)

Table of Contents

OpendTect Administrator's Manual - 6.6	1
Table of Contents	2
System Requirements	6
Windows	6
Minimum	6
Recommended	6
For Machine Learning	7
Linux	8
Minimum	8
Recommended	8
For Machine Learning	9
Mac OS	11
Minimum	11
Recommended	11
For Machine Learning	11
Installation	13
OpendTect	13
Installing OpendTect on Windows	13
Installation via Installation Manager	13
Installer logging	19
Installing OpendTect on Linux	20

Installation via Installation Manager	20
Installer logging	25
Installing OpendTect on RHEL 6.x	26
Installing OpendTect on Mac OS X	28
Security setting	28
Installation via Installation Manager	30
Licenses	36
Licensing and Host IDs	36
Installing demo/node-locked licenses	38
Installing floating licenses	40
Setting the DGBLD_LICENSE_FILE environment variable	40
Using the FlexNet Impath tool	41
Clear license installation	42
Prepare offline installation packages	44
Install OpendTect via offline installation package	46
Setup Python Settings	48
Setup a Custom Python environment	48
Setup Distributed Computing	50
Tips & tricks	53
Setup the preferred GPU with NVIDIA Control Panel	55
Setup the preferred GPU with Windows settings	60
Setup NVIDIA GPU Performance Counters	64

Environment variables	65
Firewall Rules and Open Ports	67
Firewall rules For OpendTect 6.6 (Windows)	67
Open Ports	67
Python environments	69
Information about the OpendTect Python environments	69
Installing Python environments with the OpendTect installer	71
Installing Python environments from YAML files	72
Download the OpendTect Python YAML files	72
Install the environments	72
Fixes for callbacks.py and cupti	74
License server	76
Installing a license server on Linux	76
Starting a license server	76
Error messages	77
Client installation	78
Installing a license server on Windows	79
Starting a license server	82
Client installation	83
Petrel* Plugin	85
Installing the PIP file	85
Choosing the preferred Communication Port (TCP/IP)	85

PetrelDirect user documentation	86
Installation of PetrelDirect plugin in OpendTect and Petrel*	86
OpendTect	86
Petrel*	87
Troubleshooting	91
Log files	92
Settings files in .od folder	93
Debugging OpendTect	94
Appendix A - FlexNet Licenses Explained	95

System Requirements

Windows

Minimum

- **Version:** 8.1/10
- **CPU:** Intel/AMD, 64 bits
- **GPU:** Basic Intel graphics cards or Nvidia (e.g. recent GeForce/Quadro/NVS series) graphics cards; AMD graphics cards may work.
- **Memory:** 16 GB of RAM, OpendTect itself needs at least 2 GB RAM. Therefore, 16 GB will almost certainly be the absolute minimum.
- **Storage:** Hard Disk

Recommended

- **Version:** 8.1/10
- **CPU:** Intel/AMD processor with 64 bit support, 3+ GHz multi-core.
Note that OpendTect uses all processors if necessary. The more cores and speed, the better. OpendTect will automatically use multiple threads in many situations. This depends on the type of attribute, display, etc. We put a lot of effort to get time-consuming tasks multi-threaded.
- **GPU:** Nvidia (e.g. recent main-stream up to high-end GeForce series) graphics cards. Quadro or NVS series cards could give the bit extra you want. In doubt, buy the best GeForce card you can find. When buying a laptop make sure that it has a Nvidia chip-set.
- **Memory:** on the safe side don't go for less than 32 GB.
Buy as much memory that you can afford and fits in the system. The big clients for example use nothing less than 512 GB.
- **Storage:** SSD is best, other good options are Hard Disk and Network Drive.
This is usually under-valued, but it's often the crucial performance component. SSD

disks will give a tremendous boost in performance; essentially, data on SSD disks loads almost as fast as pre-loaded, in-memory data. Performance could be miserable if data needed to stream through (relatively) slow disks and/or networks.

For Machine Learning

- **Version:** 8.1/10
- **CPU:** Intel, 64 bits for when using Python environment *Intel™ Math Kernel - MKL* for Machine Learning using CPU only. AMD, 64 bits should be fine when using the Python Environment with *CUDA 10.1* for Machine Learning on the GPU. Ideally you want the system to be expendable to 4 GPUs. The CPU will need to support all GPUs. Important to look for is how many PCIe lanes the CPU supports and how many PCIe lanes are needed for the system's number of GPUs and M.2 NVMe SSDs. We recommend to get a CPU with at least 8 cores, 16 threads and 40 PCIe lanes.
- **GPU:** Nvidia, GeForce or Quadro series. The GPU needs to be fast enough and able to fit the model and data batch in memory. When in doubt choose the one with more memory. Other things to look for is the number of CUDA cores, tensor cores and GB memory bandwidth per second. We recommend the following cards:
 - Nvidia GeForce RTX 2080 Ti with 11 GB DDR6 memory
 - Nvidia Quadro RTX 6000 with 24 GB DDR6 memory
 - Nvidia Quadro RTX 8000 with 48 GB DDR6 memory
- **Memory:** DDR4 memory, don't go for less than 32 GB. Buy as much memory that you can afford and fits in the system.
- **Storage:** The best choice is M.2 NVMe SSD that is big enough for the data. The advantage of M.2 NVMe SSD is that it is plugged into the motherboard and is super fast. Other options are SATA SSD, Hard Disk and Network Drive. Performance could be miserable if data needed to stream through (relatively) slow disks and/or networks.

Please note that:

- *For best performance OpenGL drivers should be up-to-date. For Machine Learning on GPU we provide a Python package with CUDA 10.1. Please see this [table](#) on the Nvidia CUDA Toolkit documentation page for the minimum compatible driver version.*
- *Windows needs to be updated with the latest updates from Microsoft.*

Linux

Minimum

- **Modern Linux distro.**
 - We have tested:
 - RedHat/CentOS 7.2 and higher
 - Ubuntu 16.04 and higher
 - OpenSUSE Leap 15.1 and higher
 - Other distros will probably work, possibly with as small tweak
- **CPU:** Intel/AMD processor with 64 bit support
- **GPU:** Basic Intel Graphics cards or Nvidia (e.g. recent GeForce/Quadro/NVS series) graphics cards; AMD graphics cards may work.
- **Memory:** 16 GB of RAM, OpendTect itself needs at least 2 GB RAM. Therefore, 16 GB will almost certainly be the absolute minimum.
- **Storage:** Hard Disk

Recommended

- **Modern Linux distro.** We have tested:
 - RedHat/CentOS 7.2 and higher
 - Ubuntu 18.04 and higher
 - OpenSUSE Leap 15.2 and higher
- **CPU:** Intel/AMD processor with 64 bit support, 3+ GHz multi-core.
Note that OpendTect uses all processors if necessary. The more cores and speed, the better. OpendTect will automatically use multiple threads in many situations. This depends on the type of attribute, display, etc. We put a lot of effort to get time-consuming tasks multi-threaded.
- **GPU:** Nvidia (e.g. recent main-stream up to high-end GeForce series) graphics cards. Quadro or NVS series cards could give the bit extra you want. In doubt, buy the best GeForce card you can find. When buying a laptop make sure that it has a Nvidia chipset.
- **Memory:** on the safe side don't go for less than 32 GB.
Buy as much memory that you can afford and fits in the system. The big clients for

example use nothing less than 512 GB.

- **Storage:** SSD is best, other good options are Hard Disk and Network Drive. This is usually under-valued, but it's often the crucial performance component. SSD disks will give a tremendous boost in performance; essentially, data on SSD disks loads almost as fast as pre-loaded, in-memory data. Performance could be miserable if data needed to stream through (relatively) slow disks and/or networks.

For Machine Learning

- **Modern Linux distro.** We have tested:
 - RedHat/CentOS 7.2 and higher
 - Ubuntu 18.04 and higher
 - OpenSUSE Leap 15.2 and higher
- **CPU:** Intel, 64 bits for when using Python environment *Intel™ Math Kernel - MKL* for Machine Learning using CPU only. AMD, 64 bits should be fine when using the Python Environment with *CUDA 10.1* for Machine Learning on the GPU. Ideally you want the system to be expendable to 4 GPUs. The CPU will need to support all GPUs. Important to look for is how many PCIe lanes the CPU supports and how many PCIe lanes are needed for the system's number of GPUs and M.2 NVMe SSDs. We recommend to get a CPU with at least 8 cores, 16 threads and 40 PCIe lanes.
- **GPU:** Nvidia, GeForce or Quadro series. The GPU needs to be fast enough and able to fit the model and data batch in memory. When in doubt choose the one with more memory. Other things to look for is the number of CUDA cores, tensor cores and GB memory bandwidth per second. We recommend the following cards:
 - Nvidia GeForce RTX 2080 Ti with 11 GB DDR6 memory
 - Nvidia Quadro RTX 6000 with 24 GB DDR6 memory
 - Nvidia Quadro RTX 8000 with 48 GB DDR6 memory
- **Memory:** DDR4 memory, don't go for less than 32 GB. Buy as much memory that you can afford and fits in the system.
- **Storage:** The best choice is M.2 NVMe SSD that is big enough for the data. The advantage of M.2 NVMe SSD is that it is plugged into the motherboard and is super fast. Other options are SATA SSD, Hard Disk and Network Drive. Performance could be miserable if data needed to stream through (relatively) slow disks and/or networks.

Please note that:

- *OpendTect may work when using the [Nouveau](#) driver, however for best performance the Nvidia driver should be installed. The nouveau driver does not support CUDA.*
- *For best performance OpenGL drivers should be up-to-date. For Machine Learning on GPU we provide a Python package with CUDA 10.1. Please see this [table](#) on the Nvidia CUDA Toolkit documentation page for the minimum compatible driver version.*
- *Low-level GPUs keep showing poor performances through the generations. Shading functionality requires special GPU features, present in the main-stream and high-end GeForce, Quadro and NVS cards. Nevertheless, under Linux, only Nvidia provides drivers capable of running the shading feature. If you can't see any colors on graphic elements, try disabling shading (Utilities > Settings > Look and Feel).*
- *Linux distros should be LSB compliant. You can check this using the command `lsb_release`. This is particularly stringent for commercial plugins using the FlexNet system. There is documentation on installing license files for commercial plugins, and there is a page with background information.*
- *Linux 64 bits releases require the `libstdc++` library to be present on the system. In the table you can see the minimum `libstdc++` library version that is needed:*

OpendTect version	libstdc++ library needed
6.4	6.0.19 or newer
6.6	6.0.21 or newer

- *OpendTect is known to work under Red Hat, CentOS, Debian, Ubuntu, OpenSUSE and other distributions, as well as earlier versions of the main distributions, too. Fedora usage is not recommended - although it may work it's the only distro that regularly fails to work in combination with OpendTect. This is probably because the graphics vendors do not support it well in terms of drivers.*

Mac OS

Minimum

- **Version:** Mac OS X 10.10 and up to Catalina
- **CPU:** Mac/Intel processor with 64-bit support
- **GPU:** Basic Intel or AMD graphics card, f.i. the Intel HD Graphics 4000.
- **Memory:** 16 GB of RAM
- **Storage:** Hard Disk

Recommended

- **Version:** Mac OS X 10.10 and up to Catalina
- **CPU:** Mac/Intel processor with 64-bit support
- **GPU:** Intel or AMD graphics card
- **Memory:** Don't go for less than 32 GB RAM.
Buy as much memory that you can afford and fits in the system.
- **Storage:** SSD is best. Other good options are Hard Disk and Network Drive.

For Machine Learning

- This is still to be determined.

Please note:

- *Machine Learning on GPU is not supported due to the absence of Nvidia cards for recent Mac OS X systems.*
- *Mac/PowerPC support is NOT available.*
- *Mac/ARM support is NOT yet available.*
- *Mac/Intel emulation in Rosetta 2 is not supported.*

- *A 3-button mouse is highly recommended.*

Installation

OpendTect

Installing OpendTect on Windows

OpendTect, the dGB Commercial plugins, Python packages and offline User documentation and dGB Plugins documentation can be installed via the OpendTect Installation Manager (OpendTect_Installer_win64.exe) or via an offline package.

There are some improvements in the installation manager:

- Installation packages are now grouped, e.g. all Python related packages are listed in group Python
- After installation the Windows Firewall rules can be set or unset.

Installation via Installation Manager

1. The Installation Manager is available for download via the appropriate platform link on the [download page](#) of the dGB website.

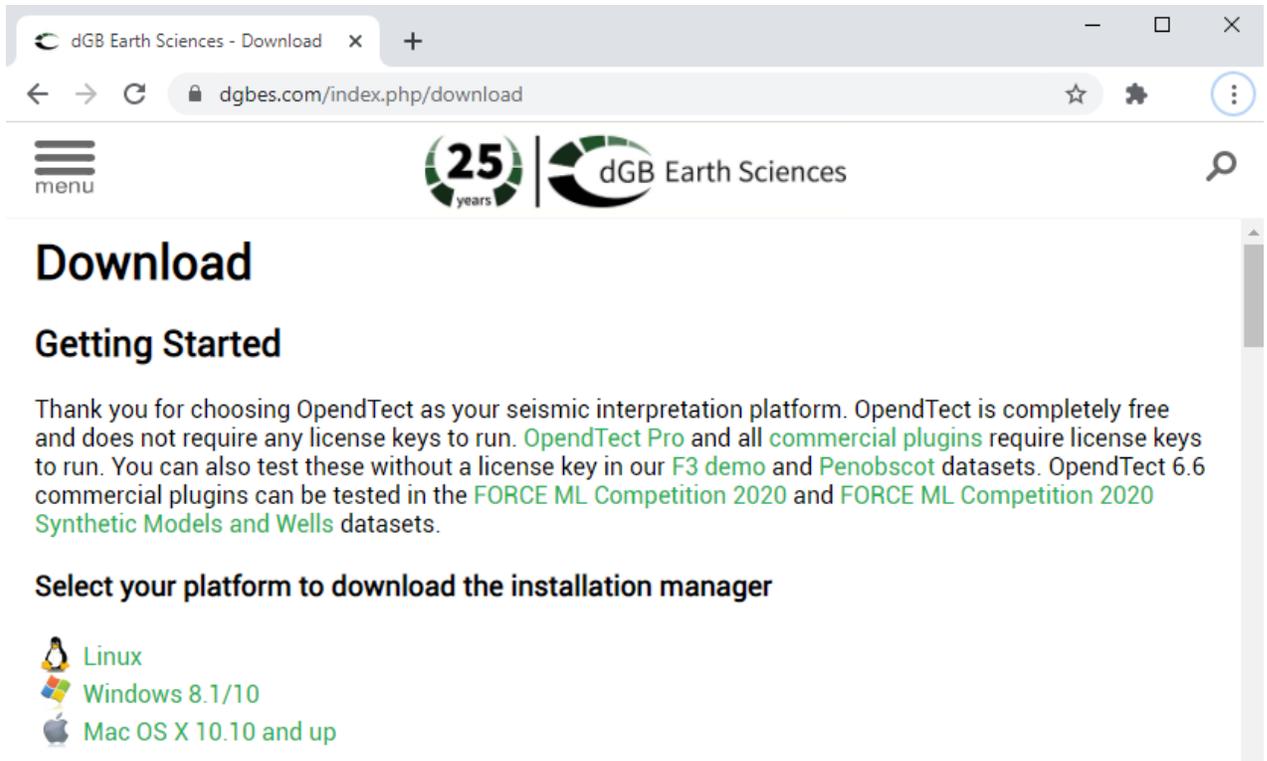


Figure: OpendTect download page

2. The Installation Manager is a wizard to install/update the existing OpendTect (Current / Previous) releases. The release type field is used to select the release that is needed to be installed/updated. The installer gives you the choices as seen below:

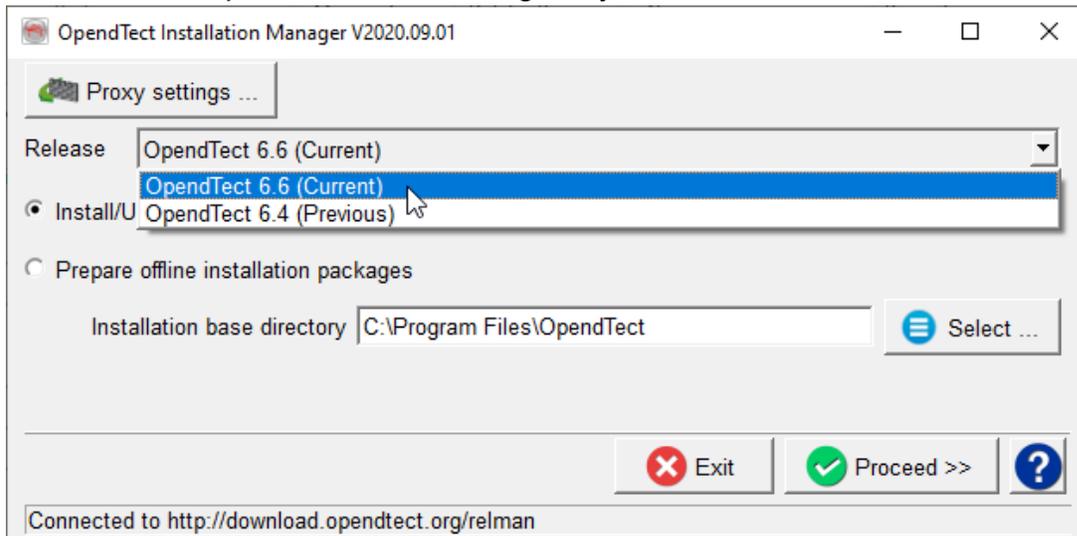


Figure: OpendTect Installation Manager

3. If it is a fresh installation then you may keep the proposed installation base directory or choose your own. If it is an update then select it's installation base directory. Now click Proceed.

Please note: installing in C:\Program Files\OpendTect will require administrative rights. In case you want to prevent using the administrative rights then we suggest to install into a folder on which a normal user has permissions, e.g. into C:\User-s\username\OpendTect

4. Select the package type of OpendTect you want to install. To read more about OpendTect packages types, please refer to our [webpage of licensing types](#).

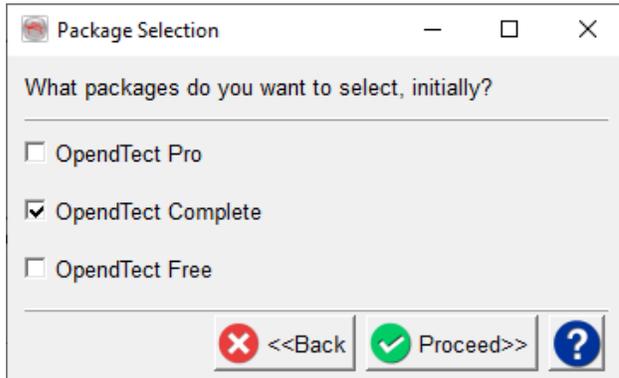


Figure: OpendTect Installer Package Selection window

5. The OpendTect Installation Manager identifies the platform on which it is running. This information is then anonymized prior to it being sent to OpendTect. We use this anonymous data solely for the purpose of getting a picture of OpendTect usage and thus improving our support capabilities.
6. The next window of the wizard is the OpendTect Package Manager. Multiple items can be toggled on or off by checking the boxes. Optionally, the relevant package combination could also be selected from the top list box.

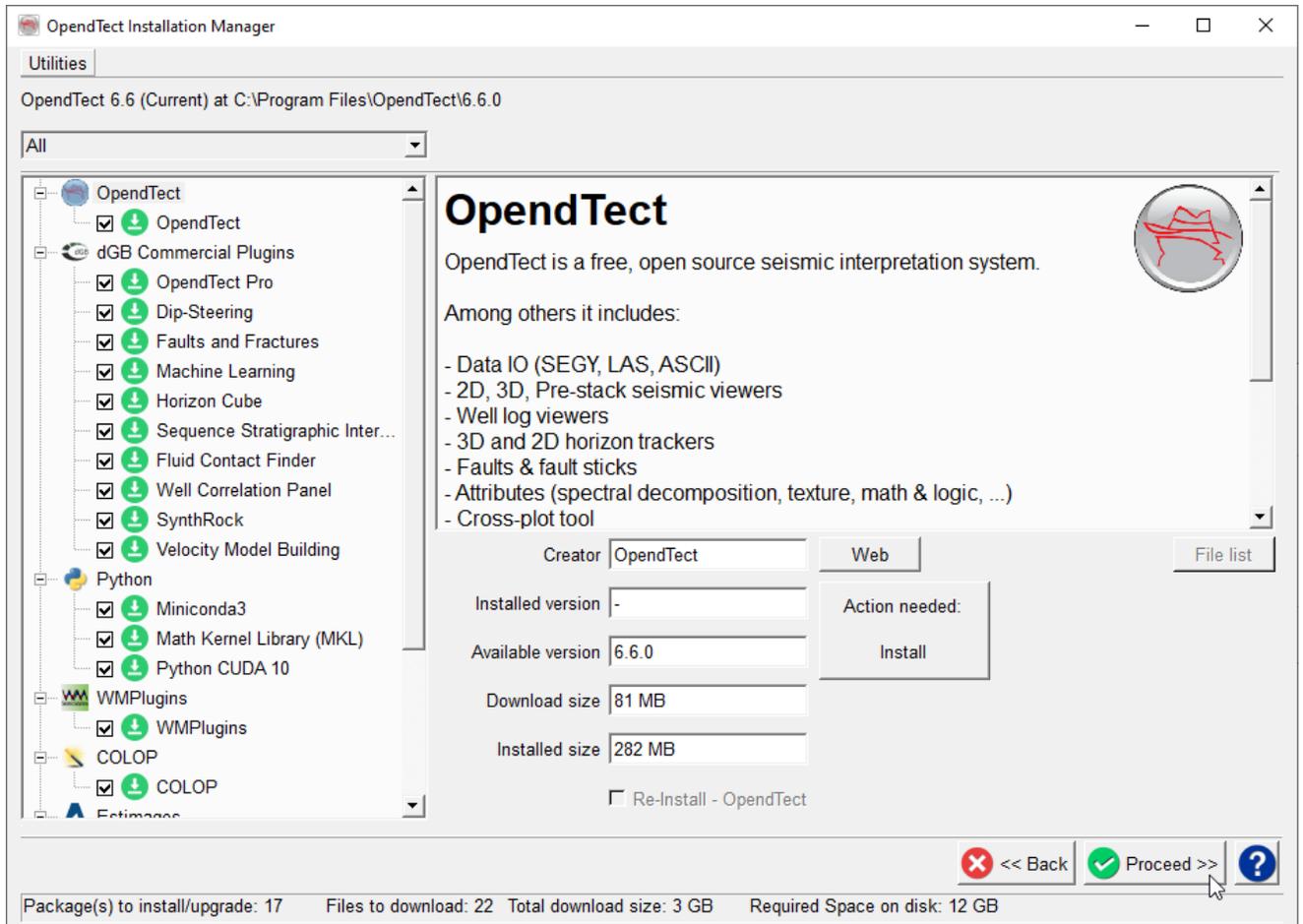


Figure: OpendTect Package Manager

7. The installation manager will automatically recognize the previously installed version at the selected path and will prompt it in the *Installed version* field. To read more about a particular item in the list, select the item by clicking on it and read the description on the right panel. For example, Machine Learning:

Machine Learning



Machine Learning is a toolkit for seismic, logs and seismic-to-logs classification, segmentation and regression. ML supports the latest deep learning algorithms (convolutional neural networks, ensemble methods, support vector machines etc.) and it fully integrates its predecessor: the **Neural Networks** plugin

[More info](#)

[Request evaluation license](#)

Creator	dGB Earth Sciences B.\	Web	File list
Installed version	-	Action needed:	
Available version	6.6.0	Install	
Download size	448 kB		
Installed size	1 MB		
<input type="checkbox"/> Re-Install - Machine Learning			

Figure: Machine Learning information in right panel of OpendTect Installation Manager

- Please note: In order to get an out of the box experience with the Machine Learning plugin it is easiest to select all Python packages. If you know for certain that you will only process on CPU or Nvidia GPU then select either the Math Kernel Library (MKL) Python environment or the CUDA 10 Python environment. The base Python environment will always be installed.
- After selecting the packages for installation click on Proceed to start the installation. You will first get a window that downloads the files from the download server. After downloading the packages they will be unpacked to the installation path.

