BOXERsolutions is acting as an agent for Cambridge Flow Solutions Ltd.

This guide is provided courtesy of Cambridge Flow Solutions Ltd. and subject to Cambridge Flow Solutions Ltd.'s waiver.

BOXER Installation Guide

BOXER version 3.10.0

For further information please contact.

BOXER solutions

BOXERsolutions Ltd.
Suite 1
200 Cambridge Science Park
Milton Road
Cambridge
CB4 0GZ
info@boxersolutions.com
www.boxersolutions.com

LIABILITY

Cambridge Flow Solutions Ltd. ("The Company") has used all best endeavours to ensure the accuracy of the work performed and any information given, but the Company makes no warranty, express or implied, as to accuracy and will not be liable for any consequences arising out of any inaccuracies or omissions.

The Company excludes all warranties, representations or liabilities to the fullest extent permitted by law. Without prejudice to the generality of the foregoing the Company excludes any liability for consequential loss or damage including economic loss, loss of profit, revenue or goodwill.

Installation

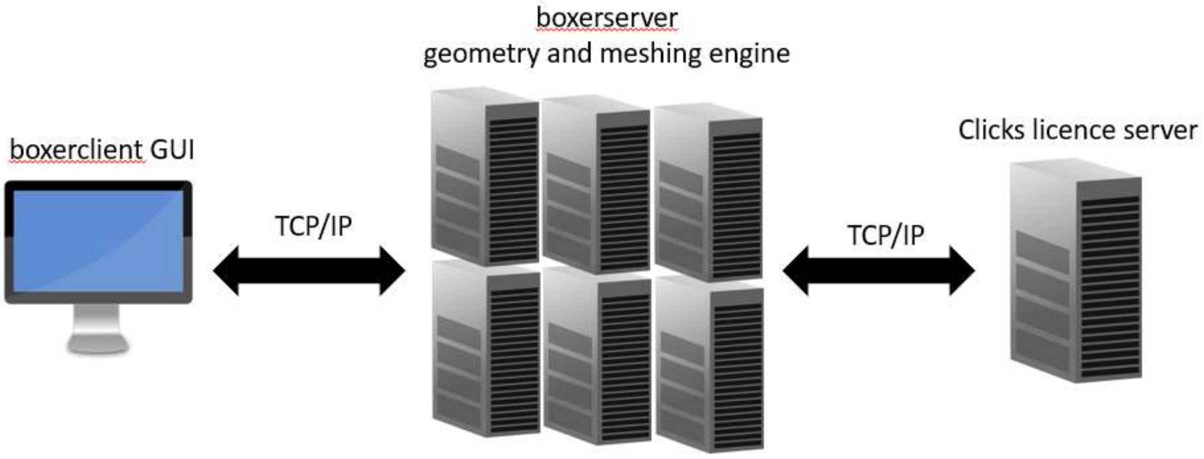
Summary:

The following steps are required to install BOXER :

- Install the licence server `clicks`. See Clicks installation for Linux on page 5, and/or Clicks installation for Windows on page 11
- Send `clicks` server ID to CFS to receive licence keys.
- Install licence keys.
- Install `boxer` packages. See BOXER installation for Linux on page 6, and/or BOXER installation for Windows on page 13.
- (Optional, Linux only) Install `prism` packages.
- (Optional, Linux only) Advanced configuration options for Linux installations.

There are three key components to a BOXER installation:

- `clicks`. A licence server.
- `boxerserver`. The core geometry/m/geometry server; connects to `clicks`.
- `boxerclient`. Meshing GUI; connects to `boxerserver`.



'Prism' - a mesh conversion utility for the Linux command line is also provided for installation.

Licence server – "Clicks"

CFS applications are licensed using a proprietary system called "Clicks". This is provided as a small utility Clicks.rpm in Linux, and a small executable installer package in Windows.

It runs as a system service on a Linux server machine, or as a standalone executable on a Windows server machine, which needs in either case to be accessible by network to all the machines running the actual BOXER software. If desired, the licence server machine can be the same as that on which the BOXER software will run.

Restriction:

BOXER versions 3.2.X and higher require Clicks version 2.0.X or higher.