[48%] Downloading files
47%
KBytes downloaded 173637 16s
Pause Abort

[79%] Archive unpacker
79%
Extracting data MBytes Processed: 721 15s
Pause Abort

Figure: Example of downloading files and archive unpacker windows

10. If, for any reason, you should choose to abort the installation mid-download, you will see the following window appear:

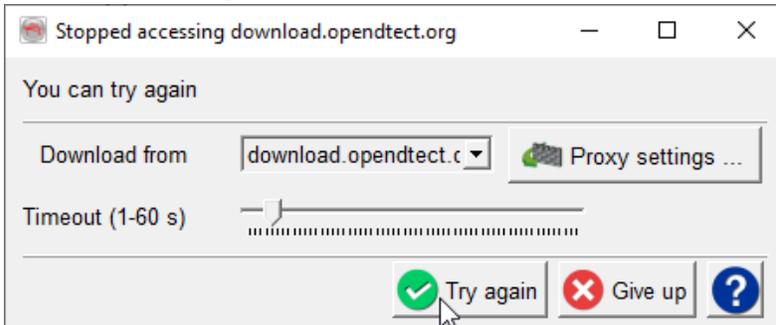


Figure: Stopped accessing download.opendtect.org

This gives you various options, including increasing the time-out from its default setting, changing the download server or changing the Proxy settings. One thing that can be tried is temporarily (partly) disable the Windows firewall to find out whether that helps.

11. The installation will finish with giving an overview of the installed packages.

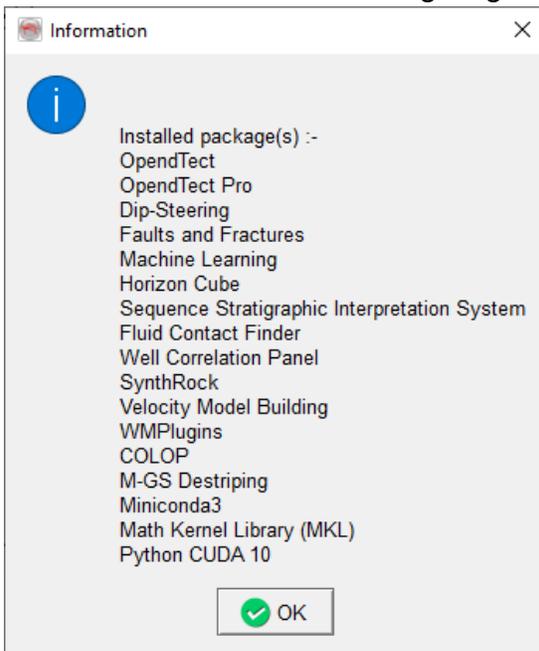


Figure: Overview of installed packages

12. Only for OpendTect 6.6: After installation is finished the Manage Firewall Program Rule window appears. This window gives the possibility to add rules for a selection of OpendTect and Python executables to the Windows Firewall. For more information please see the [Firewall rules and ports chapter](#).

Please note that for adding rules to the firewall elevated rights are needed.

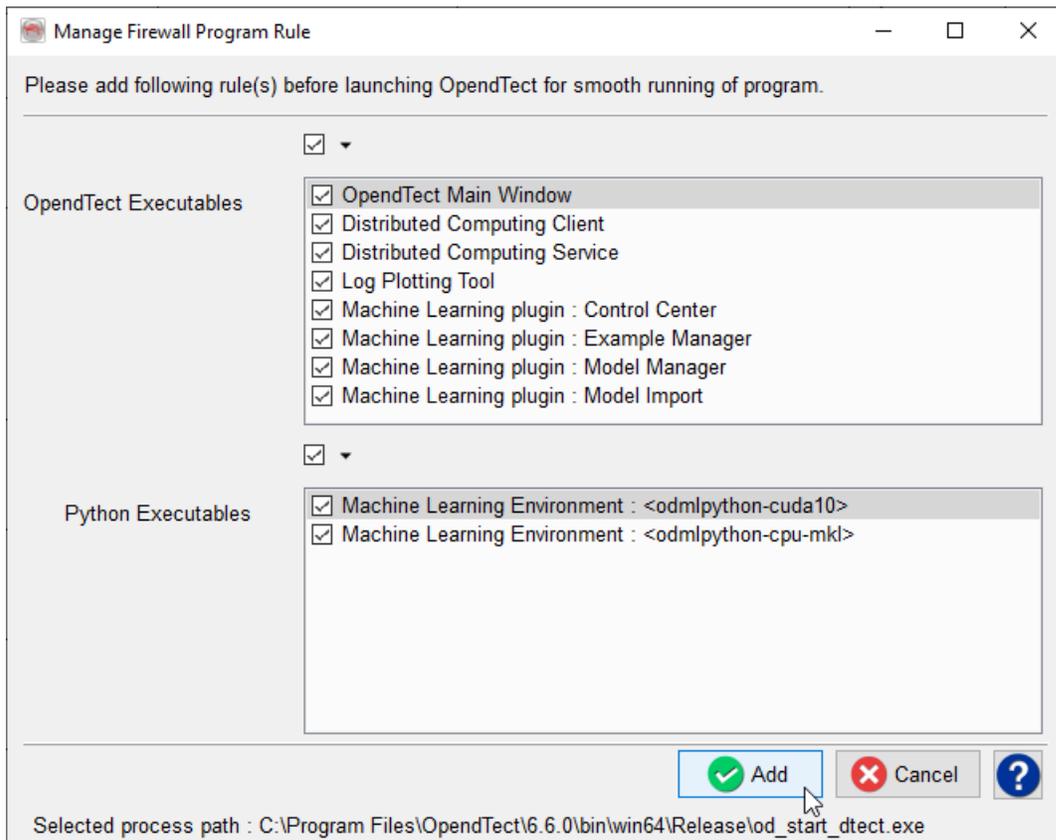


Figure: Manage Firewall Program Rule

Installer logging

The installer logs to a logfile. If something goes wrong during installation please look into the following files:

- %TEMP%\username_od_instmgr_install_log.txt
- %TEMP%\username_od_instmgr_updcheck_log.txt

Installing OpendTect on Linux

OpendTect, the dGB Commercial plugins, Python packages and offline User documentation and dGB Plugins documentation can be installed via the OpendTect Installation Manager (OpendTect_Installer_lux64.sh) or via an offline package.

There are some improvements in the installation manager:

- Installation packages are now grouped, e.g. all Python related packages are listed in group Python

Installation via Installation Manager

1. The Installation Manager is available for download via the appropriate platform link on the [download page](#) of the dGB website.

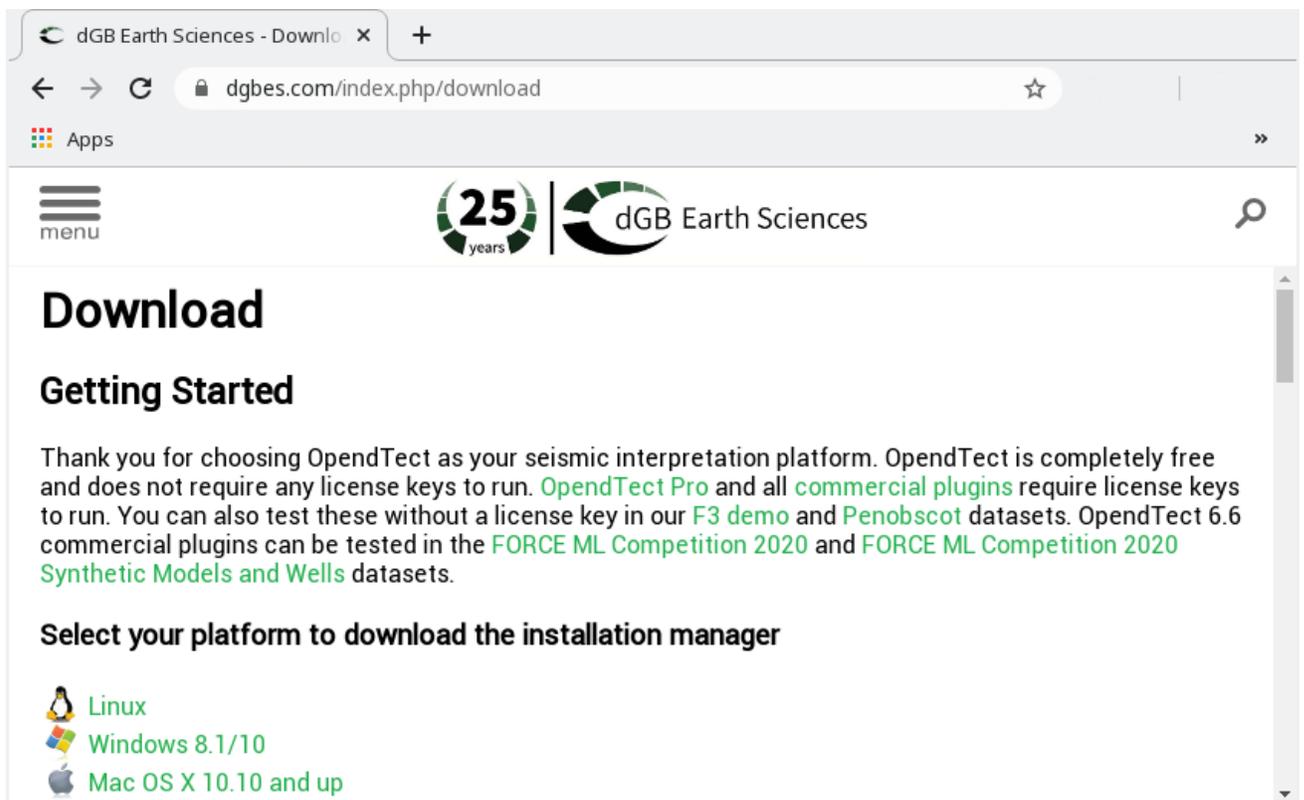


Figure: OpendTect download page

2. After downloading the installer open a terminal window.
3. In the terminal window change the directory to the directory where the installer has been downloaded to.

4. Then run `sh OpendTect_Installer_lux64.sh`
Please note: to be able to run the installer `csch` needs to be installed
5. This should unpack the OpendTect Installer into the `/tmp` directory and then run the Installer.
6. The Installation Manager is a wizard to install/update the existing OpendTect (Current / Previous) releases. The release type field is used to select the release that is needed to be installed/updated. The installer gives you the choices as seen below:

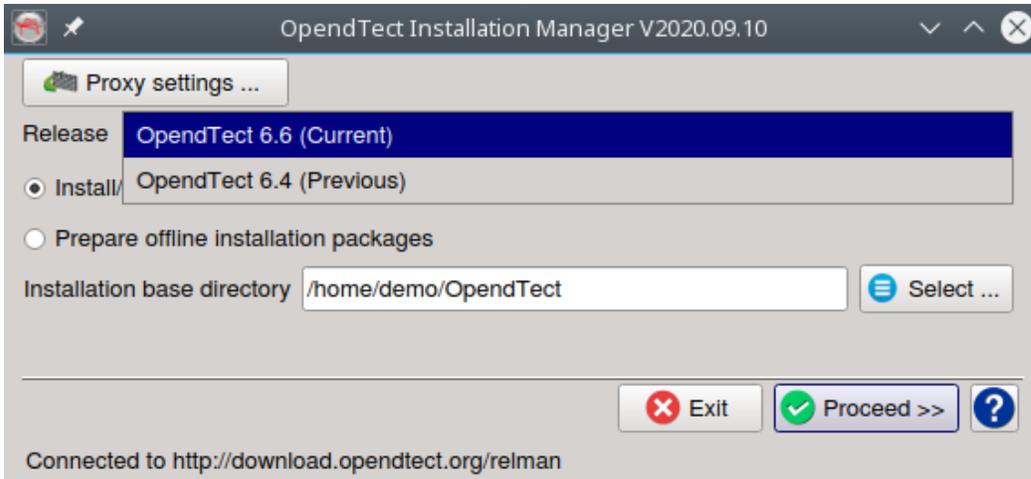


Figure: OpendTect Installation Manager

7. If it is a fresh installation then you may keep the proposed installation base directory or choose your own. If it is an update then select it's installation base directory. Now click Proceed.
8. Select the package type of OpendTect you want to install. To read more about OpendTect packages types, please refer to our [webpage of licensing types](#).

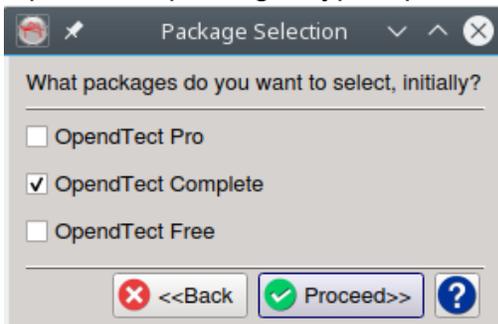


Figure: OpendTect Installer Package Selection window

9. The OpendTect Installation Manager identifies the platform on which it is running. This information is then anonymized prior to it being sent to OpendTect. We use this anonymous data solely for the purpose of getting a picture of OpendTect usage and

thus improving our support capabilities.

10. The next window of the wizard is the OpendTect Package Manager. Multiple items can be toggled on or off by checking the boxes. Optionally, the relevant package combination could also be selected from the top list box.

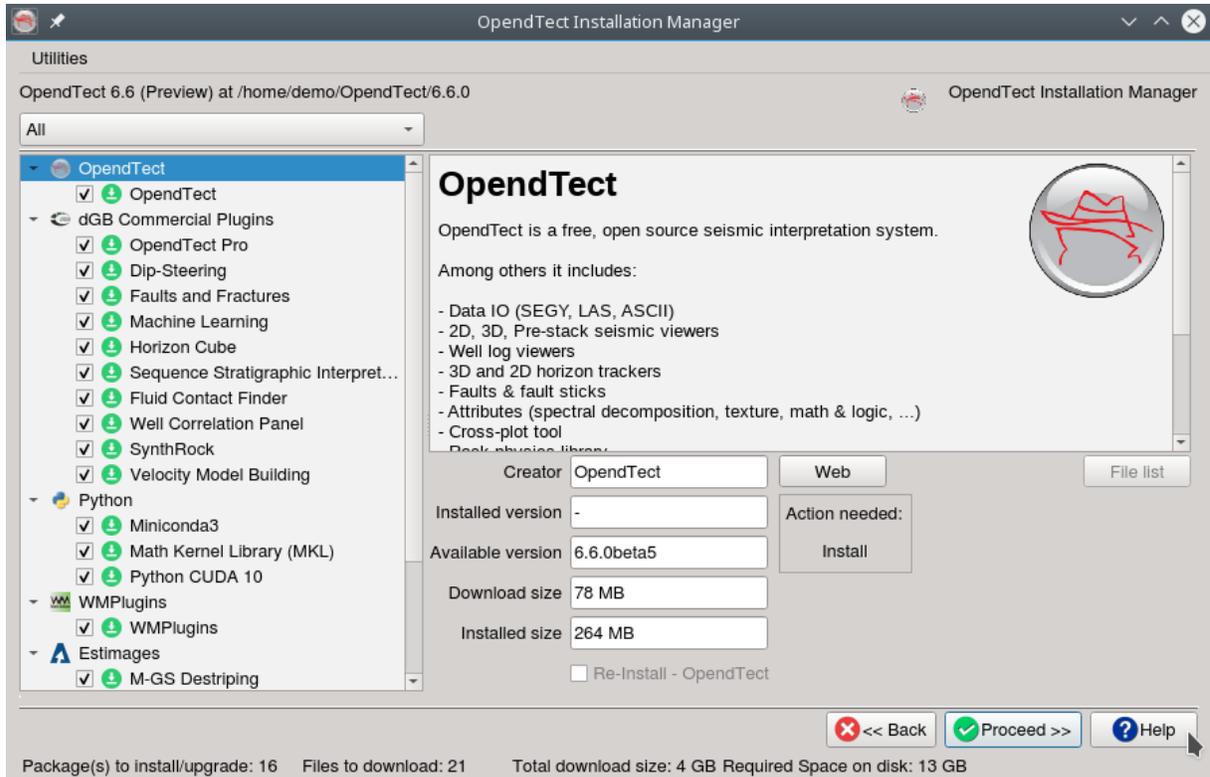


Figure: OpendTect Package Manager

11. The installation manager will automatically recognize the previously installed version at the selected path and will prompt it in the *Installed version* field. To read more about a particular item in the list, select the item by clicking on it and read the description on to the right panel. For example, Machine Learning:

Machine Learning

Machine Learning is a toolkit for seismic, logs and seismic-to-logs classification, segmentation and regression. ML supports the latest deep learning algorithms (convolutional neural networks, ensemble methods, support vector machines etc.) and it fully integrates its predecessor: the **Neural Networks** plugin



[More info](#)

[Request evaluation license](#)

Creator	dGB Earth Sciences B.V.	Web	File list
Installed version	-	Action needed:	
Available version	6.6.0	Install	
Download size	1 MB		
Installed size	4 MB		
<input type="checkbox"/> Re-Install - Machine Learning			

Figure: Machine Learning information in right panel of OpendTect Installation Manager

- Please note:** In order to get an out of the box experience with the Machine Learning plugin it is easiest to select all Python packages. If you know for certain that you will only process on CPU or Nvidia GPU then select either the Math Kernel Library (MKL) Python environment or the CUDA 10 Python environment. The base Python environment will always be installed.
- After selecting the packages for installation click on Proceed to start the installation. You will first get a window that downloads the files from the download server. After downloading the packages they will be unpacked to the installation path.

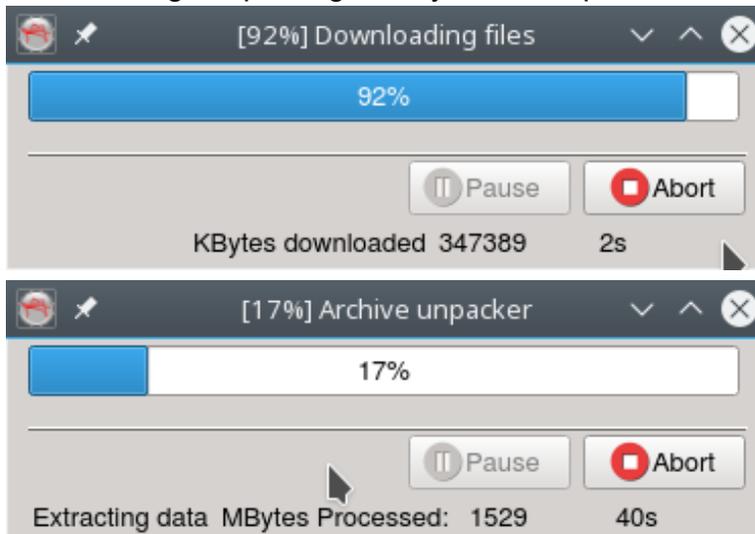


Figure: Example of downloading files and archive unpacker windows

14. If, for any reason, you should choose to abort the installation mid-download, you will see the following window appear:

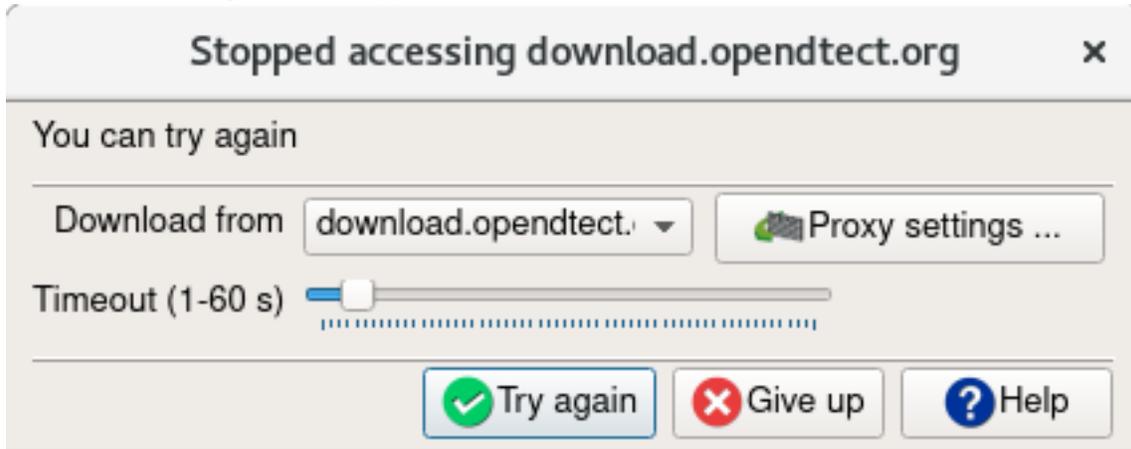


Figure: Stopped accessing download.opendtect.org

This gives you various options, including increasing the time-out from its default setting, changing the download server or changing the Proxy settings. One thing that can be tried is temporarily (partly) disable the Linux firewall or SELinux to find out whether that helps.

15. The installation will finish with giving an overview of the installed packages.

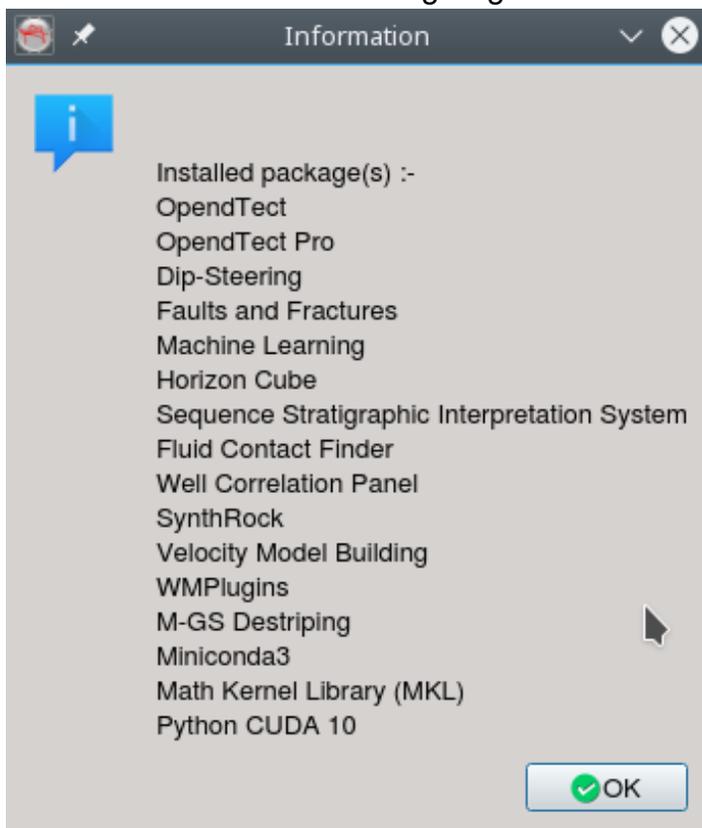


Figure: Overview of installed packages

Installer logging

The installer logs to a logfile. If something goes wrong during installation please look into the following files:

- /tmp/username_od_instmgr_install_log.txt
- /tmp/username_od_instmgr_updcheck_log.txt

Installing OpendTect on RHEL 6.x

The OpendTect Installation Manager and 6.6 are not supported for RHEL 6.x. It is however possible to install and run OpendTect 6.4 on RHEL 6.x and CentOS 6.x by installing via an offline installation package. This offline installation package can be created with the OpendTect Installation Manager. This can be done on a supported Linux Distribution or Windows.

To install the offline package:

1. Transfer the `od_offline_lux64` folder to the RHEL6.x or CentOS 6.x system

Then on the destination system in terminal:

2. `cd od_offline_lux64`

3. `./unix_install.csh`

Please note: to be able to run the installer `csh` needs to be installed

4. The installer asks for the path of where to install OpendTect, e.g. `/home/user-/OpendTect`
5. After installation is finished change the directory to the OpendTect installation directory
6. Run `./start_dtect` to start OpendTect

On RHEL 6.x and CentOS 6.x there can be error messages that will prevent you from starting OpendTect.

```
You                               might                               get:
/usr/lib64/libstdc++.so.6: version `GLIBCXX_3.4.14' not
found                               (required                               by                               /home/user-
/OpendTect/6.4.0/bin/lux64/Release/libosgQt5.so.145)
```

You might be able to solve this by installing a newer `libstdc++` package. Here are some instructions to get it installed. There are no guarantees these instructions will work for your RHEL6.x or CentOS 6.x system. However, please note that it worked on our CentOS 6.x system:

- `wget https://rpmfind.net/linux/centos/7.8.2003/os/x86_64/Packages/libstdc++-4.8.5-39.e17.x86_64.rpm`
- `yum install libstdc++-4.8.5-39.e17.x86_64.rpm`

When installing the newer libstdc++ package you will probably get the issue that you need a newer libc.so.6. Here are some instructions to get it installed. There are no guarantees these instructions will work for your RHEL6.x or CentOS 6.x system. However, please note that it worked on our CentOS 6.x system:

- `wget http://copr-be.cloud.fedoraproject.org/results/mosquito/myrepo-el6/epel-6-x86_64/glibc-2.17-55.fc20/glibc-2.17-55.el6.x86_64.rpm`
- `wget http://copr-be.cloud.fedoraproject.org/results/mosquito/myrepo-el6/epel-6-x86_64/glibc-2.17-55.fc20/glibc-common-2.17-55.el6.x86_64.rpm`
- `wget http://copr-be.cloud.fedoraproject.org/results/mosquito/myrepo-el6/epel-6-x86_64/glibc-2.17-55.fc20/glibc-devel-2.17-55.el6.x86_64.rpm`
- `wget http://copr-be.cloud.fedoraproject.org/results/mosquito/myrepo-el6/epel-6-x86_64/glibc-2.17-55.fc20/glibc-headers-2.17-55.el6.x86_64.rpm`
- `yum install glibc-2.17-55.el6.x86_64.rpm glibc-common-2.17-55.el6.x86_64.rpm glibc-devel-2.17-55.el6.x86_64.rpm glibc-headers-2.17-55.el6.x86_64.rpm`

Installing OpendTect on Mac OS X

OpendTect, the dGB Commercial plugins, Python packages and offline User documentation and dGB Plugins documentation can be installed via the OpendTect Installation Manager (OpendTect_Installer_mac.dmg) or via an offline package.

There are some improvements in the installation manager:

- Installation packages are now grouped, e.g. all Python related packages are listed in group Python

Security setting

Please do the following before installation:

- Click on the 'Apple' button (top left), go to:
- System Preferences > Security & Privacy > General Tab
- In case the option is there then toggle on *Allow apps to be installed from App Store and identified developers*, this may ask for admin credentials.
- Keep this window open when starting to install OpendTect. When you start the installer for the first time you might get the error message: *"OpendTect Installer" can't be opened because the identity of the developer cannot be confirmed.*

At System Preferences > Security & Privacy > General Tab there should be a message like *"OpendTect Installer" was blocked from opening because the identity of the developer" cannot be confirmed.* Click the lock icon to make changes and then click the Open Anyway button.

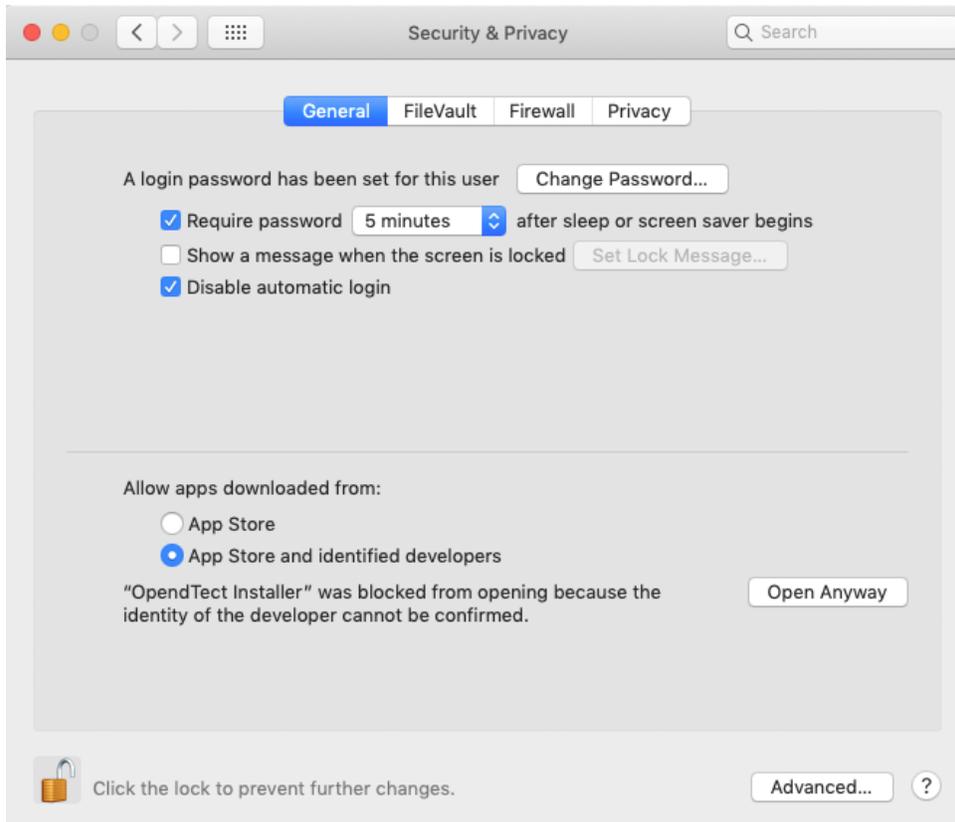


Figure: Set Security setting to App Store and identified developers and Open OpendTect Installer anyway

- Once the installation is finished, and assuming you made a change to the security setting then you may change the security back.

Please note that in order to install OpendTect into the Applications folder the user that installs OpendTect should be allowed to administer this computer. You can check whether this option is toggled on via Users & Groups.

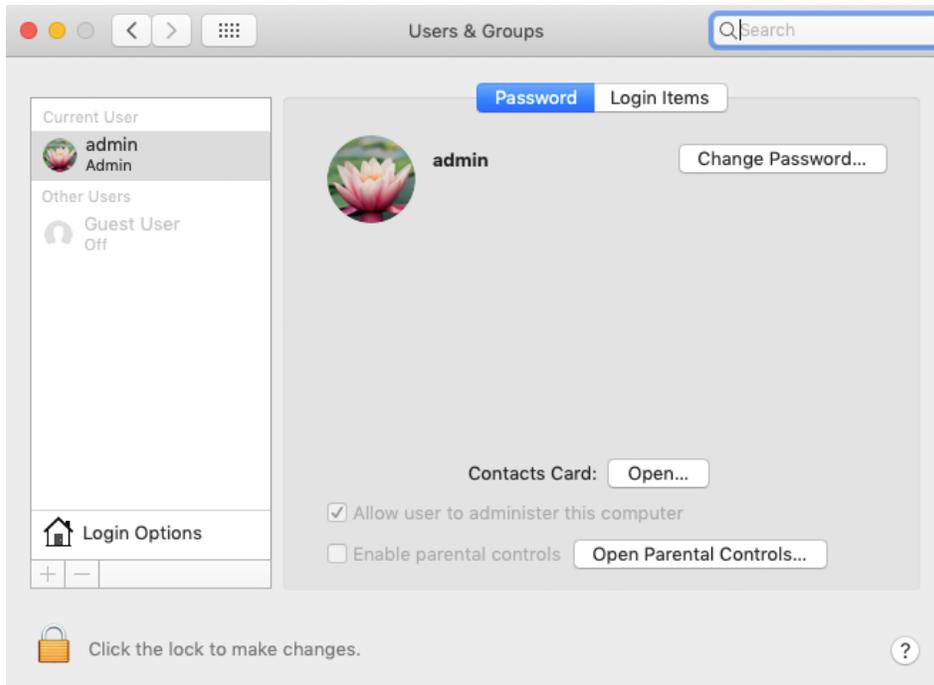


Figure: Find out whether user is allowed to write into Applications folder via Users & Groups

Installation via Installation Manager

1. The Installation Manager is available for download via the appropriate platform link on the [download page](#) of the dGB website.

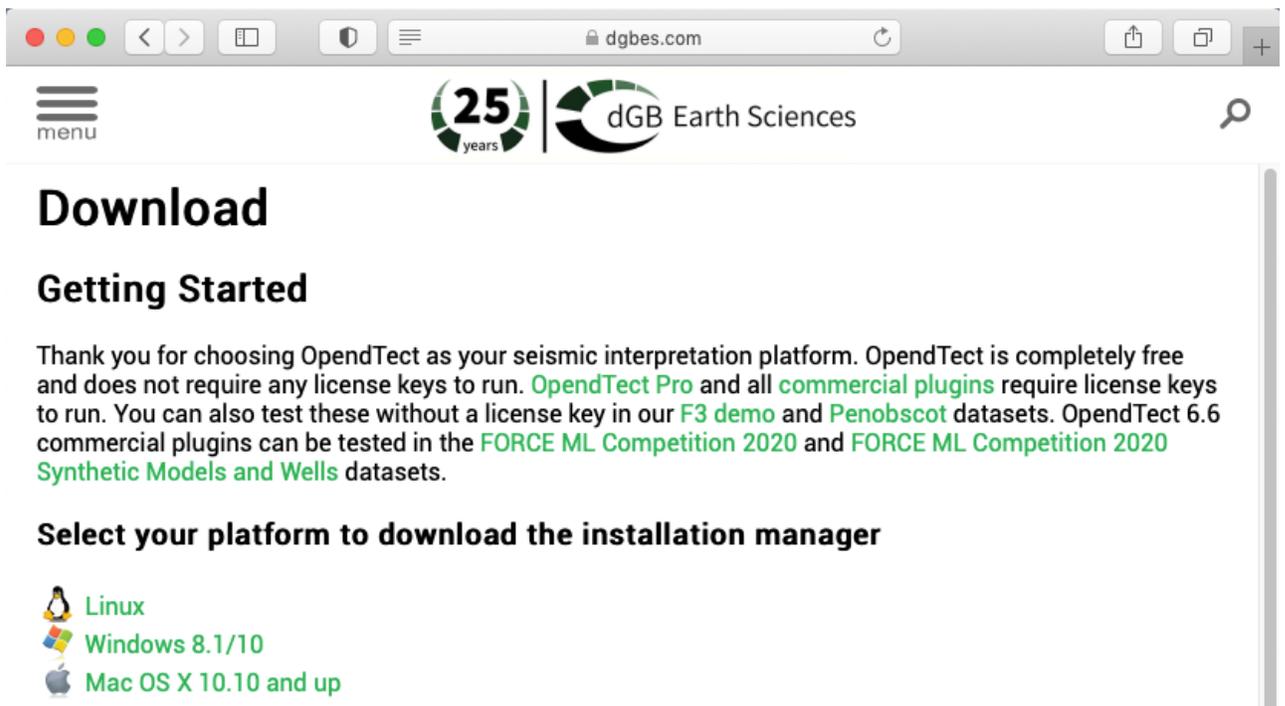


Figure: OpendTect download page

2. Run the .dmg file via Finder.
3. Then start the OpendTect Installer.

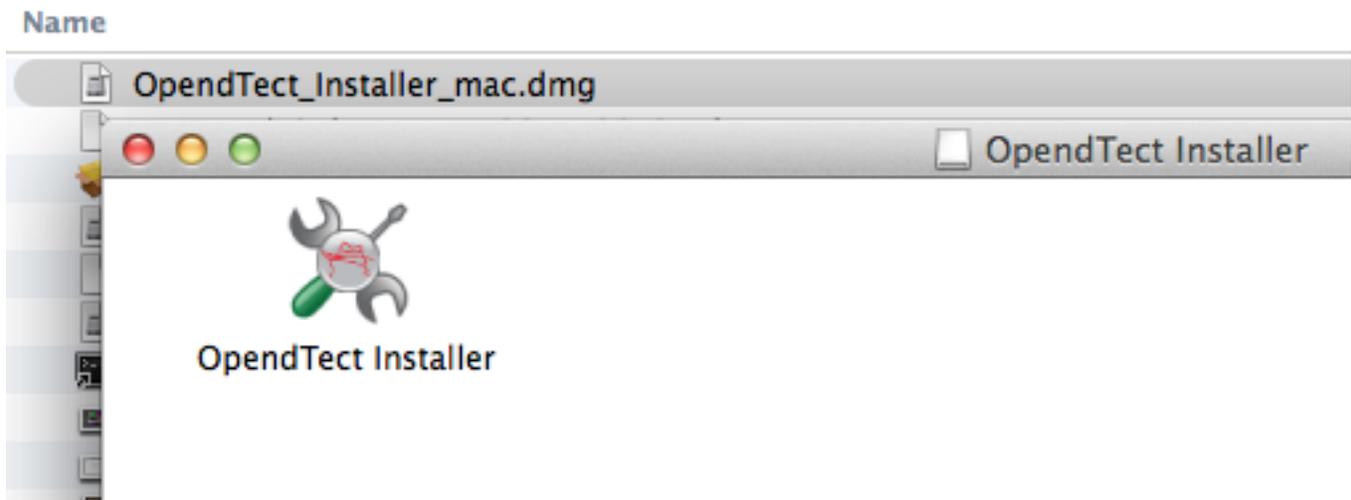


Figure: Start the OpendTect Installer by opening the .dmg file

4. In case you get a window with error message: *"OpendTect Installer" can't be opened because the identity of the developer cannot be confirmed.* then please follow the Security settings section of this document to allow the opening of the OpendTect Installer.

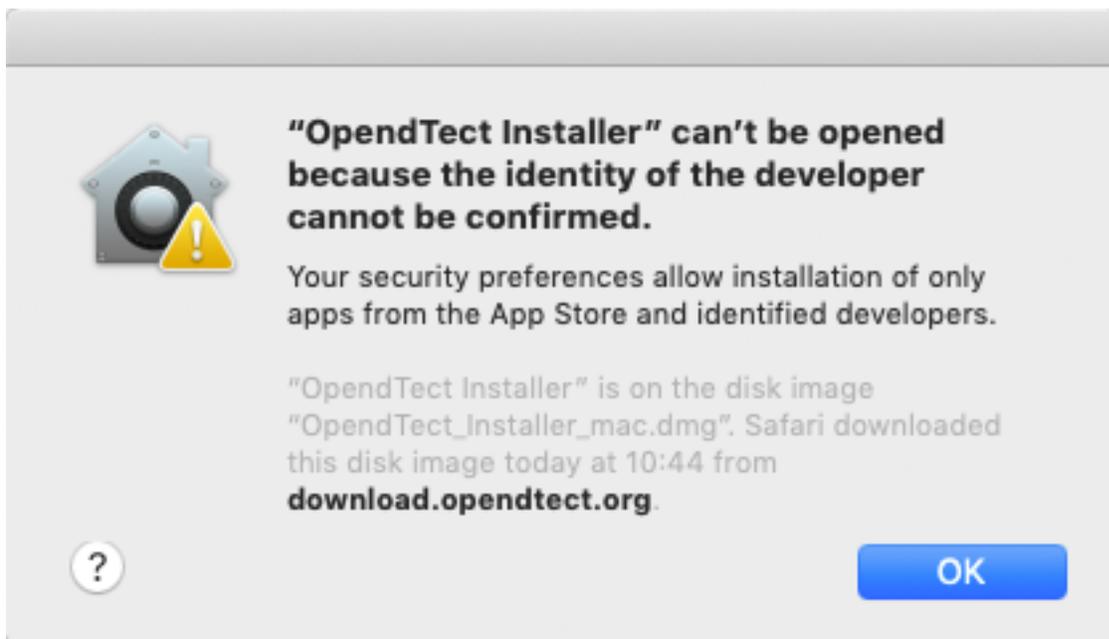


Figure: Error message explaining that the OpendTect Installer can't be opened

5. The Installation Manager is a wizard to install/update the existing OpendTect (Current / Previous) releases. The release type field is used to select the release that is needed to be installed/updated. The installer gives you the choices as seen below:

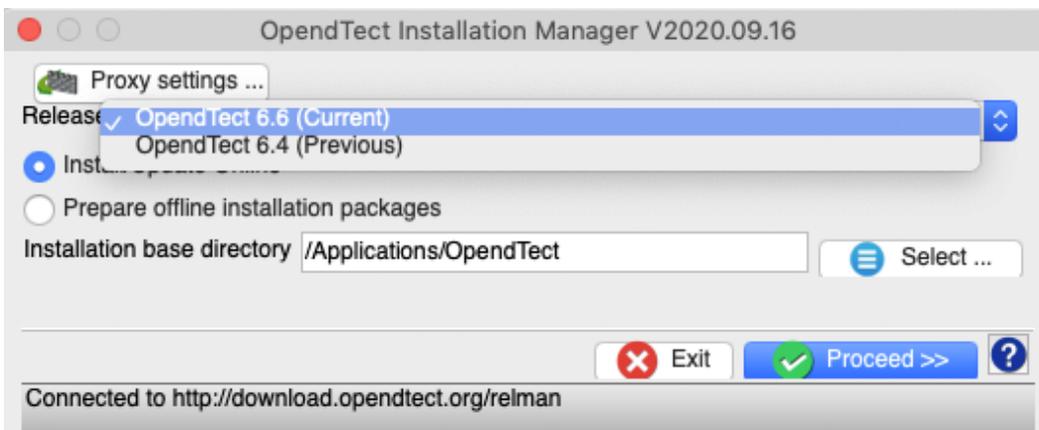


Figure: OpendTect Installation Manager

6. If it is a fresh installation then you may keep the proposed installation base directory or choose your own. If it is an update then select it's installation base directory. Now click Proceed.
7. Select the package type of OpendTect you want to install. To read more about OpendTect packages types, please refer to our [webpage of licensing types](#).

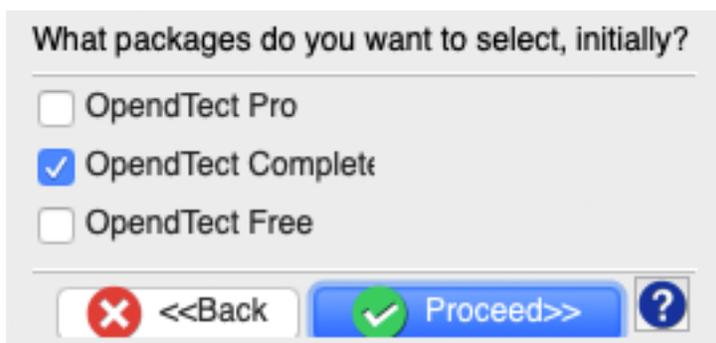


Figure: OpendTect Installer Package Selection Window

8. The OpendTect Installation Manager identifies the platform on which it is running. This information is then anonymized prior to it being sent to OpendTect. We use this anonymous data solely for the purpose of getting a picture of OpendTect usage and thus improving our support capabilities.
9. The next window of the wizard is the OpendTect Package Manager. Multiple items can be toggled on or off by checking the boxes. Optionally, the relevant package combination could also be selected from the top list box.

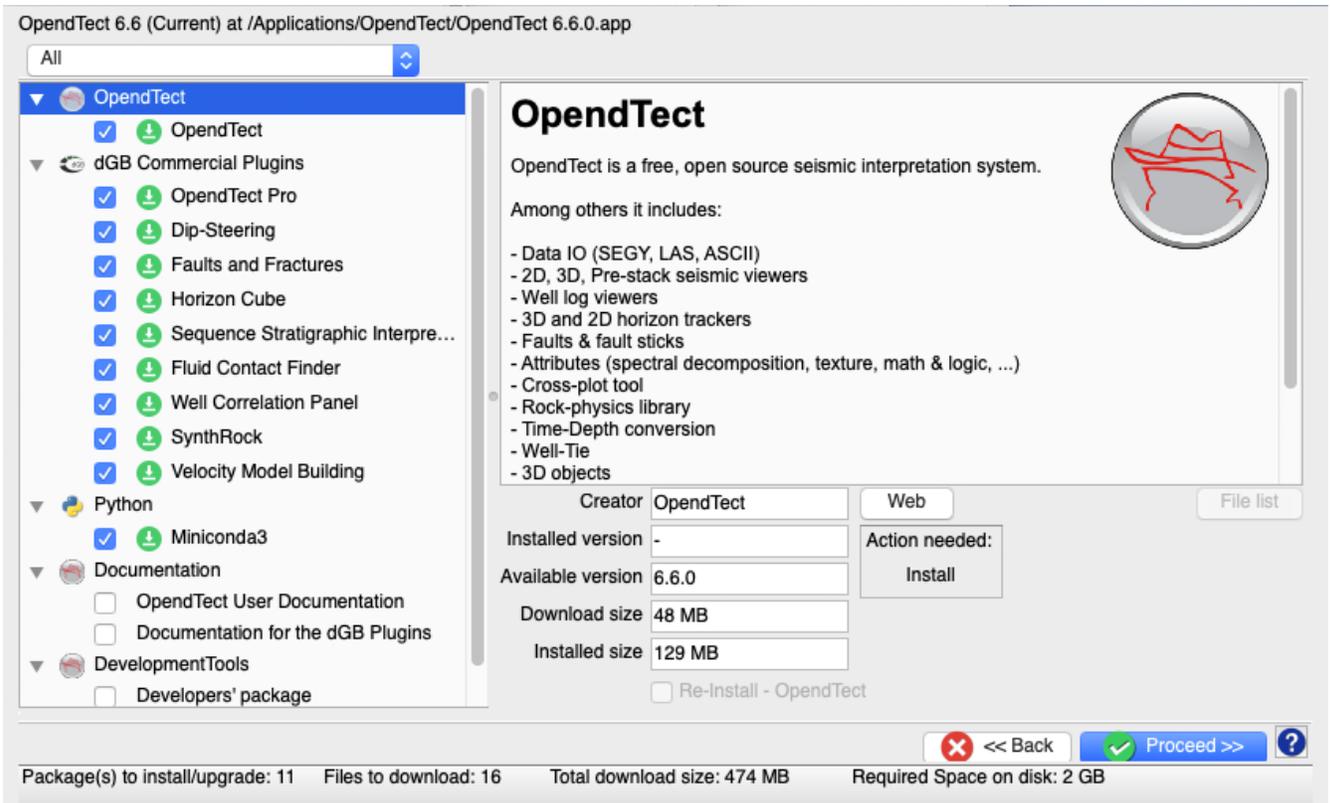


Figure: OpendTect Package Manager

10. The installation manager will automatically recognize the previously installed version at the selected path and will prompt it in the *Installed version* field. To read more about a particular item in the list, select the item by clicking on it and read the description on to the right panel.
11. After selecting the packages for installation click on Proceed to start the installation.
12. You might get a window with error message You do not have write permission in /Applications/OpendTect. This means that the current user has no administrative rights to install into the Applications folder. The solution is to either give the current user administrative rights or to install into a folder where the user is allowed to write files, like e.g. the users Documents/OpendTect folder.

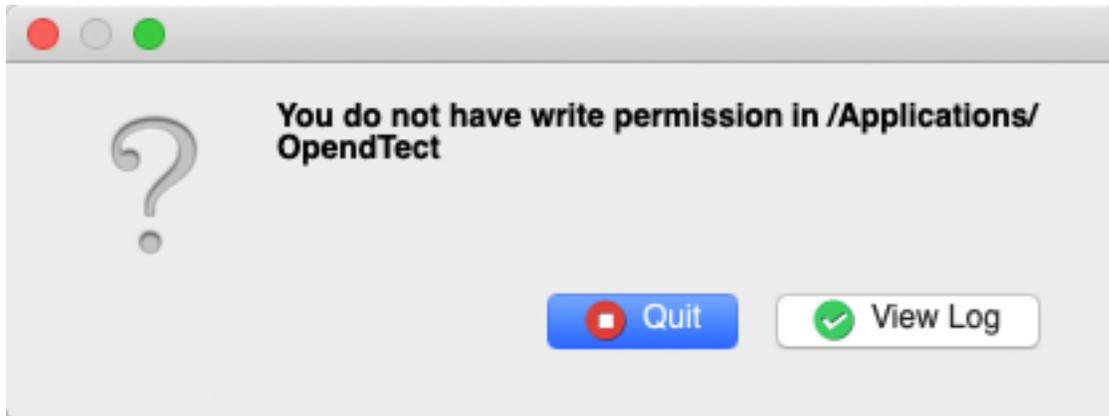


Figure: No write permissions in /Applications/OpenText

13. You will first get a window that downloads the files from the download server. After downloading the packages they will be unpacked to the installation path.

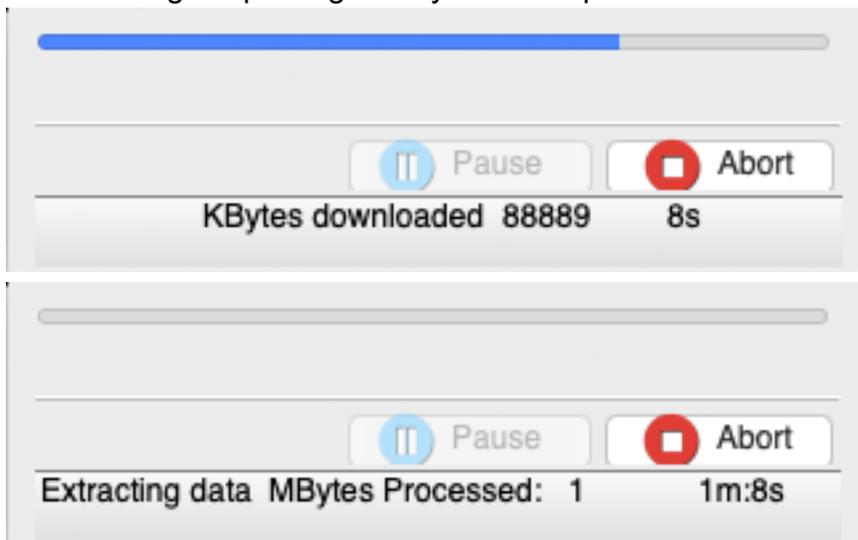


Figure: Example of downloading files and archive unpacker windows

14. If, for any reason, you should choose to abort the installation mid-download, you will see the following window appear:

This gives you various options, including increasing the time-out from its default setting, changing the download server or changing the Proxy settings.