BOXER versions 3.6.X and higher require Clicks version 2.3.X or higher.

BOXER versions 3.9.X and higher require Clicks version 3.0.X or higher.

Clicks itself is a parallel-aware licensing technology which utilises secure public key cryptographic techniques, providing the end user with an easy to install and administer licence-serving environment. Clicks has a distinction between processes and application licences, permitting the user to maximally exploit their multi-core and cluster-based parallel resources without explicit per-node licensing conditions.

A *licence server* is a machine whose purpose is to share an organisation's licences amongst its users and machines. It must have network connectivity (on a single TCP port) to all of the required client computers. The licence server and client computer may be the same.

Clicks serves licence requests via **port 8343**, so any firewalls within the system must be configured to allow this port to connect to clients.

Only one Clicks instance needs to be installed on each of the machines which are to act as an organisation's licence servers. That is, multiple CFS applications may share the same Clicks server.

A licence server may act as a *forwarder* to another licence server. For example, machines in a cluster behind a firewall may have their licence requests forwarded, via a 'head' node, to a central licence server.

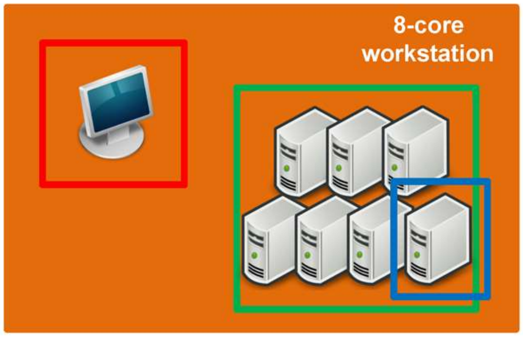
Tip:

Windows & Linux versions of BOXER and Clicks are essentially interchangeable. That is, either OS's version of the meshing server can communicate with either OS's version of the licence server.

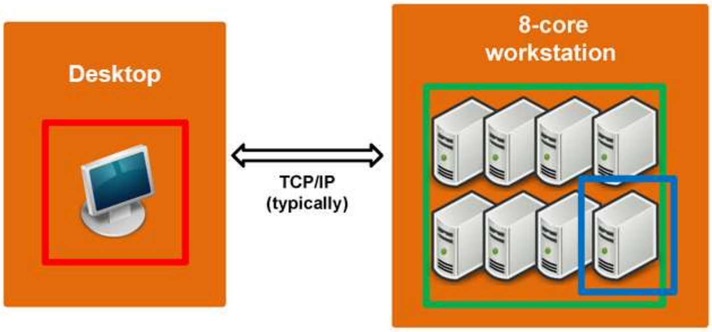
BOXER deployment scenarios

BOXER can be deployed in a number of different ways. The meshing server `boxerserver` is the core program. It runs in parallel within an MPI environment, either in server mode for a client GUI session, or in script mode for standalone batch meshing. The `boxerserver` process connects to the `clicks` server via a TCP/IP port. The `boxerclient` GUI connects to a `boxerserver` process via a TCP/IP port.

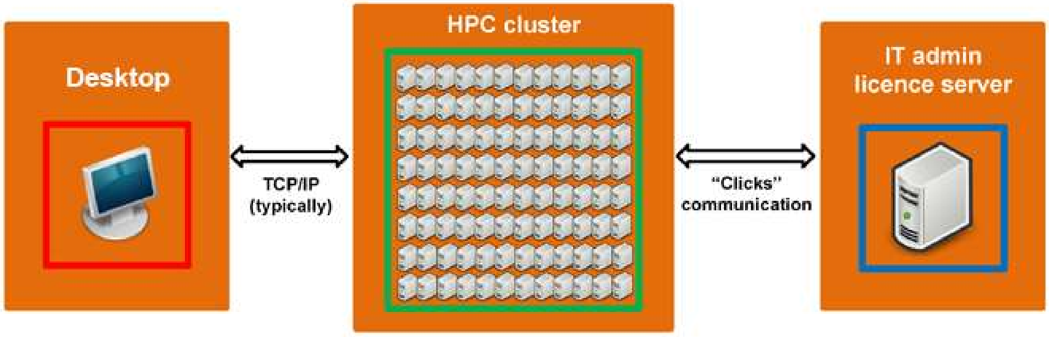
- *A standalone workstation*. In this scenario all packages (`clicks`, `boxerserver`, `prism` (Linux only), and `boxerclient`) are installed on the same workstation. This is the default scenario for BOXER in Windows.



- *A workstation for meshing, connected to a GUI*. A typical workgroup installation will see `clicks` installed on a nominated single Linux server/workstation, `boxerserver` and `prism` are installed on a range of Linux machines, and `boxerclient` is installed on Windows or Linux workstation as required.



- *Large parallel meshing on a Linux cluster, connected to a GUI*. A cluster installation is very similar to a workgroup installation, in that a single `clicks` server must be nominated and `boxerserver` must be installed across the (Linux) cluster nodes, with `boxerclient` installed on workstations that are able to see the head node of the cluster.



- *HPC and task scheduling*. An HPC installation differs from a cluster installation in that it typically requires users to submit jobs to a queue, and will require the installation to be correctly configured to use high-performance interconnects (e.g. Infiniband). See "Advanced installation configuration (Linux)" on page 7

BOXER software packages

BOXER is available for the Linux platform, and the Windows platform (except the `prism` utility).

The software packages can be downloaded from CFS's ftp server. Please contact us to obtain an account. The software is licensed and can only be run according to the terms of your licence agreement.

Linux

CFS uses the RPM binary-distribution format for Linux. RPM (Red Hat Package Manager) is a package management system and a core component of many Linux distributions including Red Hat Enterprise Linux, SuSE Linux Enterprise and the Fedora Project. It cleanly manages the installation, removal, verification and updating of software packages and provides functionality to a number of graphical package management tools including SuSE's YaST or Red Hat's system-config-packages.

CFS prepares and distributes a number of pre-compiled binary RPMs for a variety of Linux distributions (64-bit only):

- RHEL and CentOS 5, 6 and 7
- SuSE Linux Enterprise Server 11
- openSuSE 11.1, 11.2, 11.3, 11.4, and 12.1

Microsoft Windows®

CFS provides a standard Microsoft Windows Installer packages for BOXER and Clicks. These are compatible with Windows 7 (32-bit and 64-bit), Windows 8 and 8.1 (32-bit and 64-bit), and Windows 10 (64 bit).

Linux installation

Clicks

You will need root authority on the installation machine. Most Linux distributions have both command-line and graphical interfaces for RPM installation. If your preferred method fails, try the other.

Tip:

To see which CFS packages are installed on a system, type:

```
rpm -qa --queryformat '%{name}-%{version}-%{release}.%{arch} (%{vendor})\n' | grep
"Cambridge Flow Solutions"
```

RPM Signatures

Note:

The following step for RPM signatures is optional. If you skip this step, or if your machine has no internet connection, some of the later commands will flag up a "GPG-key" related warning. This can be ignored if you trust the source of the RPM.

Using public key cryptographic techniques, RPMs can be digitally signed by the issuing software vendor. This gives the end user confidence that the package has not been tampered with and does not, therefore, contain any malicious code. To check the authenticity of CFS packages, the system administrator will need to import CFS's public key once before installing any CFS packages.

To import CFS's public key, type:

```
rpm --import http://www.cambridgeflowsolutions.com/support/gpg/RPM-GPG-KEY-cfs
```

RPMs for Clicks

Clicks is provided by a single RPM, for a variety of Linux distributions. The RPM can be installed as root on the machine which is to act as the Clicks licence server :

```
rpm -ivh clicks-*.rpm
```

This will install the system service `clicks` in `/etc/init.d/clicks`, and also will make (by default) the directory `/etc/clicks/` in which licence files are stored (see later).

Configuration

Note:

When the licence server is the same machine on which the application software is to be run, there should be no more configuration necessary. The following steps are only required when the licence server is on a different machine.

The server is configured by changing (if necessary) the values of the following environment variables as defined in `/etc/init.d/clicks`

- `CLICKS_SERVER` - IP address and TCP port on which the server should listen. E.g.

  ```
  export CLICKS_SERVER=0.0.0.0:8343
  ```

- `CLICKS_LICENCES` - directory where licence files are located.
- `CLICKS_LOG` - filename of server log file.