15. The installation will finish with giving an overview of the installed packages.

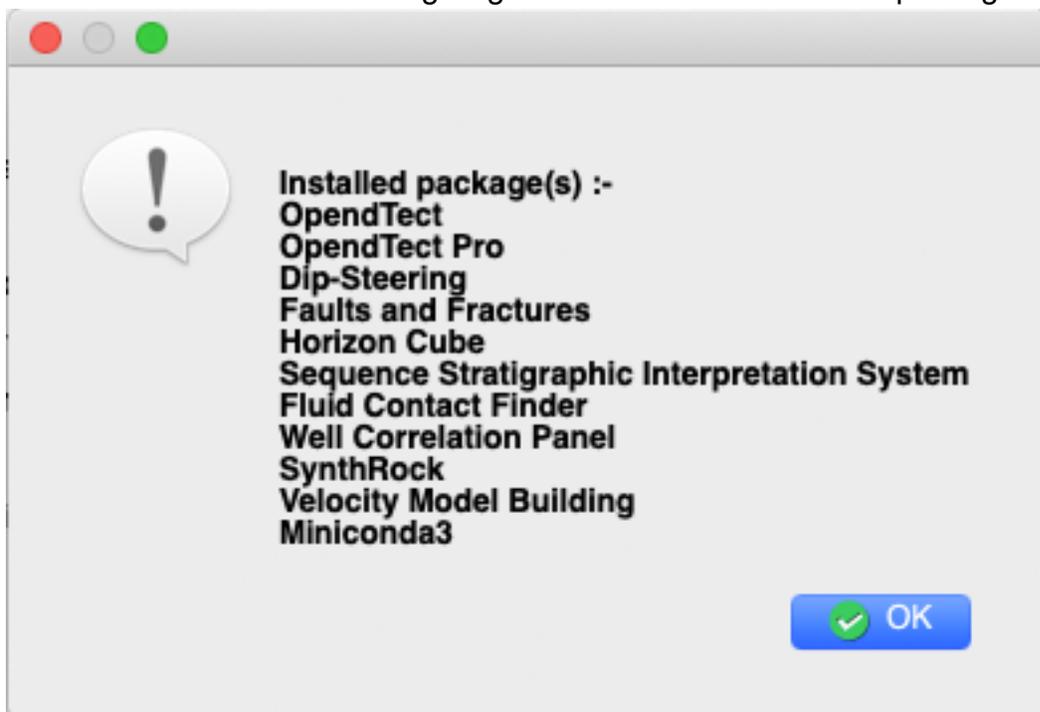


Figure: Overview of installed packages

16. To start OpendTect open it with Finder: Applications > OpendTect > OpendTect 6.6.0

Licenses

Licensing and Host IDs

OpendTect uses FlexNet to manage its licensing. There are two main types of license:

- **Node-locked:** The license file is tied directly to a specific client machine (or selection of machines) through its 'HostID'. This option is popular for laptops and for single-user desktop setups. Installation is very simple.
- **Floating License:** The license file is generated for, and tied to a dedicated server via its 'HostID'. The license manager on the server then issues licenses to client machines. This option is more suited for multiple users. Installation can be more involved.

In order to generate a license, we need the HostID (commonly known as MAC address) of the machine or server, depending on license type:

- If server-based (floating) license
 - Server (Host) Name
 - Server (Host) ID
- If node-locked license
 - Host ID

To discover the HostID(s):

- OpendTect Pro (all platforms): Utilities > Installation > Licenses > Show HostID...

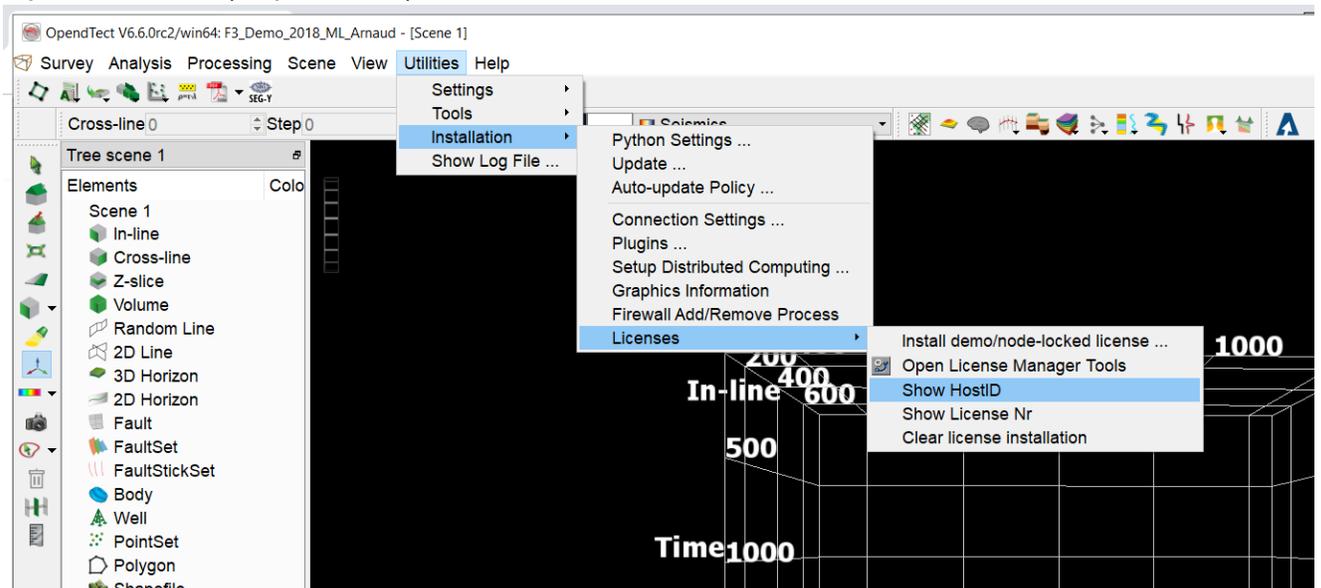


Figure: The menu option to show the machine's HostID.

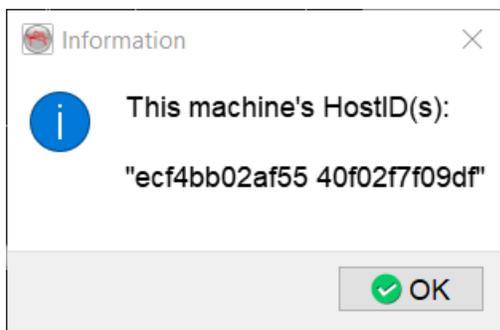


Figure: Showing the machine's HostID via OpendTect

Note: Licenses > Show HostID... does not appear under the Utilities > Installation... options in the GPL version. You will have to install OpendTect Pro to access this utility.

For additional information (including alternate methods of accessing the HostID), please refer to the [Appendix FlexNet Licenses Explained](#).

Installing demo/node-locked licenses

Under *Utilities > Installation > Licenses* you will see two sets of options, differing per platform:

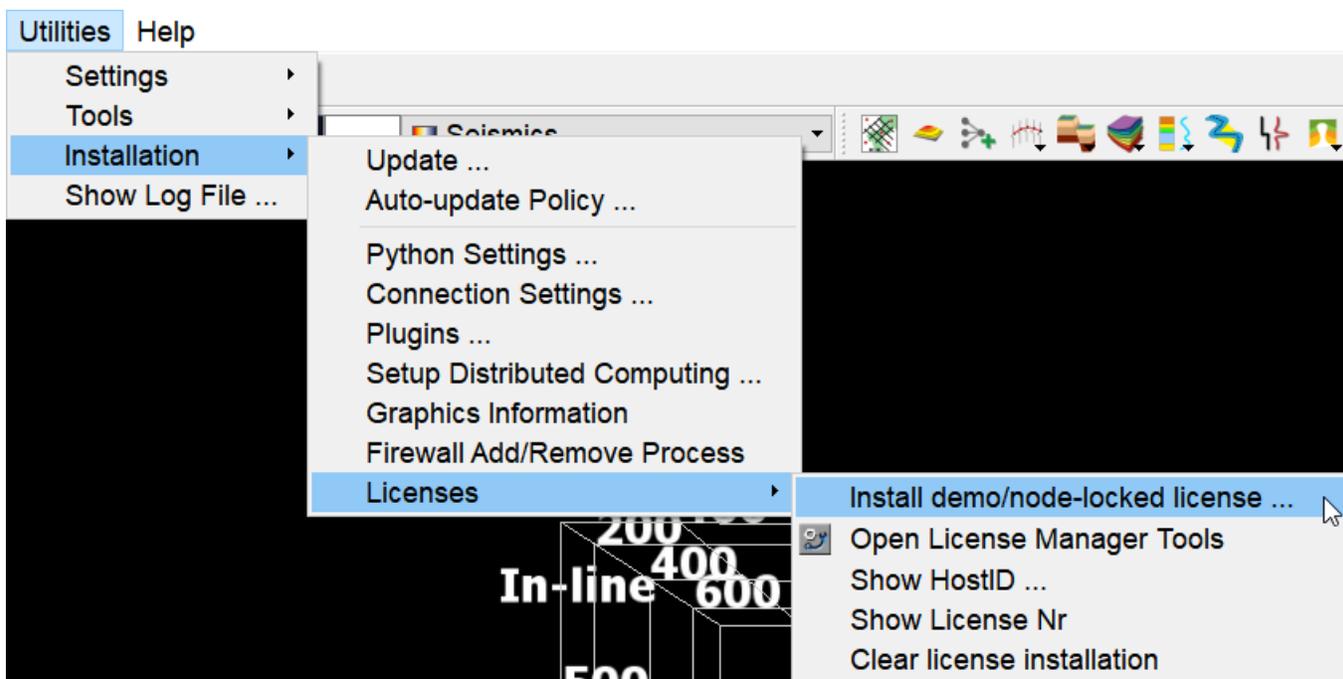


Figure: License options on Windows

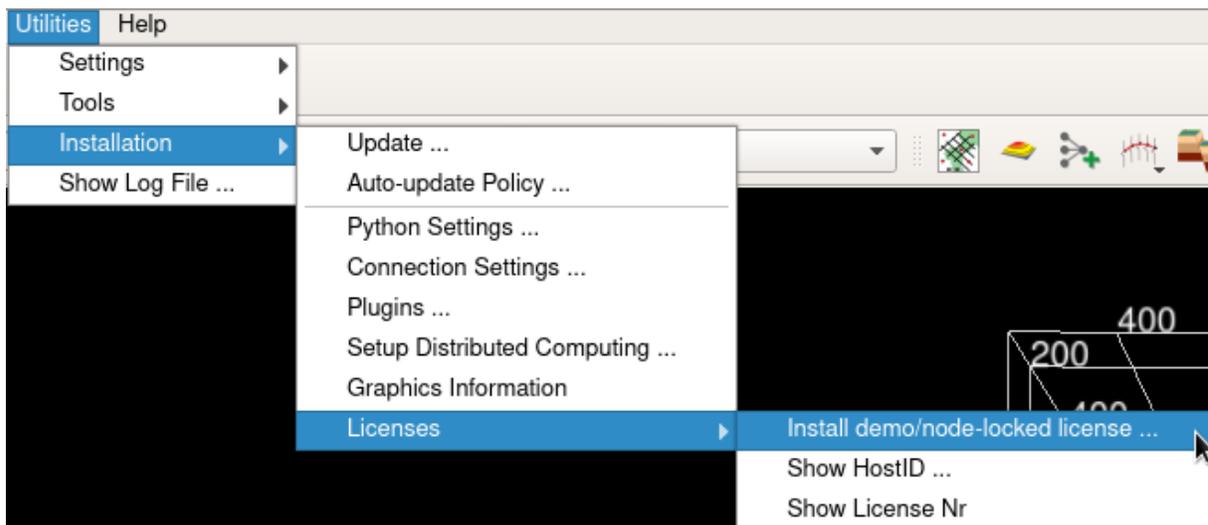


Figure: License options on Linux

For information about floating or server-based licenses, please refer to the [Appendix FlexNet Licenses Explained](#) or look in the chapter [installing floating licenses](#) for instructions on how to get them installed on the client.

For more general information about OpendTect licensing options, please see the [licenses](#) page

A more complete explanation of OpendTect license installation can be found in the [License Installation Webinar](#), available on [OpendTect's YouTube Channel](#).

Installing floating licenses

A floating license file is to be used by the dGB FlexNet license server. This license server can run on Linux, Mac or Windows. See here for instructions on [how to install the license server on Linux](#) and [how to install the license server on Windows](#).

Once the license server is running we can then instruct the client on how to borrow the license. There are two ways for installing the floating license on a client:

1. By setting the DGBLD_LICENSE_FILE environment variable
2. By using the FlexNet Impath tool via console/command prompt or on Windows through the License Manager Tools (lmtools.exe)

Setting the DGBLD_LICENSE_FILE environment variable

The DGBLD_LICENSE_FILE environment variable can be set to a license file or to the server host.

On a Linux client the variable can be set with the export or setenv command.

On a Windows client the variable can be set via Control Panel > System > Advanced system settings > Environment Variables. Please note that a user environment variable is only available for the user who sets the variable. It might be a better choice to set the variable as a system environment variable. It then can be globally accessed by all users.

The possible values are:

- port@host
- @host
- path to license file

- port = the TCP/IP port the license server is using. If port is not specified in front of @home a default TCP/IP port number in the range of 27000-27009 is used.
- host = the hostname of the license server. This comes from the SERVER line in the license file.
- path to license file = exactly that. Place the license file in a network location available to all machines running OpendTect or copy it to all of the machines running OpendTect (Typically: C:\Licenses)

Using the FlexNet Impath tool

The Impath tool can be used to add to, override, or get the current license path settings.

This and other FlexNet tools are packaged in a single executable `lmutil` or `lmutil.exe`.

OpenTect 6.6.0 ships these tools. They can be found in the following paths:

- For Linux: `OpenTect/6.6.0/bin/lux64/lm.dgb`
- For Windows: `OpenTect\6.6.0\bin\win64\lm.dgb`

To set it on Linux run:
`./lmutil lmpath -override dgbld port@host or ./lmutil lmpath -override dgbld licensefile`

On Windows run:
`lmutil.exe lmpath -override dgbld port@host or lmutil.exe lmpath -override dgbld licensefile`

Note: Impath sets the file `$HOME/.flexlmrc` on Linux and the FlexNet Licensing registry entry on Windows platforms.

To display the current license path setting, use on Linux the command:

`./lmutil lmpath - status`

Or on Windows:

`lmutil.exe lmpath -status`

For a status of the currently in use licenses, use on Linux the command:

`./lmutil lmstat -a -c port@host`

Or on Windows:

`lmutil.exe lmstat -a -c port@host`

Clear license installation

Under *Utilities > Installation > Licenses* you will see the option Clear license installation:

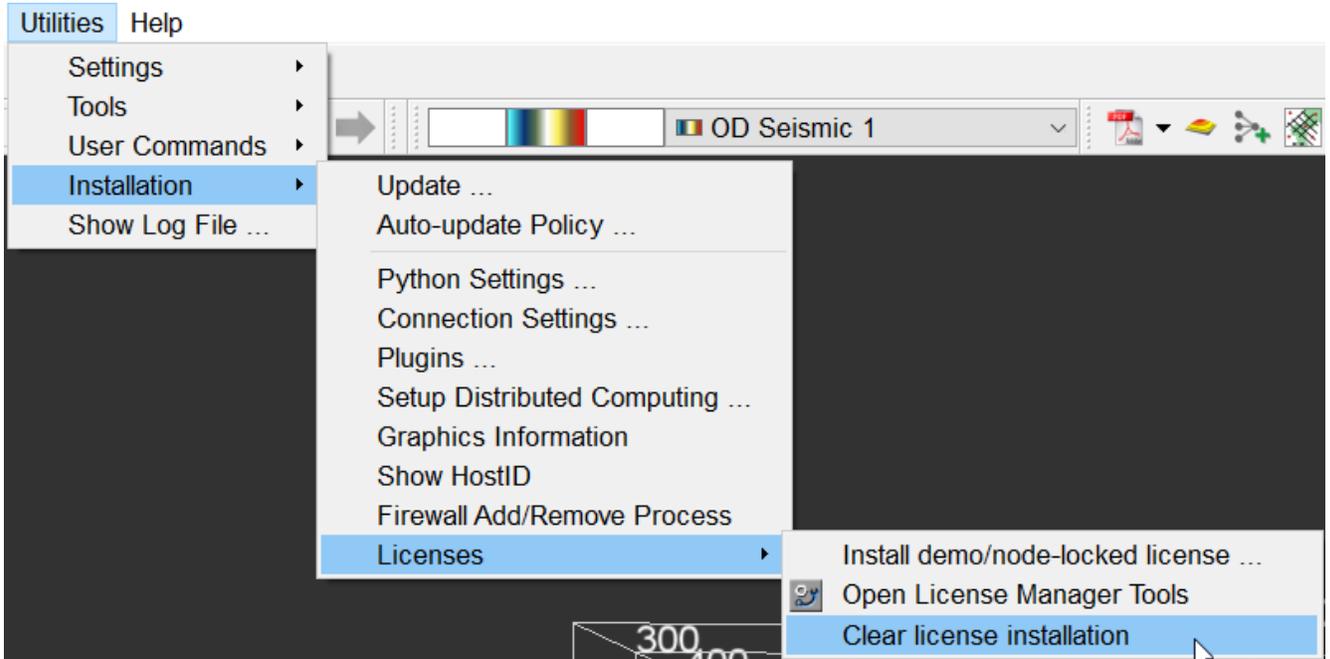


Figure: Clear license installation option

This option (Windows only) will clear:

- Demo or node-locked licenses installed via any route, including the 'Install demo license' option.
- Floating (or 'server') licenses that may have been installed (without stopping the license server).

You will get this window explaining that all commercial licenses will be deactivated and that the plugins will stop working. To remove click Remove or else Cancel.

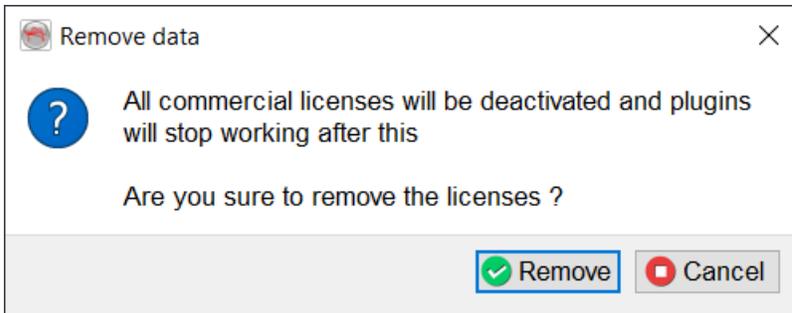


Figure: Are you sure to remove the licenses?

Once cleared, you will be prompted to restart:

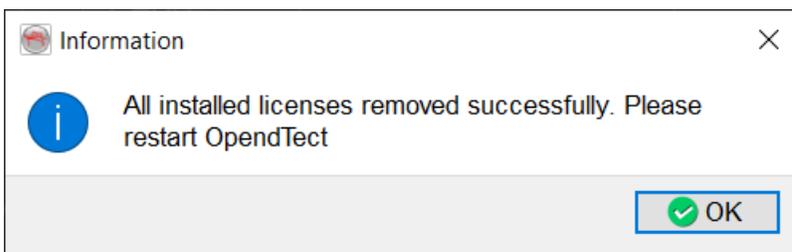


Figure: All installed licenses cleared

Users of Linux systems wishing to clear their license installation will need to do the following:

- Locate the `.flexlmrc` file in your HOME directory (eg: `$HOME/.flexlmrc`)
- Check in the file for specific lines referring to the OpendTect vendors (dGB, ARKCLS)
- If the file contains lines relevant to other software, then just delete the individual lines. Otherwise, you may choose to delete the file.

This method also applies to both demo/node-locked and floating licenses and will also not stop the server.

Prepare offline installation packages

Offline installation packages can be created with the OpendTect Installation Manager. It is useful to do this if you want to install OpendTect on a system that has no access to Internet. Please follow these steps to create such offline installation packages:

1. Download the OpendTect Installation Manager for Linux or Windows from the [OpendTect download page](#).
2. Run the installer
 - a. On Linux in the terminal run `sh OpendTect_Installer_lux64.sh`
 - b. On Windows simply double click on `OpendTect_Installer_win64.exe`
3. In the Installation Manager window select Prepare offline installation packages.
 - a. On Linux the default path of where the `od_offline_lux64` or `od_offline_win64` Download directory will be created will be `/tmp/` . On Windows this is the user's `AppData\Local\Temp`. It is possible to select another path.
 - b. At platform choose between Linux (64 bits) or Windows (64 bits).
4. Click the Proceed button to start creating the offline installation package.

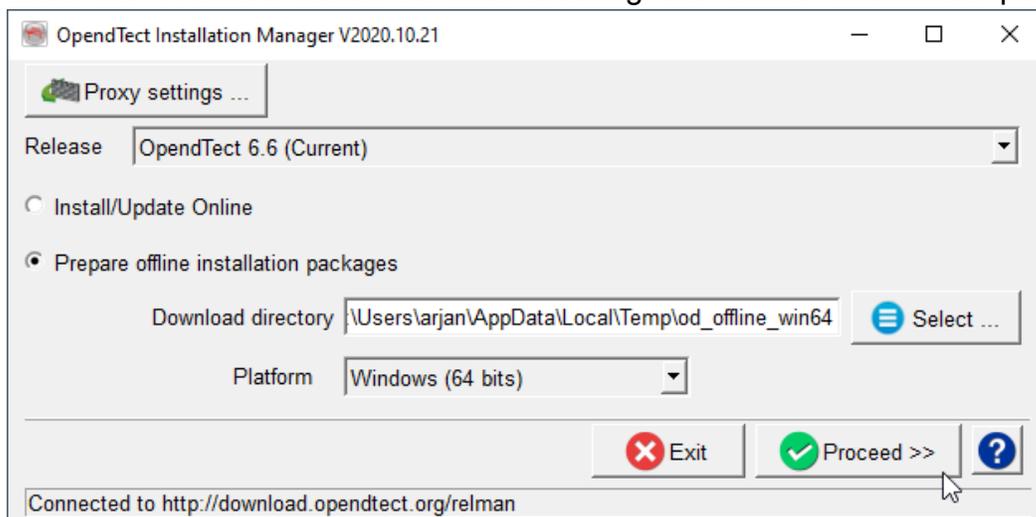


Figure: Prepare offline installation packages

5. The rest of the setup is identical to the usual way of installing OpendTect. You pick the packages that need to be included in the installation and click the *Proceed* button.
6. When the offline installation package has been created this will be shown in the information window.

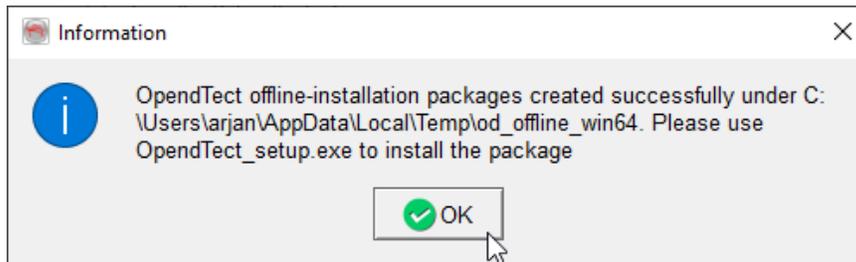


Figure: OpendTect offline-installation packages created successfully

7. The offline installation folder can now be transferred to the offline system by copying the folder via an USB stick or some other way onto the offline system.

Install OpendTect via offline installation package

On Linux:

1. Transfer the `od_offline_lux64` folder to the offline system.

Then on the offline system in terminal:

2. `cd od_offline_lux64`

3. `./unix_install.csh`

Please note: to be able to run the installer `csh` needs to be installed

4. The installer asks for the path of where to install OpendTect, enter it, e.g. `/home/user-/OpendTect`

5. The questions about a previous installed version of OpendTect and the initial Project data folder can be answered or ignored.

6. After a while the installation will be finished.

7. If the Python packages have been installed then in the terminal run the following to prepare the Python environment(s) on the system:

```
cd OpendTect/Python
./setup.Python
```

8. OpendTect can be started by running the `start_dtect` script, which can be found in the `OpendTect/6.6.0` folder.

On Windows:

1. Transfer the `od_offline_win64` folder to the offline system.
2. With the file explorer go into the `od_offline_win64` folder.
3. Run `OpendTect_setup.exe`

4. Set the destination folder and press the *Install* button to start the installation.

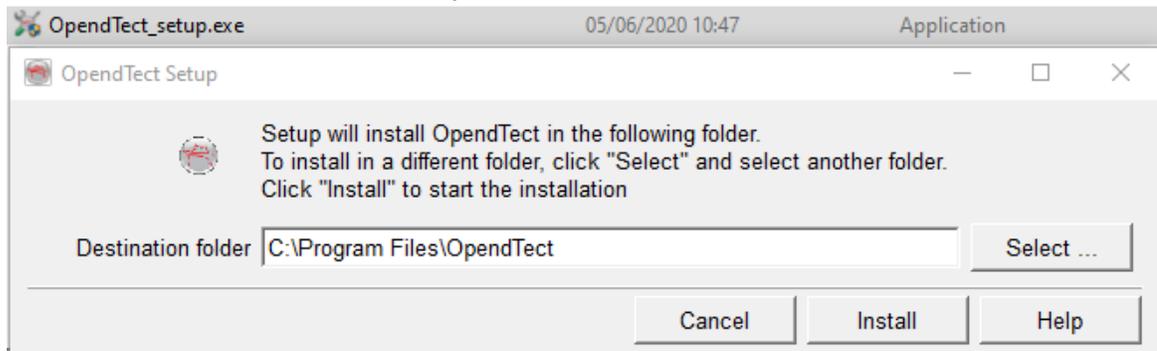


Figure: Install OpendTect with the offline installation package on Windows

5. After a while the installation will be finished.
6. OpendTect can be started by running the `od_main.exe` executable, which can be found in the `OpendTect\6.6.0\bin\win64\Release` folder.

Setup Python Settings

For information about the Python environments and how to install them please see the following chapters:

- [Information about the OpendTect Python environments](#)
- [Installing Python environments with the OpendTect installer](#)
- [Installing Python environments from YAML files](#)

Setup a Custom Python environment

To use the environment that has been installed with the YAML file or if you have set a custom Python environment in some other way, please do the following:

1. Start OpendTect 6.6
2. From the menu choose Utilities > Installation > Python Settings
3. Set Python environment to Custom
4. As Custom environment root choose the root directory of your own miniconda3 or anaconda3 installation, e.g. *C:\ProgramData\Anaconda3*.
5. With anaconda3 it is possible to install the environments into a different path than the root directory. From OpendTect 6.6.3 and onwards OpendTect will allow this by reading out the paths from *%UserProfile%\conda\environments.txt* on Windows or *\$HOME/.conda/environments.txt* on Linux.
6. As Virtual environment choose *odmlpython-cpu-mkl* or *odmlpython-cuda10*. The virtual environments are shown as full paths when they have been read from the *environments.txt* file.

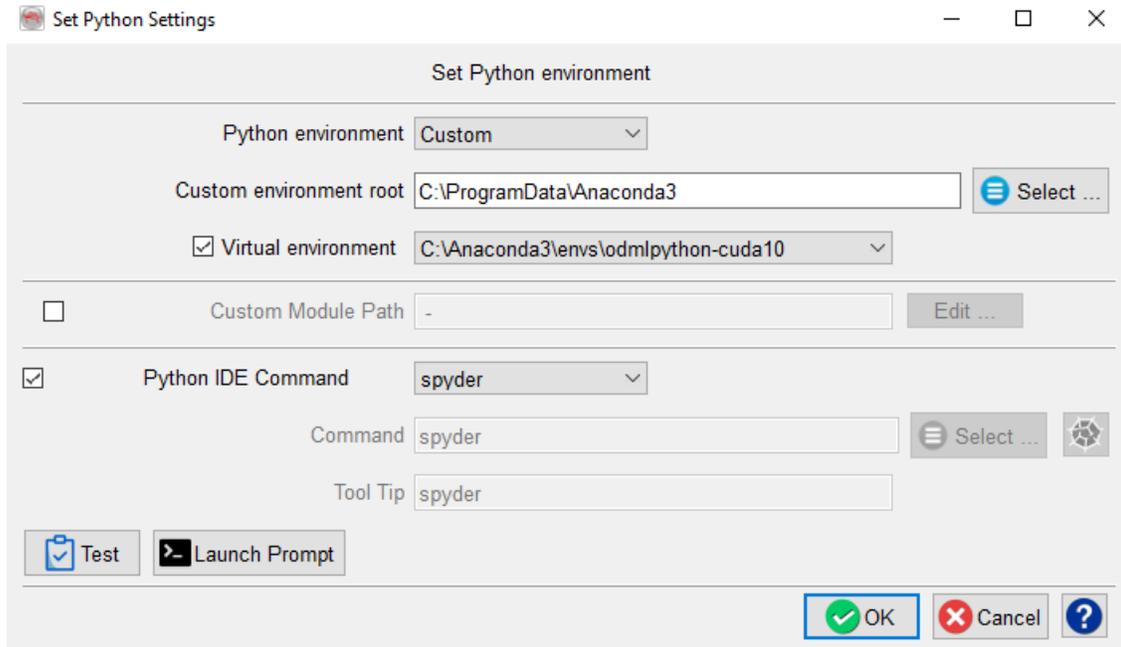


Figure: Python Settings window with a selected custom Python environment

7. If you want to use a Python IDE you can choose spyder or idle3 that come with the environment or choose your own IDE.
8. Press OK to save the Python settings.

Setup Distributed Computing

In order to utilize OpendTect's capability for Multi-Machine Processing (MMP), a *BatchHosts* file must be created and used. This file contains the list of remote machines (*host machines* or *nodes*) and some relevant details about these machines and the path to the Survey Data Root. OpendTect will use this file to communicate to the remote hosts and launch processes remotely on them. Follow the example format (shown below) to add the list of remote machines and their details in the respective fields.

In order to minimize complications, the Setup Distributed Computing tool (formerly known as Setup Batch Processing tool) can be used to create a tailor-made BatchHosts file (via *Utilities > Installation > Setup Distributed Computing...*).

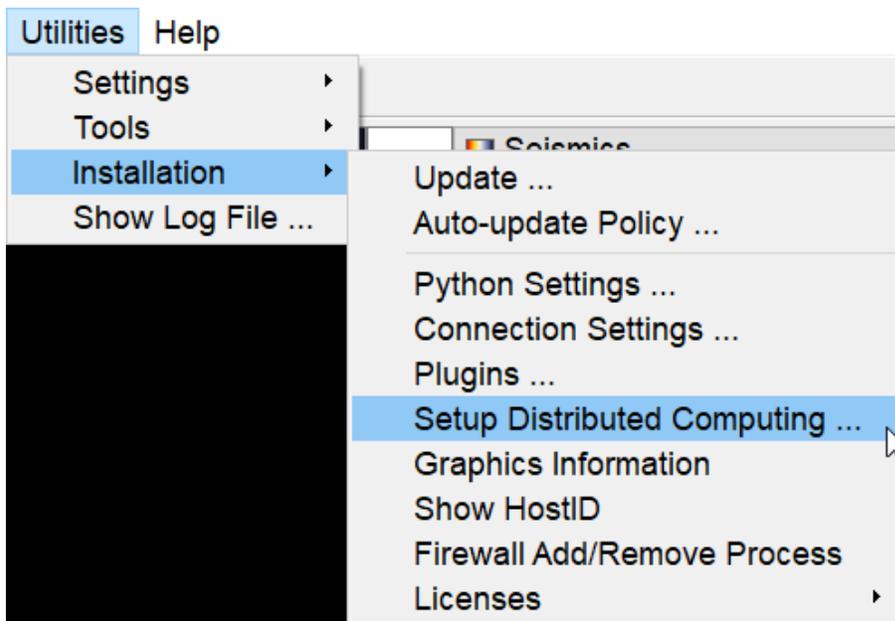


Figure: Launching Setup Distributed Computing tool

As default OpendTect will try to create a new or edit the existing BatchHosts file in its 6.6.0/data directory. If this directory is not writable OpendTect will advise to launch this process with administrator rights.

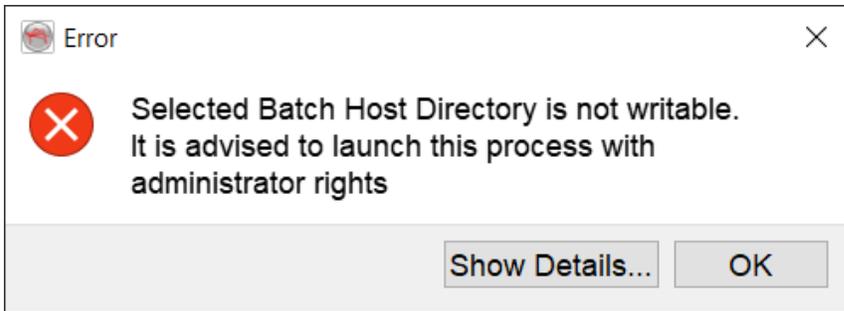


Figure: Selected Batch Host Directory is not writable

It is also possible to use a custom BatchHosts filepath by setting environment variable `DTECT_BATCH_HOSTS_FILEPATH`.

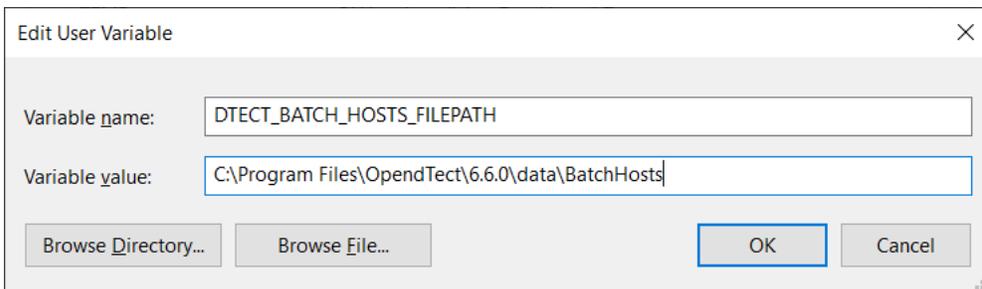


Figure: Setting `DTECT_BATCH_HOSTS_FILEPATH` environment variable

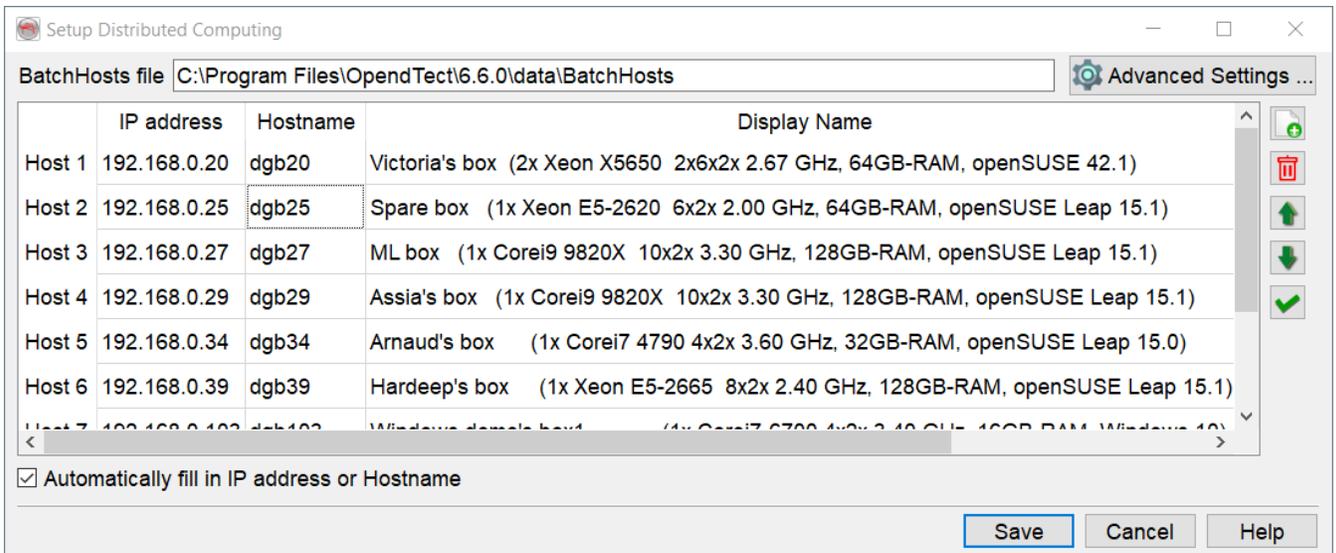


Figure: Setup Distributed Computing window

BatchHosts file: This field is not editable in the User Interface.

IP address: IP address of the node machine(s). If the background of this field is in red then there is a problem with the resolving of the hostname into the IP address.

Hostname: Hostname of the node machine(s). If the background of this field is in red then there is a problem with the resolving of the hostname.

Display name: Free-text field. Text entered here appears in the Distributed Computing window.

Platform: Select platform type, the options are: Linux (64 bits), Windows (32 bits), Windows (64 bits) and Mac OS X.

Survey data root: Location of the survey (the path to the survey data root folder from the host machine)

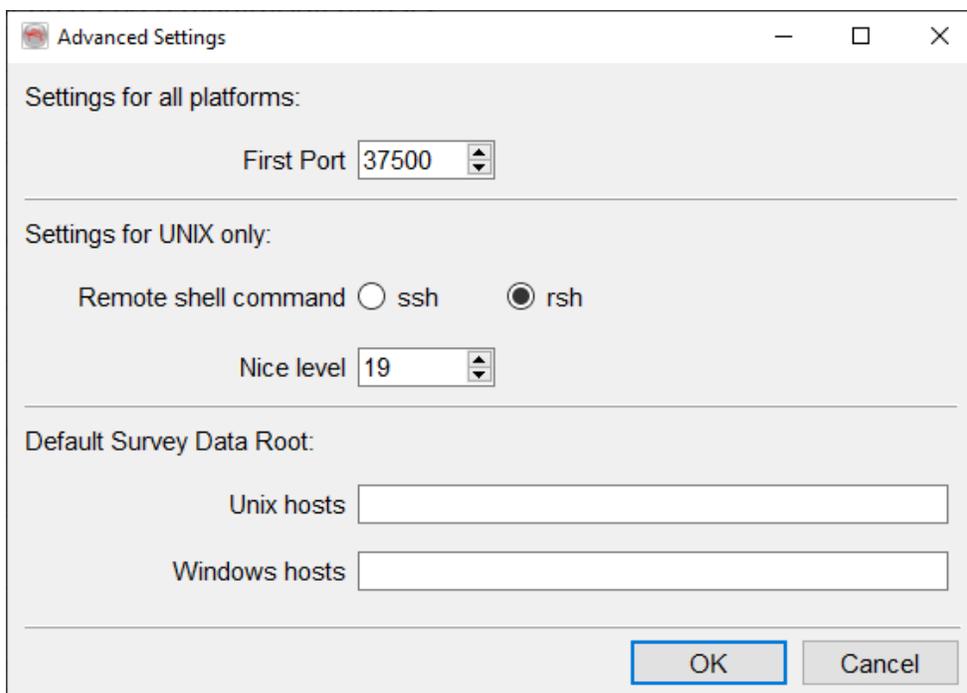


Figure: Advanced Settings window

Advanced Settings:

- Here you may change the first port value (in the case that it is blocked or in use). By default this first TCP port is 37500. We advise to open up to 5 ports, e.g. 37500-37504.
- Linux users may decide to change the remote shell command from the default ssh to rsh.

When setting to ssh it is required that the user who is running OpendTect is able to login to the other nodes without a password. This can be done by setting up public key authentication between the nodes. We will not go into detail of how to do this. In short this is done by generating the SSH key on the machine you are using to start the jobs,

the public key then needs to be uploaded to the nodes and added to the user's `.ssh/authorized_keys` file.

- The Nice level sets the priority on the host machines, 19 being nicest and 1 being least nice).
- Finally, the Default Data Root can be set per platform.

Description of icons:

 **Add** new host.

 **Remove** selected host.

 **Move** host up or down.

 **Test hosts.** Will perform tests to ensure that the server and nodes can communicate to the necessary extent to perform the MMP. (ie: can the nodes find the data root folder and read/write into it)

Tips & tricks

In case you are having issues with the Distributed Computing between the server that launches the scheduler and the nodes please check the following:

- Can you ssh into the node(s) without having to type a password? e.g. run: `ssh nodehostname ls -la`
- Can you the access the data project on the node(s)? e.g. run: `ssh nodehostname ls -la /pathto/dataproject`
- Can you access OpendTect on the node(s)? e.g. run: `ssh nodehostname ls -la /pathto/opendtect`
- When you run the scheduler is it listening on the port you set in the BatchHosts file? e.g. run: `netstat -antpu | grep 37500`

- Is there a time difference on the server that is starting the scheduler and node(s)? It is a good idea to sync them via a ntp server.
- Is the hostname resolution correct? Do the nodes get the correct IP address to which they need to connect?

For more information on this topic, please refer to [OpenTect's Youtube Channel](#) where you may find the webinar: [Multi-Machine Processing Setup](#).

Setup the preferred GPU with NVIDIA Control Panel

Laptops with an Intel CPU usually come with two integrated graphic cards. The default one the laptop starts up with would be one such as Intel HD graphics. OpenTect however will have the best performance when using the Nvidia GPU. Therefore that one should be used instead.

When OpenTect starts it will scan your graphics card. If it finds an Intel card there will be a warning message: 'Intel card found. If your computer has multiple graphics cards, consider switching from the integrated graphics.'

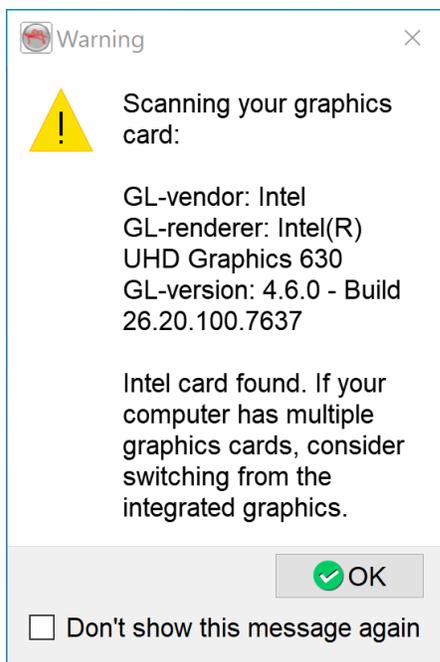


Figure: Warning message about the Intel card

On Windows 8.1 and 10 up to feature release 2004 the Nvidia GPU can be made the preferred GPU with the NVIDIA Control Panel. We assume that you have installed the latest Nvidia driver that is available for your GPU. If not then please go to the [Nvidia website](#) and download and install the latest driver for your type of GPU.

In case you are running Windows 10 feature release 20H2 or newer the preferred GPU has to be [set with the Windows Settings](#).

Please do the following:

1. Search for NVIDIA Control Panel, then start it

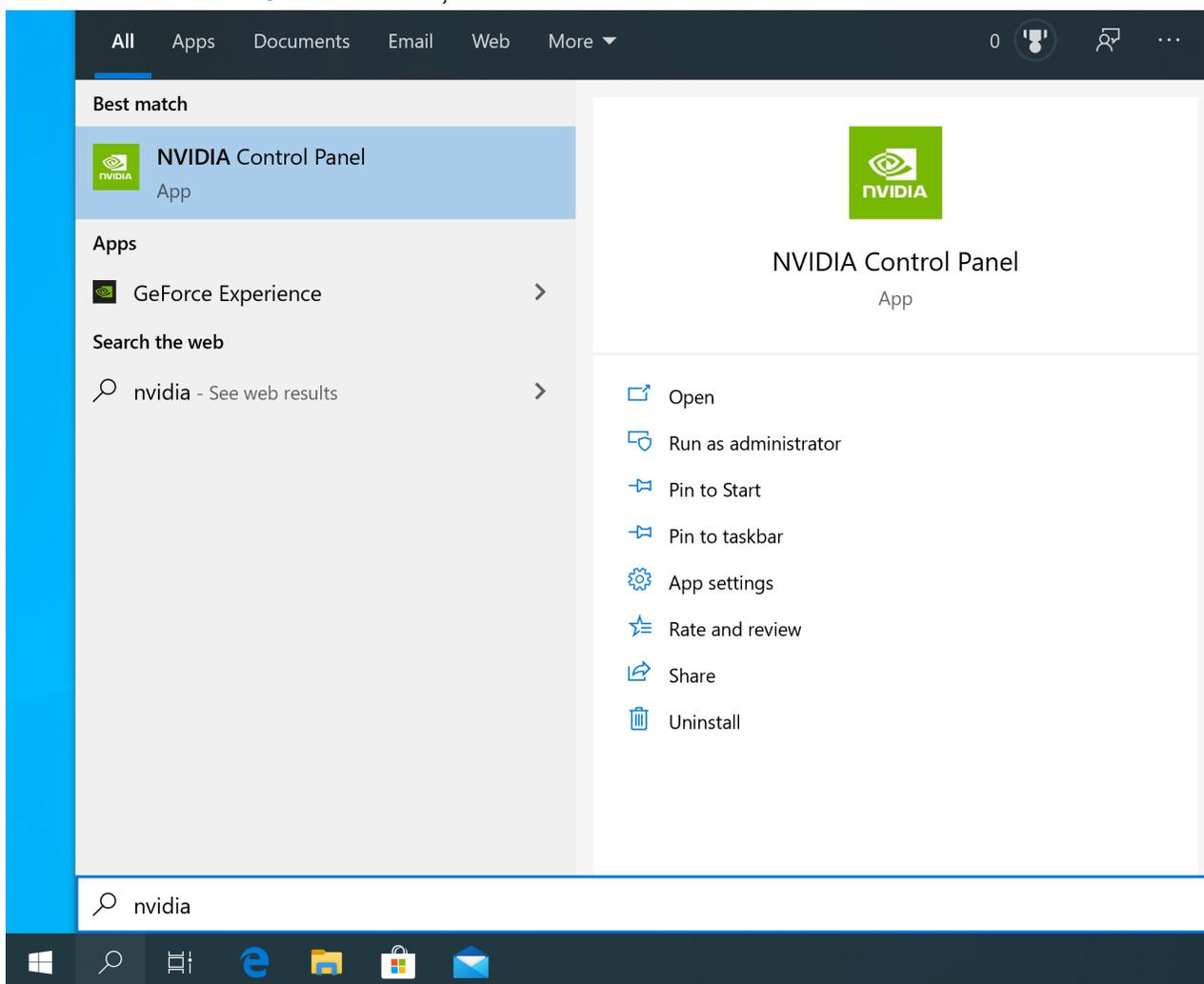


Figure: Starting NVIDIA Control Panel on Windows 10

2. In the toolbar on the left choose Manage 3D settings

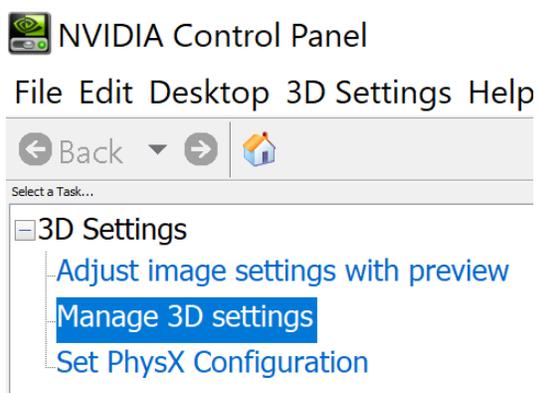


Figure: NVIDIA Control Panel: 3D Settings > Manage 3D settings

3. At Tab Program Settings click the Add button

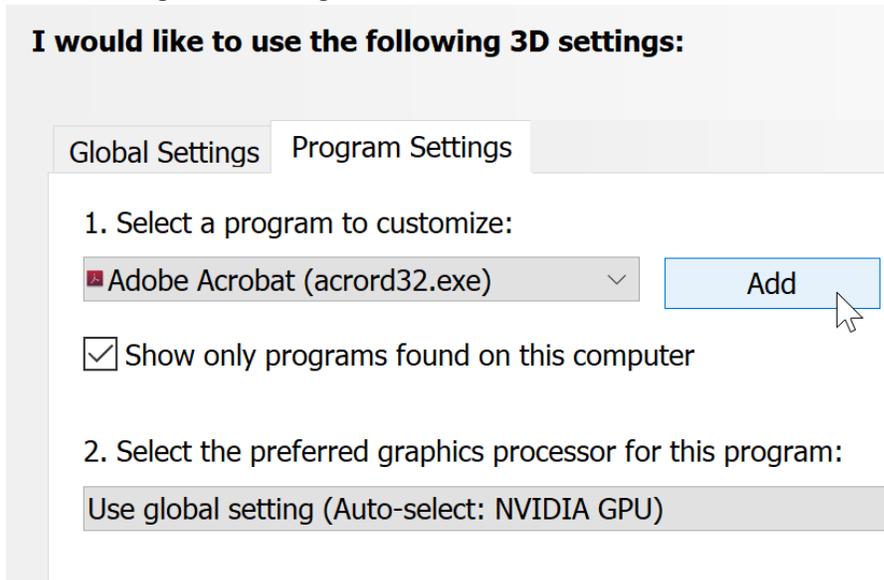


Figure: Add a program for customization of the 3D settings

4. In the new window it should show you OpendTect as a recently used program. Select it and click on Add Selected Program.