- CLICKS_PROXY - IP address and TCP port of licence server to which this licence server should forward requests. This variable should be unset if the server does not have to forward requests.

Server ID

In order to issue licences, CFS needs to know a unique identifier of the physical hardware which is to serve licences. This can be found, after installing the Clicks software, by starting the Clicks service.

Clicks is configured to start automatically on every reboot, but to start it manually, either use a system service manager (e.g. system-config-services on Red Hat operating systems) or, as root, type

```
/etc/init.d/clicks start
```

at a command line prompt.

The server ID can be found in the licence server log, /var/log/clicks. Either email the whole of /var/log/clicks to CFS, or just the 'Server ID' (e.g. **x3MqJEqDU3pYoMy5nXhDEYwOVmc**)

Installing licences

After receiving a licence file from CFS, copy it into the /etc/clicks directory on the licence server. The licence file needs to be permitted readable by the server daemon; make sure that its permissions are at least set to 644 using chmod.

Then, restart the Clicks service.

```
/etc/init.d/clicks stop
```

```
/etc/init.d/clicks start
```

Clicksinfo

Clicksinfo is provided by a single RPM, for a variety of Linux distributions. The RPM can be installed as root on any machine.

```
rpm -ivh clicks-clickstat-3.1.0-.x86_64.rpm
```

Clicksinfo connects to the remote clicks server via TCP/IP to provide licence status information to the user.

BOXER

You will need root authority on the installation machine. Most Linux distributions have both command-line and graphical interfaces for RPM installation. If your preferred method fails, try the other.

RPMs provided in the software distribution.

BOXER as an overall package is provided by four separate RPMs.

- boxer-docs*.rpm. This RPM has no dependencies. It creates the subdirectory pdf/ under /usr/share/boxer/doc when default-installed by rpm.
- boxer-server*.rpm. This has no X11 dependencies, and so can be installed in cluster nodes without X11 graphics. It creates /usr/bin/boxerserver and /usr/bin/mpirun-boxerserver - and also creates /usr/bin/stl2btfpro.
- boxer-client*.rpm. This requires X11 libraries to have been installed. Creates /usr/bin/boxerclient which is the GUI client; which must connect to a pre-started boxerserver instance.

- boxer-workstation*.rpm. This requires boxer-client and boxer-server, and so must be installed at the same time. This creates /usr/bin/boxer (see below), installs icons in the KDE and GNOME menus and registers the *.bxr file extension and associates MIME-type and icon with the desktop environment.

To install BOXERmesh v3.10.0 by upgrading an existing legacy BOXERmesh version:

1) Workstation only : `rpm -Uvh boxer-*.rpm`

2) Cluster node or meshing server : `rpm -Uvh boxer-server*.rpm`

3) GUI client workstation (no server) : `rpm -Uvh boxer-client*.rpm boxer-docs*.rpm`

The script /usr/bin/boxer

This script :

- Starts N copies of the meshing server locally, where N >= 2 and should be equal to the number of local cores (as determined from /proc/cpuinfo)
- When the server is running, launches a client session and opens any files that were passed as command-line arguments
- Uses a "local socket connection" which does not require opening any network ports and has lower communication overheads

The Prism utility, and OpenMPI

An OpenMPI build is included within the boxerserver package. This will be installed in the CFS-specific path on the host machine, and thus will not interfere with any other MPI distributions that may be installed from/by other software packages. This OpenMPI build is configured with support for queuing systems (SGE and SLURM) and Infiniband hardware.

Also provided alongside the core BOXERmesh software, is the prism utility prism*.rpm. This package provides simple command-line translation of geometry and mesh formats between each other; providing, for example, mesh files for various commercial analysis packages as straight translations of the raw BOXERmesh mesh output.

Connecting Clicks and BOXER

The environment variable CLICKS_SERVER tells BOXER where to look for a licence.

If the Clicks licence server is different from the machine on which BOXER is to run, the environment variable CLICKS_SERVER should be set to the IP address or host name of the licence server machine.