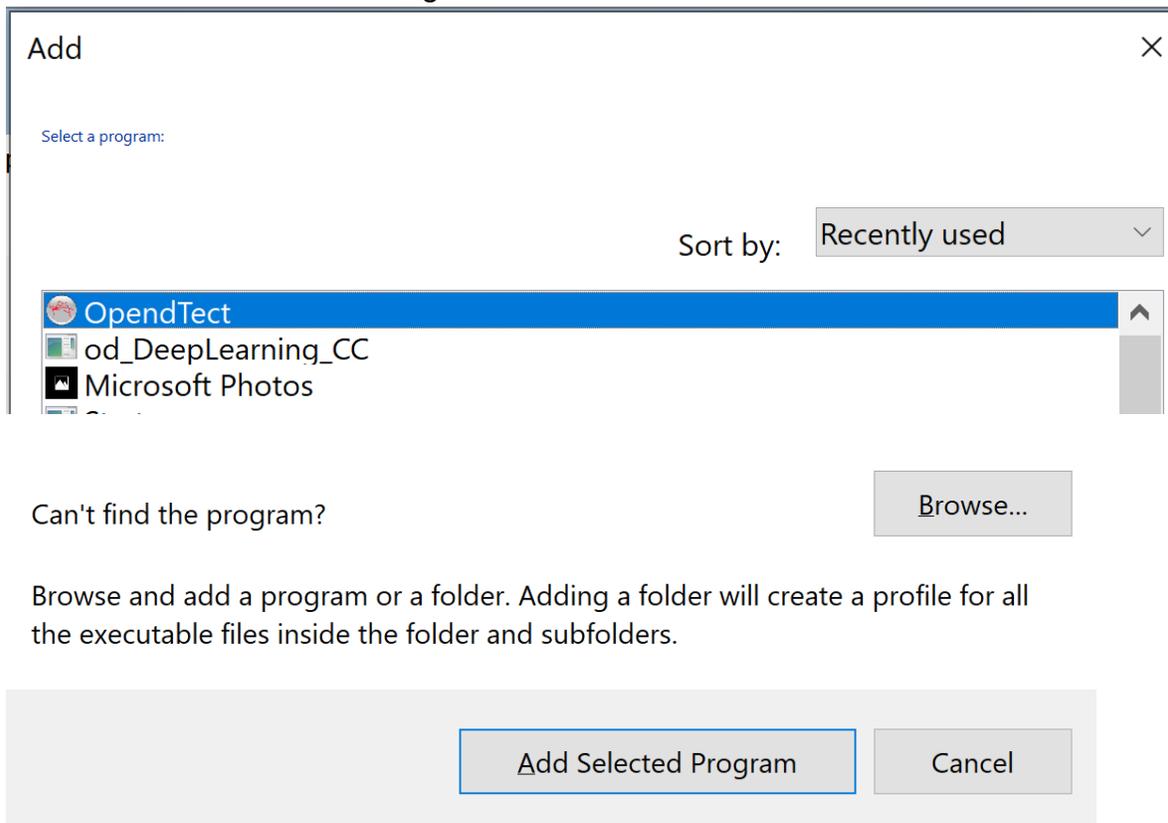


Figure: List of recently used programs in NVIDIA Control Panel, add OpendTect

5. Select the preferred graphics processor: High-performance NVIDIA processor.

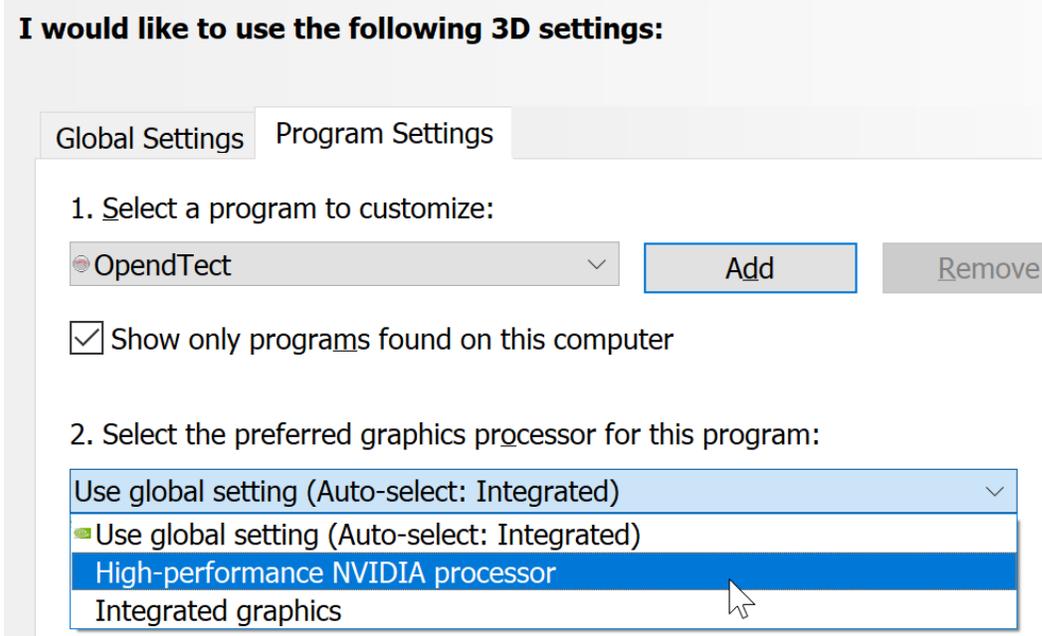
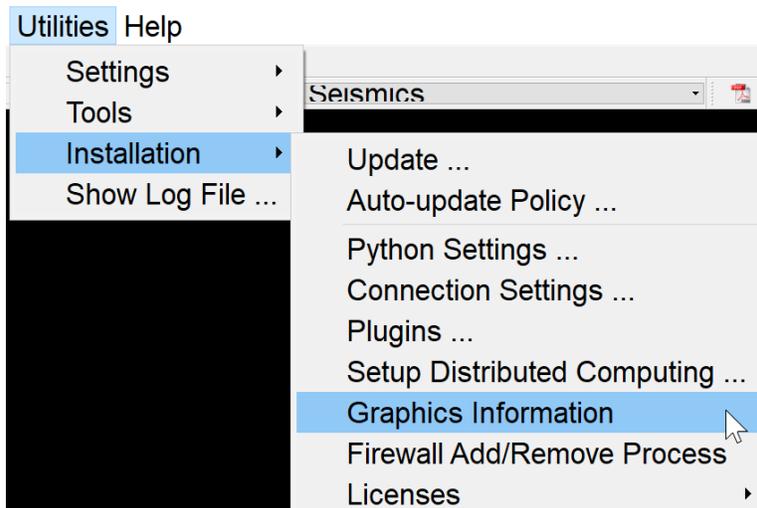


Figure: Customize OpenText to use the NVIDIA GPU

6. Click Apply to confirm the OpenText program customization.
7. The next time you start OpenText it should no longer show the Intel card found message popup. To check the graphics card used in OpenText you might want to check *Utilities > Installation > Graphics Information*. It should now show the Nvidia card information and driver version.



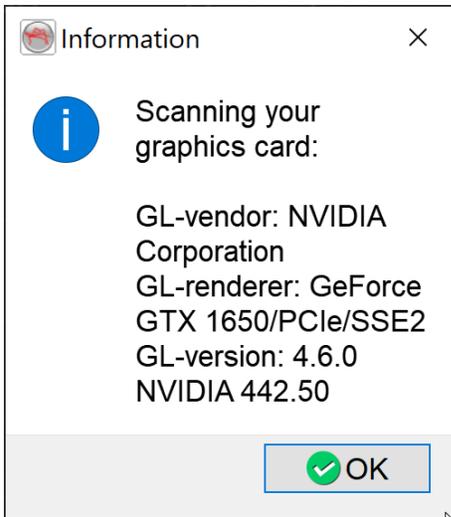


Figure: Check which graphics card OpendTect uses

Setup the preferred GPU with Windows settings

Laptops with an Intel CPU usually come with two integrated graphic cards. The default one the laptop starts up with would be one such as Intel HD graphics. OpenTect however will have the best performance when using the Nvidia GPU. Therefore that one should be used instead.

When OpenTect starts it will scan your graphics card. If it finds an Intel card there will be a warning message: 'Intel card found. If your computer has multiple graphics cards, consider switching from the integrated graphics.'

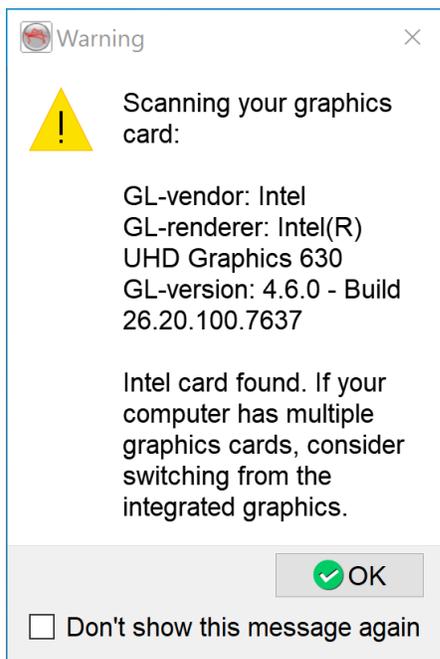


Figure: Warning message about the Intel card

On Windows 10 feature release 20H2 and newer the Nvidia GPU can be made the preferred GPU with the Windows settings. We assume that you have installed the latest Nvidia driver that is available for your GPU. If not then please go to the [Nvidia website](#) and download and install the latest driver for your type of GPU.

In case you are running Windows 8.1 or 10 feature release 2004 or older the preferred GPU has to be [set with the NVIDIA Control Panel](#).

Please do the following:

1. Click the Start icon and then go to Settings > System

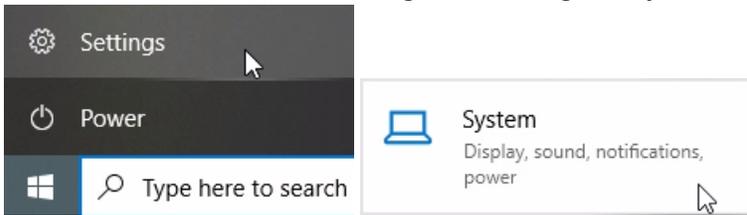


Figure: Navigating to the Windows system settings

2. Windows settings should open the display settings. Browse down to Graphics settings and click it.

[Advanced display settings](#)

[Graphics settings](#)

Figure: Display > Graphics settings

3. Now choose an app to set the preference. OpendTect is a desktop app, so make sure that the drop down bar has "Desktop app" selected. Now click the "Browse" button and navigate to the OpendTect installation directory, e.g.: C:\Program Files\6.6.0\bin\win64\Release and add od_main.exe

Choose an app to set preference

Desktop app

Browse

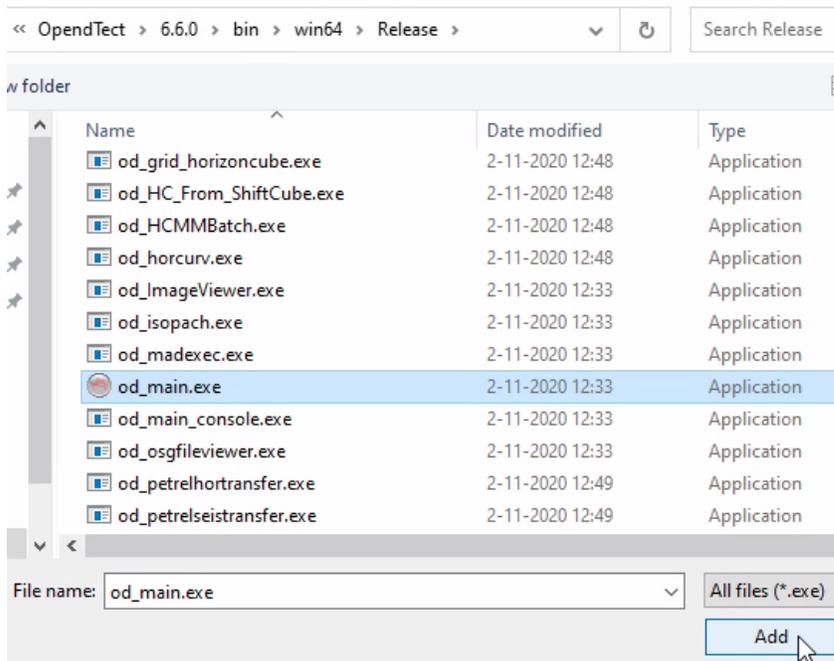


Figure: Choosing od_main.exe to set the GPU preferences

4. Now click the "Options" button and choose the High performance option and clicking the "Save" button for setting the Nvidia GPU as default for OpendTect.

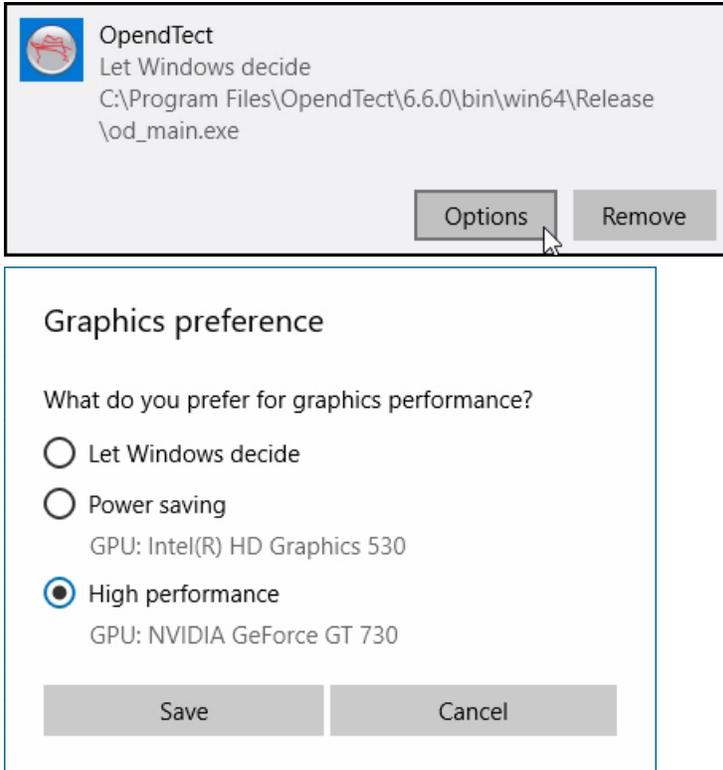
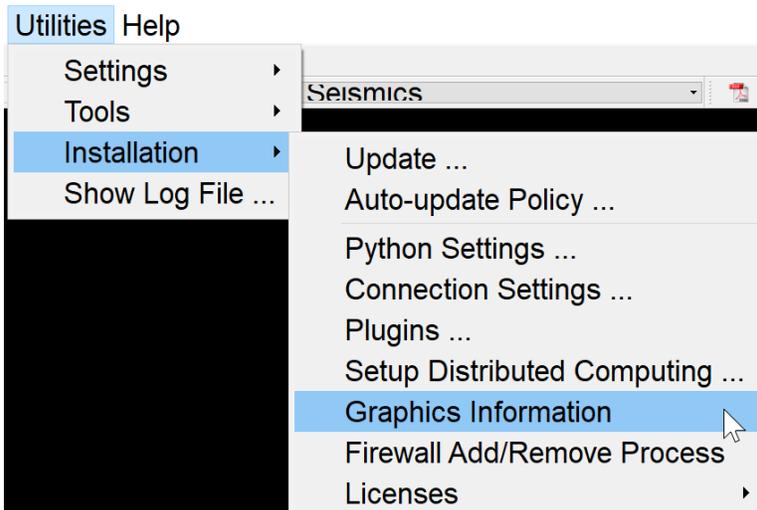


Figure: Setting the Nvidia card as high performance graphics preference for OpendTect

5. The next time you start OpendTect it should no longer show the Intel card found message popup. To check the graphics card used in OpendTect you might want to check *Utilities > Installation > Graphics Information*. It should now show the Nvidia card information and driver version.



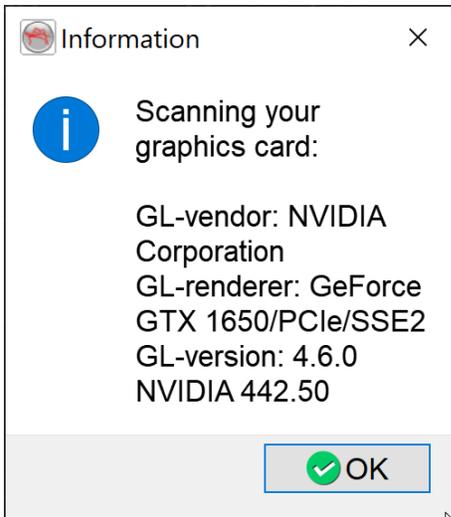


Figure: Check which graphics card OpendTect uses

Setup NVIDIA GPU Performance Counters

When running jobs with the Machine Learning plugin you might get the following error message: CUPTI_ERROR_INSUFFICIENT_PRIVILEGES

To fix this please follow the [instructions](#) from the NVIDIA site. It will be a good idea to do this anyway to prevent of getting the above mentioned error message.

In short the way to set this on Windows is:

1. Open the NVIDIA Control Panel
2. From the menu select *Desktop* and ensure that *Enable Developer Settings* is checked
3. From the tray select *Developer > Manage GPU Performance Counters*
4. Select *Allow access to the GPU performance counter to all users* to enable unrestricted profiling

Environment variables

There are several environment variables that can be set. OpendTect will look for these environment variables. They can be set to:

- let OpendTect use the dGB license from the FlexNet license server,
- get special features working,
- or to use custom paths for certain functionality.

Environment variable	Expected Values	Description
DGBLD_LICENSE_FILE	port@host f.i. 27000@myserver	Used on clients to use a dGB FlexNet server license. - port = the TCP/IP port the license server is using - host = the hostname of the license server
	@host f.i. @myserver	Used on clients to use a dGB FlexNet server license. - host = the hostname of the license server - port = not specified. A default TCP/IP port number in the range of 27000-27009 is used.
	path to license file	Used on clients to use a node-locked or floating license. - path to license file = exactly that. Place the license file in a network location available to all machines running OpendTect or copy it to all of the machines running OpendTect (Typically: C:\Licenses)
DTECT_BATCH_HOSTS_FILEPATH	Path to a BatchHosts file	This sets the path to a tailor-made BatchHosts file. If it is not set then OpendTect will use the default BatchHosts file in the OpendTect installation directory, e.g. C:\Program Files\OpendTect\6.6.0\data\BatchHosts

DTECT_CLUSTER_ PROC	yes or no	When set to yes OpendTect will have cluster processing enabled.
DTECT_PETREL_ PORT	Value of preferred port for the OpendTect Petrel* plugin access.	This sets the preferred port for the OpendTect Petrel* plugin access.
DTECT_SETTINGS	Path to the OpendTect settings home directory	This sets the path to the OpendTect settings home directory. Use this if you do not want to use the .od directory in the user's home folder. Use this environment variable for Linux and Mac.
DTECT_ WINSETTINGS	Path to the OpendTect settings home directory	This sets the path to the OpendTect settings home directory. Use this if you do not want to use the .od directory in the user's home folder. Use this environment variable for Windows.
LM_LICENSE_FILE	port@host or @host or path to the license file	Although it works it is preferred to set DGBLD_LICENSE_FILE instead.
OD_USE_ VIRTUALMEM	yes or no	When set to yes OpendTect will support using virtual memory.
RSFROOT	Path to the installation directory of Madagascar	This sets the path to the Madagascar installation directory. If it is not set then and one would try to start Madagascar from OpendTect an error message would be displayed.

* *Petrel is a mark of Schlumberger*

Firewall Rules and Open Ports

Firewall rules For OpendTect 6.6 (Windows)

The installer will ask to add firewall rules to the Windows Firewall. Which .exe programs it adds to Windows Firewall depends on the packages that have been installed. Here is the full list of .exe files that require a rule in the Inbound Rules in the Windows Firewall. You may also do this manually. The default OpendTect base directory is C:\Program Files\OpendTect. The base directory can be another directory if that is chosen during installation. If that is the case then replace the default base directory in below list with the actual base directory.

- C:\Program Files\OpendTect\6.6.0\bin\win64\Release\od_start_dtect.exe
- C:\Program Files\OpendTect\6.6.0\bin\win64\Release\od_main.exe
- C:\Program Files\OpendTect\6.6.0\bin\win64\Release\od_SeisMMBatch.exe
- C:\Program Files\OpendTect\6.6.0\bin\win64\Release\od_remoteservice.exe
- C:\Program Files\OpendTect\6.6.0\v7\bin\win64\Release\od_LogPlot.exe
- C:\Program Files\OpendTect\6.6.0\v7\bin\win64\Release\od_DeepLearning_CC.exe
- C:\Program Files\OpendTect\6.6.0\v7\bin\win64\Release\od_DeepLearning_EM.exe
- C:\Program Files\OpendTect\6.6.0\v7\bin\win64\Release\od_DeepLearning_TM.exe
- C:\Program Files\OpendTect\6.6.0\v7\bin\win64\Release\od_DeepLearning_ModelImport.exe
- C:\Program Files\OpendTect\Python\python.exe
- C:\Program Files\OpendTect\Python\envs\odmlpython-cpu-mkl\python.exe
- C:\Program Files\OpendTect\Python\envs\odmlpython-cuda10\python.exe

Open Ports

OpendTect will need to open TCP ports on the localhost for Batch Processing and Machine Learning. Here is a list of ports that OpendTect needs to open if you run a certain job:

Ports	Description
5050	Is used by executable od_remoteservice, which needs to be started on the compute node before job is submitted to it, but only when the job is started from Windows.
20050-20080	Starting from OpendTect 6.6 these ports are used for communication between OpendTect and Python, starting at port 20050. We advise to have a broad range to be open, e.g. until 20080.

37500-37504 Ports used for the Distributed Computing, starting at port 37500. We advise to have a port range size of 5 open. That allows a user to start several Distributed Computing jobs from the same client machine.

57375 Is used for the OpendTect Petrel* plugin access.

** Petrel is a mark of Schlumberger*

Python environments

Information about the OpendTect Python environments

Miniconda3 is a basic Python environment. It allows using the Presentation Maker plugin without having to install Python yourself. Provides libraries, including:

- Python 3.7
- pandas
- python-pptx

Math Kernel Library (MKL) is a complete Python environment for Machine Learning using CPU only (with Intel™ Math Kernel Library - MKL).

Provides libraries, including:

- Python 3.7
- tensorflow-mkl
- keras
- scikit-learn
- pandas
- bokeh
- numpy
- h5py
- joblib
- psutil
- spyder

- xgboost
- python-pptx

Python CUDA 10.1 is a complete Python environment for Machine Learning on the GPU with CUDA 10.1.

Requires minimum driver version 418.XX, for specifics see this [table on the Nvidia CUDA Toolkit documentation page](#).

Provides libraries, including:

- Python 3.7
- tensorflow-gpu
- keras
- scikit-learn
- pandas
- bokeh
- numpy
- h5py
- joblib
- psutil
- spyder
- xgboost
- python-pptx

Installing Python environments with the OpendTect installer

When running the OpendTect installer there is the option to install the Python environments. The installer will auto select the Miniconda3 environment and also auto select the Math Kernel Library (MKL) and CUDA 10 when selecting the Machine Learning plugin.

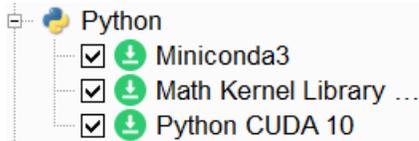


Figure: Install Python environments with the OpendTect installer

If you have your own miniconda3 or anaconda3 environment you might want to consider to [install the OpendTect Python environments from YAML files](#) instead.

Installing Python environments from YAML files

The Python environments we provide can be installed into:

- the Python miniconda3 base directory that is installed with the OpendTect installer,
- your own miniconda3 installation or into
- your own anaconda3 installation.

Download the OpendTect Python YAML files

For Linux:

- [odmlpython-cpu-mkl_lux64.yml](#)
- [odmlpython-cuda10_lux64.yml](#)

For Windows:

- [odmlpython-cpu-mkl_win64.yml](#)
- [odmlpython-cuda10_win64.yml](#)

The YAML files contain a list of essential packages that are needed inside the Python environment. You can choose to install either the cpu-mkl or cuda10 one or both. The cpu-mkl one is for running the Machine Learning jobs on the CPU only. In most cases the Cuda10 one would be the best option so that the Machine Learning jobs can be run on the Nvidia GPU.

Install the environments

On Linux:

1. Open a terminal/konsole.
2. Then run the following:

- a. `cd conda installation directory (e.g. cd /home/user/Miniconda3)`
- b. `cd bin`
- c. `bash ; source activate`
- d. `conda env create --file /path/to/odmlpython-cpu-mkl_lux64.yml`
- e. `conda env create --file /path/to/odmlpython-cuda10_lux64.yml`
- f. Conda will collect the packages it needs to install from the YAML file.
- g. Then it will download and extract the packages, and do the transaction.
- h. To make the Python environments completely operational with OpendTect 6.6's Machine Learning plugin it is necessary to replace a `callbacks.py` of keras on the system.

On Windows:

1. Open the Command Prompt. Run as administrator if the Python installation directory is on a location for which the user requires administrative rights.
2. Then run the following:
 - a. `cd conda installation directory (e.g. cd C:\Users\user\Miniconda3)`
 - b. `cd condabin`
 - c. `activate.bat`
 - d. `conda env create --file C:\path/to/odmlpython-cpu-mkl_win64.yml`
 - e. `conda env create --file C:\path/to/odmlpython-cuda10_win64.yml`

```
Administrator: Command Prompt - activate.bat - "c:\Users\arjan.DGBES\Anaconda3\condabin\conda.bat" env create --file C:\Users\...
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\windows\system32>cd c:\Users\arjan.DGBES\Anaconda3
c:\Users\arjan.DGBES\Anaconda3>cd condabin
c:\Users\arjan.DGBES\Anaconda3\condabin>activate.bat
c:\Users\arjan.DGBES\Anaconda3\condabin>conda.bat activate

(base) c:\Users\arjan.DGBES\Anaconda3\condabin>conda env create --file C:\Users\arjan.DGBES\Downloads\odmlpython-cpu-mkl-win64.yml
Collecting package metadata (repodata.json): \
```

Figure: Install Python environments from the YAML file on Windows

- f. Conda will collect the packages it needs to install from the YAML file.
- g. Then it will download and extract the packages, do the transaction and install xgboost with pip.
- h. To make the Python environments completely operational with OpendTect 6.6's Machine Learning plugin it is necessary to replace a callbacks.py of keras and to get the cupti library on the system. The cupti library is only needed for the cuda10 environment.

Fixes for callbacks.py and cupti

callbacks.py

For Linux:

- Download [callbacks.py](#).
- It needs to be copied into `odmlpython-cpu-mkl/lib/python3.7/site-packages/tensorflow_core/python/keras` **and/or** `odmlpython-cuda10/lib/python3.7/site-packages/tensorflow_core/python/keras` **so that it overwrites the existing callbacks.py file.**

For Windows:

- Download [callbacks.py](#).
- It needs to be copied into `odmlpython-cpu-mkl\Lib\site-packages\tensorflow_core\python\keras` and/or `odmlpython-cuda10\Lib\site-packages\tensorflow_core\python\keras` so that it overwrites the existing `callbacks.py` file.

cuPTI libraries (Windows only)

There are two ways to apply the fix:

1. You can download [cuPTI.lib](#) and [cuPTI64_101.dll](#). Then copy them into `odmlpython-cuda10\Library\bin`.
2. Or you can download and install [CUDA Toolkit 10.1 update 2 from the Nvidia CUDA site](#). After installation add `C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.1\extras\CUPTI\lib64` to the System environment variable Path.

For information about how to use the custom environment in OpenTect 6.6 please see the chapter [Setup Python Settings](#).

License server

Installing a license server on Linux

The FlexNet vendor package with the binaries that are needed can be downloaded from the [FlexNet Vendor Packages download page](#).

1. Download the [Linux 64 \(FlexNet v11.18.1.0\) zip-file](#) for OpendTect Pro & dGB Plugins for OpendTect 6.4 and newer.
2. Create a directory on the server that is to be used as license server for OpendTect.
E.g. `mkdir /opt/dgb_licserver`
3. Unzip the zip-file in the directory that has been created in previous step.
4. The zip-file has the following files:
dgblid : The dGB vendor daemon
lmgrd : The FlexNet license manager daemon
lmutil : The FlexNet license management utilities
5. The next thing that is needed is the server license file. This license file will be provided by the dGB support via support@dgbes.com.
Best is to place the license file in the same directory.
6. Now the license file needs to be edited. Hence, open the license file with a text editor like Vim.
7. In the top of the license file there is a line with text *DAEMON dgblid full_path_to/dgblid* .
Replace the fullpath_to/dgblid with the path to the dgblid file.
In this example we will change it into: *DAEMON dgblid /opt/dgb_licserver/dgblid* .
Save the license file.

Starting a license server

The license server can be started by executing commands:

```
cd /opt/dgb_licserver
./lmgrd -c licensefile.lic -l debug.log
```

Please refer to the debug.log file in case of issues.

To see the license server status you may run:
./lmutil lmstat - a
You will then see the port number the server uses, the path to the license file(s) that are in use, whether the license server is up or down, how many licenses for the features are issued and in use.

Error messages

You may encounter error messages when starting the license server. Listed here are the most common ones and how to solve them.

- ./lmgrd: No such file or directory

This error occurs when either the file or the executable's loader is not present. To find the executable's loader run the following command:
file lmgrd
It should show something like:
lmgrd: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-lsb-x86-64.so.3, for GNU/Linux 2.6.18, stripped
Now check whether the file */lib64/ld-lsb-x86-64.so.3* exists. If not then lsb needs to be installed.

How to install lsb? This differs per distro:

- Debian/Ubuntu: `sudo apt-get install lsb`
 - RHEL/CentOS: `yum install redhat-lsb`
 - OpenSUSE: `zypper install lsb`
- Cannot open "debug.log" as log file (- 1)

This error message occurs if the user has not enough permissions to create the log file in the path specified. To solve this error message either become a user that has permissions to create the log file in the specified path or create it in a path where the user has sufficient permissions, e.g. in the user's home directory. Debian/Ubuntu users may want to run `sudo ./lmgrd -c license-file.lic -l debug.log`

- license manager: can't initialize: Cannot read license file.

This error occurs if the user who starts the license server has no permissions to read the license file. To solve this change the owner of the license file.

- Can't make directory /usr/tmp/.flexlm, errno: 2 (No such file or directory)

This error message occurs if there is no /usr/tmp folder on the system. It can be solved by creating a symbolic link /usr/tmp that points to /tmp, for example run as root user:

```
ln -s /tmp /usr/tmp
```

- Error getting status: Cannot find license file (-1,73:2 "No such file or directory")

This error message occurs when you run `./lmutil lmstat -a`, but there is no license manager daemon running. It could be that the license manager daemon failed to start. More information can be found in the `debug.log` file.

Client installation

Please see the [installing floating licenses](#) chapter for more information on how to instruct the client on how to borrow the license from the server.

Installing a license server on Windows

The FlexNet vendor package with the binaries that are needed can be downloaded from the [FlexNet Vendor Packages download page](#).

1. Download the [Windows 64 \(FlexNet v11.18.1.0\) zip-file](#) for OpendTect Pro & dGB Plugins for OpendTect 6.4 and newer.
2. Create a directory on the server that is to be used as license server for OpendTect. E.g. create the directory `C:\dgb_licserver`.
3. Unzip the zip-file in the directory that has been created in previous step.
4. The zip-file has the following files:
 - dgbl.exe** : The dGB vendor daemon
 - lmgd.exe** : The FlexNet license manager daemon
 - lmtools.exe** : The FlexNet LMTOOLS utility
 - lmutil.exe** : The FlexNet license management utilities
5. Create in the directory a new text document and name it *debug.log*.
6. The next thing that is needed is the server license file. This license file will be provided by the dGB support via support@dgbes.com. Best is to place the license file in the same directory.
7. Now the license file needs to be edited. Hence, open the license file with a text editor like Notepad or Vim.
8. In the top of the license file there is a line with text *DAEMON dgbl full_path_to/dgbl* . Replace *fullpath_to/dgbl* with the path to the dgbl.exe file. In this example we will change it into: *DAEMON dgbl C:\dgb_licserver\dgbl.exe* . In case there are spaces in the path then the path should be within double-quotes. Save the license file.

9. Double-click the *lmtools.exe* file. This opens the LMTOOLS utility.

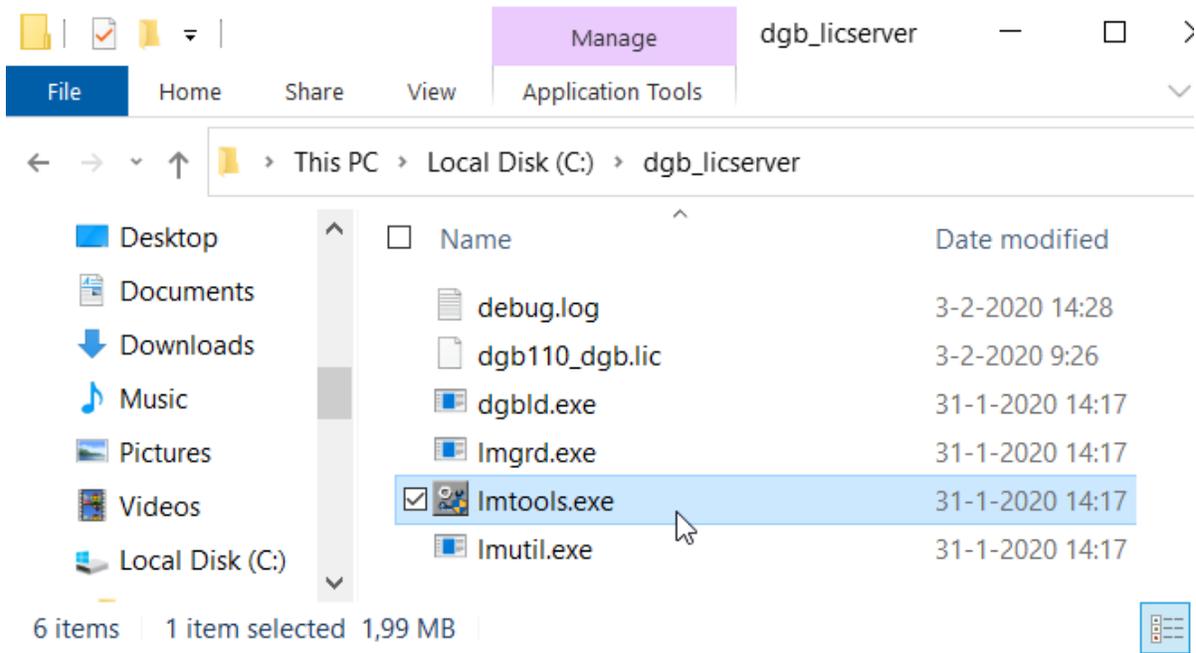


Figure: contents of the dgb_licserver directory

10. On *Service/License File* tab toggle on Configuration using Services and LMTOOLS ignores license file path environment variables.

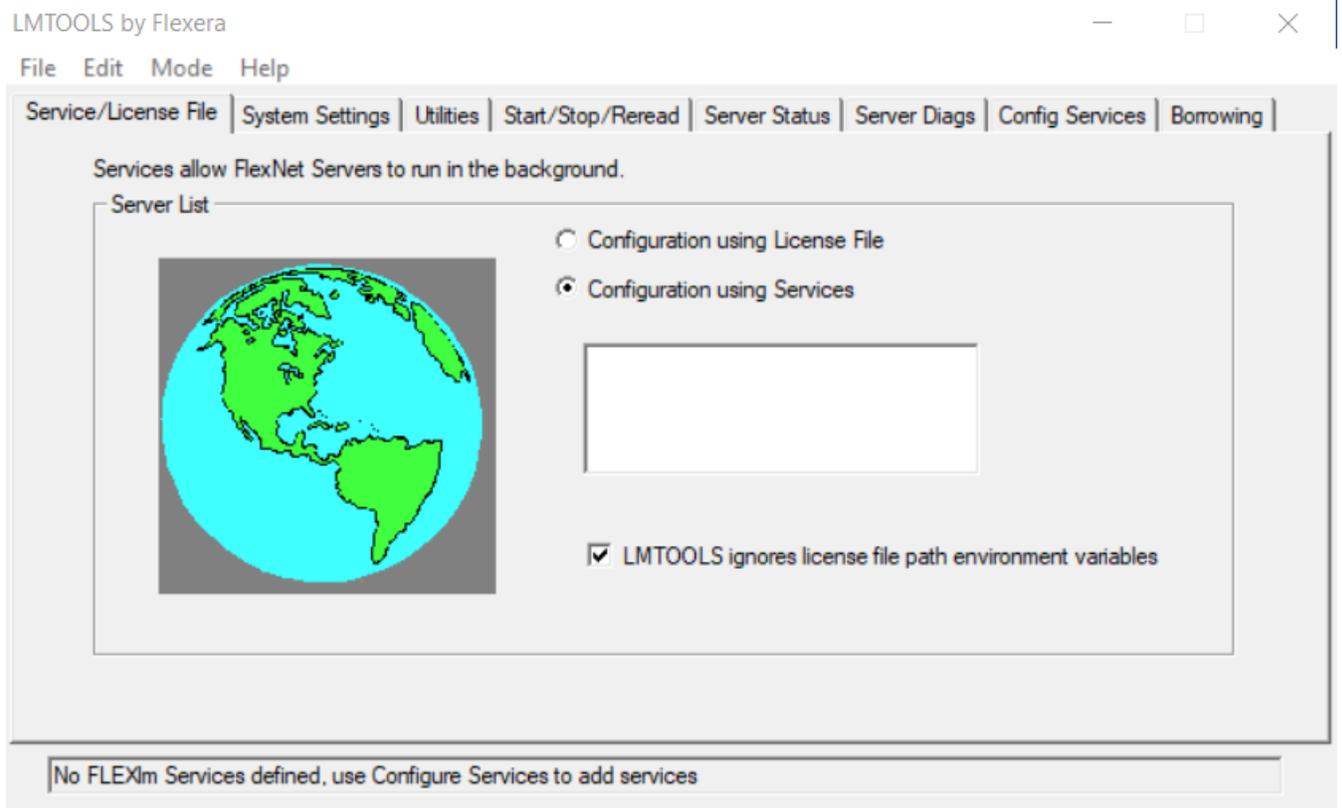


Figure: LMTOOLS utility: Service/License File tab

11. Click the Config Services tab.

1. At *Service Name* drop down create a new service name that will manage your OpendTect Pro & dGB plugins licenses. The default service name is *Flexlm Service 1*. Best is to change this into a name like dGB License Service or OpendTect Pro License Service.
2. At *Path to the Imgrd.exe file* browse to the directory that contains the Imgrd.exe file. Select Imgrd.exe and then click button *Open*.
3. At *Path to the license file* browse to the directory that contains the license file. Select the license file and then click button *Open*.
4. At *Path to the debug log file* browse to the directory in which you created the debug.-log file. Select the log file and then click button *Open*.
5. Toggle on *Use Services* and then toggle on *Start Server at Power Up*. This will make sure that the license server will start automatically at power up or reboot.

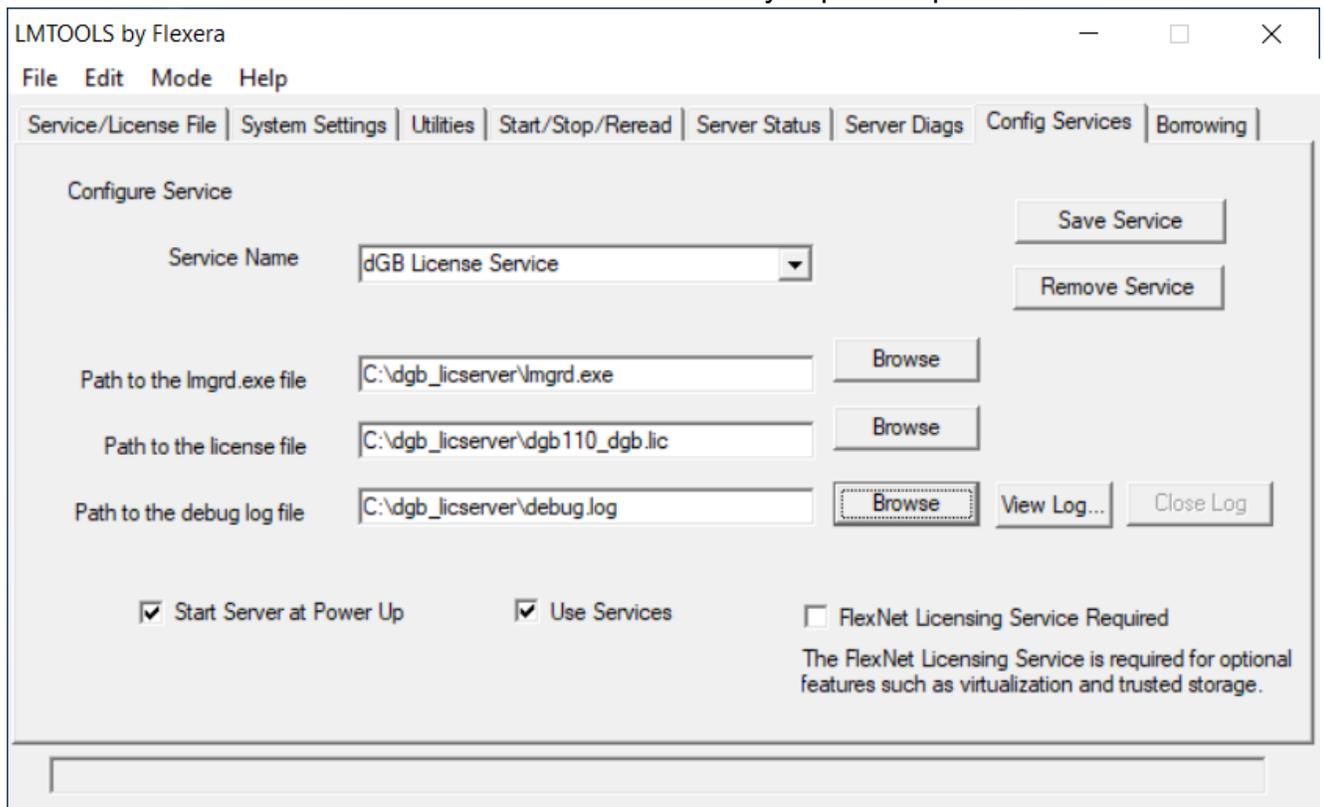


Figure: LMTOOLS utility: Config Services tab

6. Click button Save Service and Yes in the confirmation window.

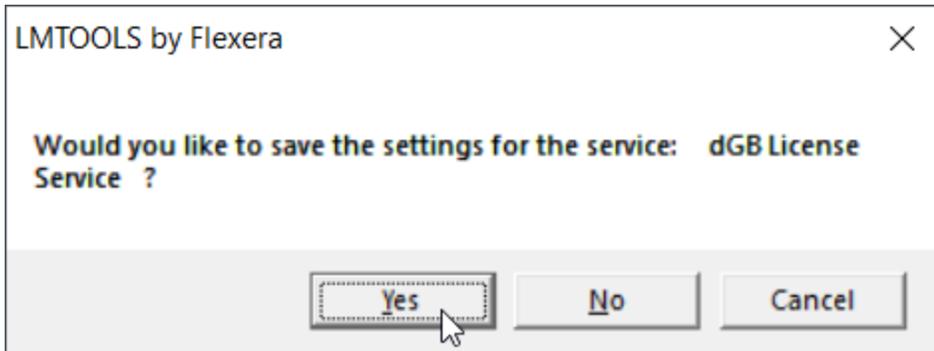


Figure: Save the service

7. LMTTOOLS will show a warning window that the preferred path <SystemDrive>\ProgramData to store service data is not set. You get this because debug.log is not configured in the ProgramData folder. You can ignore this as we should have enough permissions to read from and write to the C:\dgb_licserver directory.

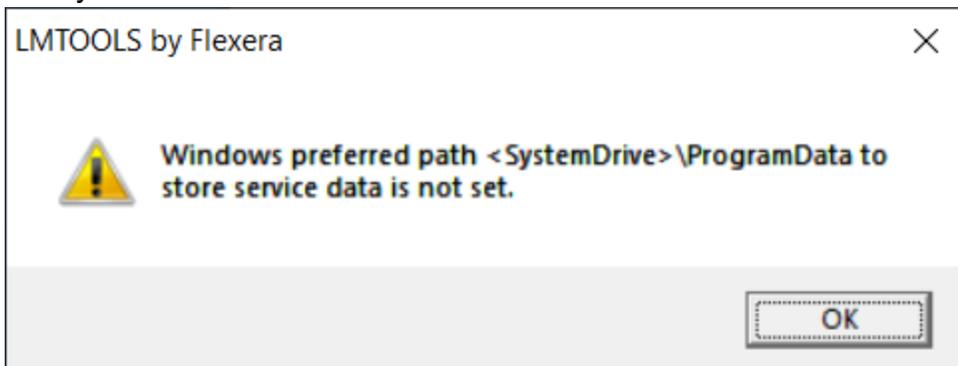


Figure: Warning about Windows preferred path for service data

Starting a license server

1. Start LMTTOOLS by double-clicking the *lmtools.exe* file.
2. Click the *Start/Stop/Reread* tab.
3. Make sure that the correct license server is selected in the list. Then click button *Start Server*.

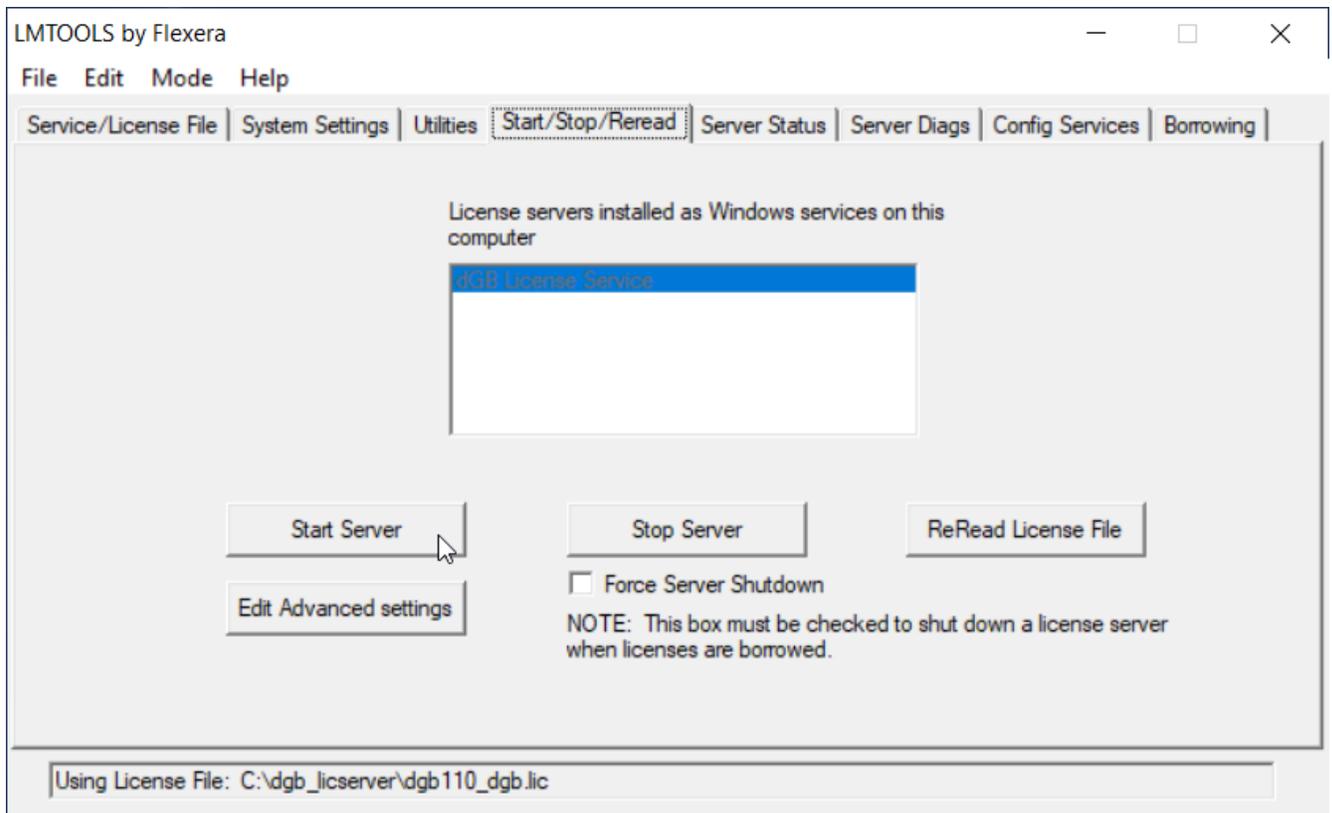


Figure: Starting dGB License Server

4. It is possible that the Windows firewall or your virus scanner blocks dgbl.exe . If this is the case then you would want to add dgbl.exe to the list of Trusted Programs and/or Firewall Rules. After making this change click button *Start Server* again.
5. If the license server start is successful LMTOOLS will show the message: *Server Start Successful*.
6. Please refer to the debug.log file in case of issues.