This can be done on a per-user basis by editing $HOME/.bashrc and adding the line:

```
export CLICKS_SERVER=host
```

or

```
export CLICKS_SERVER=host:port
```

where port = 8343

Advanced installation configuration (Linux)

There are various advanced options and methods for installing and running BOXER in, for example, customised HPC environments.

Installing RPMs in non-standard file locations

There may be situations where you wish to install the BOXER RPMs in non-standard locations on your filesystem, for example for permissions reasons, or to install on a shared filesystem that has a different mount point.

This is easily achieved with the `rpm` system command; the `boxer-server`, `boxer-client`, and `prism` packages are relocatable. To install in a non-standard location, simply use the `--prefix` flag when issuing RPM commands, for example:

```
rpm --prefix /path/to/local/installation/ -i *.rpm
```

For further information on this, please see the RPM documentation.

Installing RPMs without root access

It is best to install BOXER packages on Linux machines as root, as this minimises the amount of additional configuration that might be required. However, if this is not possible, then there are two methods by which the RPMs can be installed by a regular user: using the `rpm` command with an alternate RPM database; unpacking the RPM package and copying the files.

Note:

When installing BOXER using either of these two methods, you may need to additionally set the `PATH` and `LD_LIBRARY_PATH` environment variables.

Using `rpm` with an alternate RPM database

You can use a separate local RPM database to avoid the requirement that regular users cannot modify the system RPM database. This is achieved using the `--dbpath` RPM option. This is an advanced topic and you are advised to read the RPM documentation before proceeding.

You may also need to use the `--prefix` option as described above to avoid filesystem permission issues.

In practice, you must set up a fresh RPM database, as follows:

```
 cd /path/to/new/root/
 mkdir rpm_db
 rpm --initdb --dbpath=/path/to/new/root/rpm_db/
```

You only need to perform this step once. Then, install/upgrade/query/etc. packages as normal using the rpm tool as follows:

```
 rpm --dbpath=/path/to/new/root/rpm_db/ --prefix=/path/to/local/installation/
--nodeps -i *.rpm
```

The `--dbpath` option sets the path to the local RPM database and the `--nodeps` flag is required because `rpm` only knows about dependencies based on the current RPM database being used, thus any system-installed packages are not visible. (You need to make sure that any dependencies are satisfied.)

The advantages of this method are that you can use the `rpm` tool to keep your BOXER installation up to date, and you don't have to worry about manual filesystem operations.

Unpacking rpm packages

You can unpack RPM packages using the `rpm2cpio` tool, and then copy the resultant files to a location of your choice.

To unpack `x.rpm` to the current directory, use:

```
 rpm2cpio x.rpm | cpio -ivd
```

This will unpack all the files in the RPM package and will create a local directory structure.

HPC issues

BOXER works very well on HPC systems but some additional configuration may be required to run optimally. The three areas where issues can arise are:

- Use of local MPI installations
- Launching `boxerserver` via a queuing system
- Exposing the `clicks` licence server to the compute nodes

Local MPI Installation

The OpenMPI v1.10 build that is included in the `boxer-server` rpm package contains support for SGE and SLURM queuing systems, and is also built with Infiniband drivers. For many systems, no further configuration should be necessary and it should be possible to run BOXER using the commands previously described:

```
mpirun-boxerserver -np N boxerserver --script $my_script.lua
```

However, there may be instances where running with a locally compiled version of OpenMPI is either necessary or desirable. For example, you may need support for an alternate queuing system or need to compile against system-specific network infrastructure libraries. This is supported provided the advice below is followed. For reasons of compatibility, stability, and performance, you should use OpenMPI v1.10.x

To run BOXER with a local OpenMPI build, care must be taken in the following areas: Specifying the correct path to the locally-built `mpirun`; Setting `LD_LIBRARY_PATH` such that the core `boxerserver` libraries can be found; Specifying the full path to the correct `boxerserver` image, and avoid using the `boxerserver` wrapper program.