This is an indication that the dgbl vendor daemon is not able to start:

```
(lmgrd) License server manager (lmgrd) startup failed:
(lmgrd) CreateProcess error code: 0x5 File= C:\dgb_lic-
server\dgbl.exe
License server manager (lmgrd) startup failed:
(lmgrd) CreateProcess error code: 0x5 File= dgbl.exe
```

Client installation

Please see the [installing floating licenses](#) chapter for more information on how to instruct the client on how to borrow the license from the server.

Petrel* Plugin

Installing the PIP file

The plugin can be installed to Petrel* with the help of the PIP file in the usual way, where you have to use the Plugin manager tool from the menu: File > Options > Plugin manager. Press 'Install plugin' and select the downloaded PIP file. After installation restart Petrel* to activate the plugin.

In order to install the updated version of the plugin, please remove the old one from the Petrel* plugin manager. Then restart Petrel* and install the plugin as described above.

The PIP files can be downloaded from the [dGB's download page](#).

** Petrel is a mark of Schlumberger*

Choosing the preferred Communication Port (TCP/IP)

By default the plugin should use the TCP/IP port 57375; in case this port is not available / accessible it can be changed in two ways:

- From plugin user interface, which is available under the 'Seismic interpretation' tab in the Petrel* ribbon.

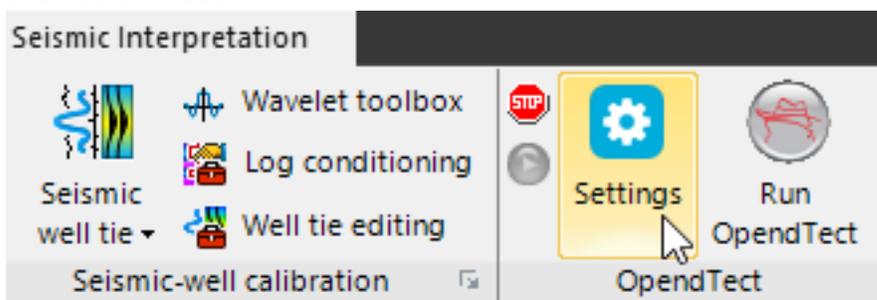


Figure: OpendTect Settings in ribbon

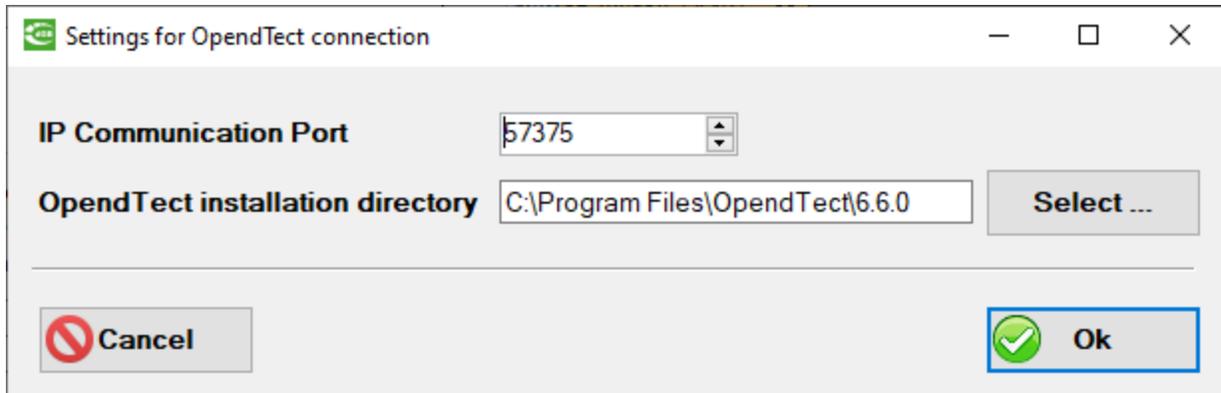


Figure: Settings for OpendTect connection

- By adding an environment variable `DTECT_PETREL_PORT`, and setting its value to the preferred port number which is available for access.

PetrelDirect user documentation

For information about PetrelDirect go to [the PetrelDirect chapter](#) of the OpendTect Pro and dGB Plugins Documentation.

Installation of PetrelDirect plugin in OpendTect and Petrel*

PetrelDirect plugin allows OpendTect Pro users to seamlessly exchange data with the Petrel* data store of a running Petrel* project. On the OpendTect side, PetrelDirect plugin is installed as a part of OpendTect Pro. On the Petrel* side, *Data access for OpendTect (dGB) plugin* must be installed. Currently supported Petrel* versions are Petrel 2018*, Petrel 2019* and Petrel 2020*. This is a step-by-step guide on how to install and configure PetrelDirect plugin in both OpendTect and Petrel*.

* *Petrel is a mark of Schlumberger*

OpendTect

1. OpendTect 6.6: PetrelDirect plugin is a part of OpendTect Pro:
2. Once OpendTect Pro is installed, PetrelDirect status button can be found in the lower right corner of the main OpendTect window:

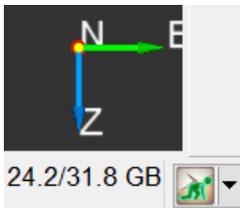


Figure: PetrelDirect status button

3. Click the button to check the Petrel connection settings. Make sure that the TCP port is the same port as is set in Petrel*.

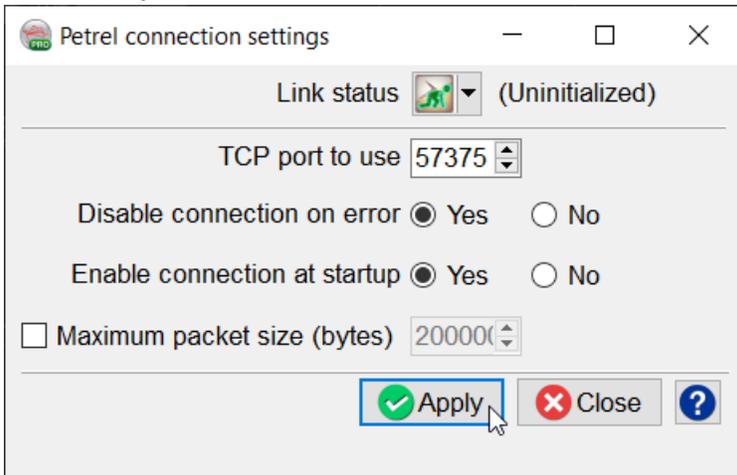


Figure: Petrel connection settings window

Petrel*

To be able to use PetrelDirect functionality in Opentect Pro, *Data access for Opentect (dGB) plugin* must be installed in Petrel*. Installation can be done either via Windows installer (MSI file) or Plugin Installer Package (PIP file).

Via Windows installer (MSI file):

The Windows installer does both the first-time plugin installation and an update of already installed plugin to a newer version without any extra actions.

1. Download msi file from the [dGB's download page](#).
2. Run the msi file and follow the instructions.
3. Start Petrel*

4. Go to *Seismic Interpretation* tab and observe that OpendTect toolbar is there now.

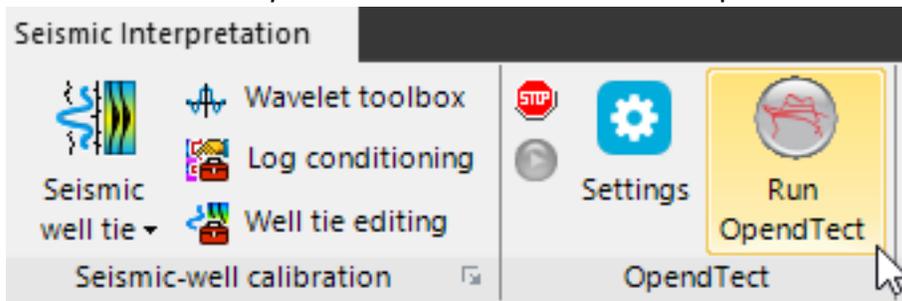


Figure: OpendTect toolbar

Via Plugin Installer Package (PIP file):

For the plugin to update to a newer version, an old version must be uninstalled first:

1. Start Petrel*
2. In Petrel* main window: go to File > Options > Plugin Manager ...
3. In Ocean Plugin Manager window: uninstall the old version of *Data access for OpendTect (dGB) plugin* by selecting it and clicking *Uninstall* button.
4. Close Petrel*

For the first-time installation or once the plugin version is uninstalled:

1. Download PIP file from the [dGB's download page](#).
2. Start Petrel*

3. In Petrel* main window: go to File > Options > Plugin Manager ...

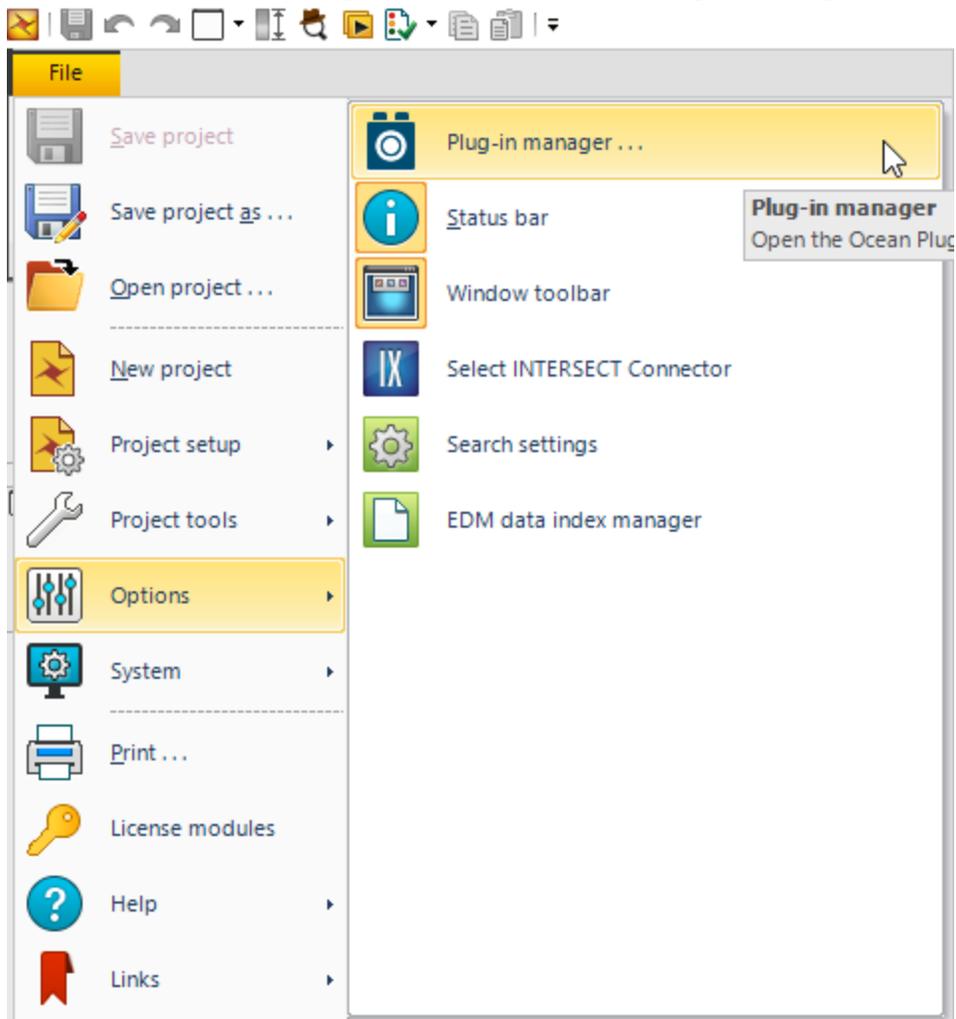


Figure: File > Options > Plugin Manager

4. In *Ocean Plugin Manager* window:

1. Click on *Install plugin* button
2. Locate PIP file and click *Open*
3. Once the installation is finished click *Close* in both windows

5. Restart Petrel*

6. Go to *Seismic Interpretation* tab and observe that OpendText toolbar is there now.

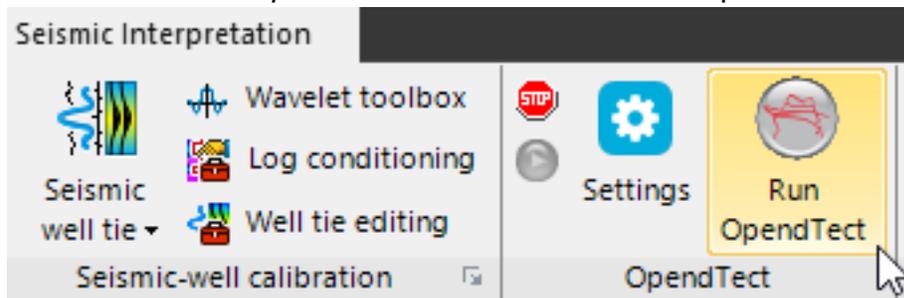


Figure: OpendText toolbar

** Petrel is a mark of Schlumberger*

Troubleshooting

When experiencing issues with OpendTect there are several ways to troubleshoot:

You can

- look in the log file(s) inside the *Survey Data Root\LogFiles* and *%TEMP%* folder;
- review, modify and/or delete the OpendTect settings in the *\$HOME/.od* folder on Linux or *%UserProfile%\od* on Windows;
- set the environment variable *DTECT_DEBUG* to *yes* to get more information from OpendTect in the terminal or the Command Prompt when starting OpendTect.
- set the environment variable *QT_DEBUG_PLUGINS* to *1* to get more information about Qt in the terminal or the Command Prompt when starting OpendTect (Linux and Windows) or for the OpendTect installer (Linux only).
- set the environment variable *OSG_NOTIFY_LEVEL* to *DEBUG* to get more information about OSG in the terminal or the Command Prompt when starting OpendTect.
- set the environment variable *OD_NOSCENE_AT_STARTUP* to *1* to start OpendTect without a scene. This is to determine whether there is a GPU driver issue.

You can also

- look for an answer on the [OpendTect Users Group](#) or use your favorite search engine;
- send an email to the OpendTect Users Group via users@opendtect.org. To post one needs to be a member. You can become a member by sending an email to users+subscribe@opendtect.org. For more information please read the [dgbes.com Users Group Q&A page](#). Perhaps someone in the group experienced the same issue and/or can tell you how to fix the issue;

- get priority support for commercial users by sending an email to the OpendTect Support: support@dgbes.com;
- choose to submit a crash report in the event of a malfunction of OpendTect. This crash report contains information that will help OpendTect's developers to locate and fix the issue. If you choose to identify yourself in the comments, the crash report will be added to your file by OpendTect developers, and OpendTect support may contact you for more information, or to inform you about the fixes of the issue.

The crash report is a text-file that contains:

- Technical information about the crash.
- Technical information about the computer such as amount of memory, number of computer cores and a unique system ID. The unique system ID is hashed, and cannot be used to identify the user.
- Any comments and/or email you add to the crash report before sending it.
- The IP address of the user.

Log files

Every time OpendTect starts a logfile is created in the *Survey Data Root\LogFiles* folder. The naming of the log file is *username_datetime.txt* file.

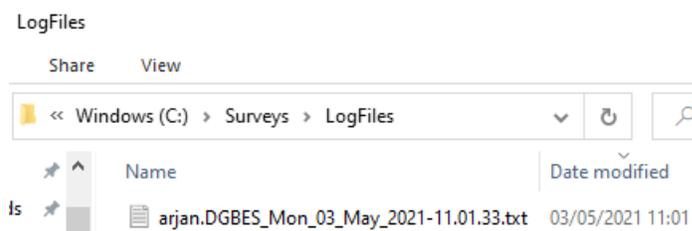


Figure: log files in LogFiles folder

The OpendTect Installer writes logfiles into the */tmp* folder on Linux or *%TEMP%* folder on Windows, the naming of these files are:

- *username_od_instmgr_install_log.txt*
- *username_od_instmgr_updcheck_log.txt*

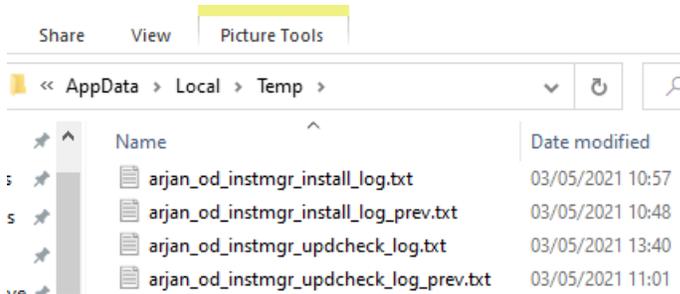


Figure: OpendTect installer log files

Settings files in .od folder

OpendTect writes its settings into the `$HOME/.od` folder on Linux or `%User-Profile%\od` on Windows. You can find the following files:

- qtsettings_odver; e.g. qtsettings_660
- settings
- settings_coltabs
- settings_dataroot
- settings_presentation
- settings_python
- settings_snapshot
- settings_welldisp
- settings_welltie
- survey

If one or more of the above listed settings files, like the `settings_coltabs`, does not exist in your `.od` directory this means that OpendTect hadn't (yet) had any reason to create that particular settings file.

Editing of these files is at your own risk. If you delete them OpendTect will create a new file when needed. If a file gets corrupt then deleting is exactly what you should do.

Debugging OpendTect

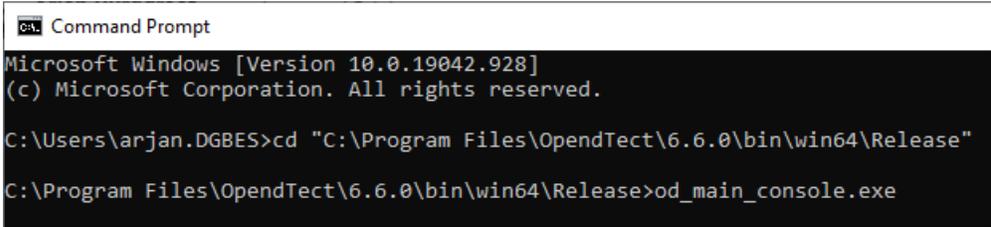
The default setting is that OpendTect only shows minimum debug info when starting OpendTect via the *start_dtect* script on Linux. Windows users will not see any debug info when starting *od_main.exe*.

In order to get more debug information an environment variable DTECT_DEBUG needs to be set to yes.

It is also possible to get debug information from Qt by setting an environment variable QT_DEBUG_PLUGINS to 1. On Linux one can also get Qt debug information when launching the OpendTect Installer.

When launching *start_dtect* from terminal on Linux the debug messages will be visible in the terminal.

Windows users will have to launch the Command Prompt, change the directory to *path_to\OpendTect\6.6.0\bin\win64\Release* directory and from there run *od_main_console.exe*



```
Command Prompt
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\arjan.DGBES>cd "C:\Program Files\OpendTect\6.6.0\bin\win64\Release"
C:\Program Files\OpendTect\6.6.0\bin\win64\Release>od_main_console.exe
```

Figure: starting *od_main_console.exe*

Appendix A - FlexNet Licenses Explained

Let us suppose you have received a 2-user license for a module. Normally you will have obtained a floating license, meaning that you will be able to use the module from any machine, and two users can be busy with it at the same time.

Server side: The License Manager Daemon

To be able to enforce the license, there must be a piece of software somewhere that keeps track of who is using the module. That is the License Manager Daemon (LMD). The LMD can run on any machine, also on machines that you will never use the module itself on. Good candidates are stable UNIX servers.

When the LMD is started, it looks at a file containing the information about what should be supported. This file is called the License File. Actually, the LMD can only be started on the machine indicated in the license file. The license file could look like this:

```
SERVER                licserv                000347e8b845
DAEMON  dgbld  /apps/opendtect/6.6.0/bin/lux64/lm.dgb/dgbld
FEATURE dTect  dgbld  6.600  1-jan-2020  2  6592FDC619EA  DUP_
GROUP=D
FEATURE dTectDS  dgbld  6.600  1-jan-2020  2  011D5153D870  DUP_
GROUP=D
```

The first line tells us that the LMD must be started on the machine `licserv`, with FlexNet ID `000347e8b845`. The second line is interesting for the LMD only but then you see two actual license FEATURE lines (`dTect` and `dTectDS`). These licenses are valid for versions 6.6 and lower, until the 1st of January 2020, for two users (6.600 1-jan-2020 2).

The Client side: Your program

Now let's look at the machine that you run your software on. The program will at some point in time need to check whether there is a license for what you are trying to do. For this, the program looks at the same license file. It sees that it has to contact the machine 'licserv' to ask for permission. The LMD keeps track of the number

of users already using the license feature. If a license is granted, your program will go on, otherwise you'll get an error message.

Non-floating licenses

In some cases the software will never be used other than on a certain machine. In that case a node-locked license can be issued. For such a license you do not need to start a license manager daemon - an unlimited number of users can use the module at the same time provided they work on that particular machine. A special case is the unlimited demo license, which grants unlimited access for any number of users on any machine. It goes without saying that this kind of license is always for short periods.

Host identification

For the above schemes to work, the license server or the running machine must be uniquely identified. Therefore, you will be asked to provide a unique host ID and a hostname when you want to obtain a license. Different operating systems require different ways to obtain this information:

- **Windows:** On windows, there is a FlexNet utility that delivers both in a simple file that can be sent by email. It is delivered together with commercial plugins so you need to install one of those first. From the Start menu, select OpendTect > License Manager Tools. Select the tab 'System Settings'. Then push the 'Save HOSTID Info to a File' button.
- **Linux and Mac OS X:** The hostname is obtained with the command `hostname`. The host ID differs per distribution, but can always be obtained by the 'lmhostid' tool. This tool is delivered with OpendTect, and can be run from the Utilities > Batch Programs dialog.

If you need to obtain the host ID before OpendTect is installed:

- **Windows:** Open a 'command prompt' (For example, by running `cmd`) and issue a command like: `ipconfig /all > c:\Temp\ipcfg.txt` You can send `ipcfg.txt` or look for the 'Physical address'.
- **Linux and Mac OS X:** Run `ifconfig` in a terminal. Or if `ifconfig` is not installed run `ip a`. Look for `HWAddr` or `ether`. You want the MAC address, looking like `xx:xx:xx:xx:xx:xx` - that's 6 groups of 2 hex numbers. Example: `00:1C:C0:38:22:F1`. Usually you would want to provide the one that is reported with devicename `eth0` or `eno1`. In doubt you can provide all.

If for some reason one of these commands does not work: we need the MAC address of the main network card. Sending the IP address of a machine is never helpful.

Conclusion (Manager's summary)

The FlexNet license system is based upon internet technology. Therefore you can run your software on any machine, using any operating system, to get licenses from any other machine regardless of operating system or physical location. Thus, a Linux license server in Houston can manage the licenses for Windows, Linux and Mac OS X machines in Houston, Caracas and Paris. The only restriction is the number of users actually using the 'feature' at a certain time, but that is what you pay for.

For more information, see the [FlexNet Licensing End User Guide](#).