- *Path to local* `mpirun`. In practice, HPC systems typically use the module system to correctly set paths. You should ensure the correct openmpi-1.10 module is loaded and that `which mpirun` resolves to the correct version of `mpirun`. If your system does not use modules, then you must take care to fully specify the path to `mpirun`.
- *Setting* `LD_LIBRARY_PATH`. Under this scenario, we avoid using the `boxerserver` wrapper script, therefore we must set `LD_LIBRARY_PATH` such that `boxerserver` can locate the correct libraries. You will need to `export LD_LIBRARY_PATH=/usr/lib64/boxerserver/lib64` or if you installed `boxerserver` to a non-standard location using `--prefix` as above, then `export LD_LIBRARY_PATH=[$PREFIX]/usr/lib64/boxerserver/lib64` . If you are a HPC system administrator, or you can ask your administrator to do this, it may be worth generating a boxer module to set these environment variables on your system for ease of use.
- *Correct* `boxerserver` *image*. Instead of running `/usr/bin/boxerserver`, you must execute `/usr/lib64/boxerserver/bin/boxerserver64` or `/usr/lib64/boxerserver/bin/boxerserver128` directly (for standard-precision or extended-precision as appropriate). Again, if you used an install prefix, then take care to ensure you have the correct path.
- *Putting it all together*. `export LD_LIBRARY_PATH=/usr/lib64/boxerserver/lib64; /path/to/my/mpirun -np [np] /usr/lib64/boxerserver/bin/boxerserver64 --server`

We have found that there is a performance benefit from using certain OpenMPI mca flags. If you are using a locally built version of OpenMPI, please try running with the following options:

```
mpirun -mca coll_tuned_use_dynamic_rules 1 -mca coll_tuned_alltoallv_algorithm 1
```

These options are enabled by default in the provided `mpirun-boxerserver` and do not need to be specified additionally.

Queuing systems

BOXER can be run via a job submitted to a queuing system, e.g. SGE or SLURM. Please seek advice from your HPC system administrator regarding the correct syntax and settings to use. In broad terms, you will need to take care with the following points: Set the path to `boxerserver` correctly, particularly when using alternate MPI installations; Set `LD_LIBRARY_PATH` correctly, if appropriate; Set the path to correct `mpirun` executable.

As above, deployment on a HPC system with a queuing system is made easier if a local BOXER module is created and you should speak to your system administrator about this.

`clicks` *licence server visibility*

On clusters and HPC system in particular, the slave nodes are typically connected to the head node via a separate network. For `boxerserver` to run successfully, the slave nodes must be able to communicate with the `clicks` server.

For simple, standalone clusters this is most easily achieved by installing the `clicks` server on the head node of the cluster thus ensuring visibility to all the slave nodes. However, in many instances this may not be desirable nor possible, and you must open up a route from the slave nodes to the licence server.

`clicks` can be configured to provide basic port forwarding using the `CLICKS_PROXY` option (See "Licence server" on page 1) but this may not be sufficient. For these more complex situations, tunnelling via ssh port forwarding must be used and you are advised to seek advice from your system administrator as to the best approach for this, considering your particular setup.

Linux Clicks server access control

Note:

The following section is only relevant if you wish to restrict access to the licence server, or are having problems connecting to it.

Clicks has built in support for TCP Wrappers, a standard UNIX mechanism for controlling clients' access to Internet services. It is possible to restrict access to the licence server depending upon the IP address of the client via the `/etc/hosts.allow` and `/etc/hosts.deny` files that are commonly used for system administration. For example, to only permit clients from the `192.168.8.*` subnet, you could place the following in `/etc/hosts.allow`

```
clicksd: 192.168.8.0/255.255.255.0
```

And in `/etc/hosts.deny`

```
clicksd: ALL
```

For more details, please consult your local manual page, `man 5 hosts_access`.

Caution:

If '`ALL: ALL`' is present in `/etc/hosts.deny`, you must add '`clicksd: ALL`' to `/etc/hosts.allow` before any connections to the licence server will be possible.

Windows installation

Clicks

Before You Install

- You will need administrator privileges.
- You should close all other programs before starting this installation.

Start the installer

Double-click on the installer file, `clicks-*.exe`. This will ask you to run as Administrator before launching the Clicks setup wizard.

Click `Next`.

Review the Licence Agreement (EULA)

If you agree, click the `I Agree` button to proceed.

Choose options for the install

You can choose whether to add Clicks to the system path for all users, or just for the current user - and whether to create a desktop icon. By default, the installer does not add Clicks to the system path for all users and does not create a desktop icon, **however we do recommend creating a desktop icon for ease of implementation.**

Choose the install location

Choose the name of the folder where you want to install the Clicks server software. Accept the default destination folder, or click `Browse` to select a different one. If the folder does not exist the installer will create it.

After making your selection, click the `Next` button.

Choose the Start Menu folder

Choose the name where the installer puts shortcuts for the Clicks server in the Windows Start Menu.

After making your selection, click the `Next` button.

Complete the installation

When the installation has successfully completed, the installer displays the `Completing the Clicks Setup Wizard` dialogue.

Click `Finish` to complete this installation.

Set the following as SYSTEM level environment variables:

```
CLICKS_SERVER=<hostname>:8343

CLICKS_BIND=<hostname>:8343
```

`<hostname>` is the device name of the computer on which you have done the Clicks installation, it can be found from `Start > Settings > System > About > Device Specifications`

Obtaining a Server ID

In order to issue licences, CFS needs to know a unique identifier of the physical hardware which is to serve licences. This can be found, after installing the Clicks software, by starting the Clicks service.

Go `Start > Clicks 3.0.0 > Clicks server` to start the Clicks licence server. Alternatively, double-click the desktop icon:

A command window will start, showing the log of the licence server process. The server will either find a valid licence and begin normal operation, or will not find a valid licence – in which case it will show the server machine ID in the command window (from where it can be copy-pasted Ctrl-C Ctrl-V), and then exit upon user key-press.

Caution:

When you start the Clicks server, the software will change the system-wide terminal modes for "Quick edit mode" and "Insert"; these modes will become disabled. Without this, if you click in the terminal window within which the Clicks server is running, then the Clicks program is halted by the operating system (i.e. licence serving is stopped) due to the default Windows terminal settings.

These Windows settings can be reset to system defaults if necessary; however if so, then you might find that Clicks operation is interrupted in case of clicking in the terminal window.

The server ID can also be found in the licence server log `C:\ProgramData\CFS\clicks\log.txt`. Either email the whole of `C:\ProgramData\CFS\clicks\log.txt` to CFS, or just the 'Server ID'.

Note:

The Windows Clicks data directory `C:\ProgramData\CFS\clicks` is not visible by default in Windows. To see it, you need to enable view of “Hidden items” in the Windows File Explorer.

Installing licences

Licence files need to be copied into the directory `C:\ProgramData\CFS\clicks\etc\`.

When the licence server is restarted (close any previous instances of it, then re-open it e.g. by double-clicking the desktop icon) the command window will relay the scan of the licence file and the licence server process will start.

Note:

The normal startup state of Clicks with a valid licence ends with the line “Examining licence file…” in the terminal window.

Clicksinfo

Double-click on the installer file, `clicksinfo-3.1.0-x64.exe` . This will ask you to run as Administrator before launching the Clicks setup wizard.

Click `Next`.

Review the Licence Agreement (EULA)

If you agree, click the `I Agree` button to proceed.

Choose options for the install

You can choose whether to add clicksinfo to the system path for all users, or just for the current user - and whether to create a desktop icon. By default, the installer does not add Clicks to the system path for all users and does not create a desktop icon, **however we do recommend creating a desktop icon for ease of implementation.**

Choose the install location

Choose the name of the folder where you want to install the clicksinfo server software. Accept the default destination folder, or click `Browse` to select a different one. If the folder does not exist the installer will create it.

After making your selection, click the `Next` button.

Choose the Start Menu folder

Choose the name where the installer puts shortcuts for the clicksinfo server in the Windows Start Menu.

After making your selection, click the `Next` button.

Complete the installation

When the installation has successfully completed, the installer displays the `Completing the Clicksinfo Setup Wizard` dialogue.

Click `Finish` to complete this installation. You should now see the `clicskinfo` icon on your desktop

BOXER

Before You Install

- You will need administrator privileges.
- You should close all other programs before starting this installation.

Start the installer

Double-click on the installer file, `boxer-*.exe`. This will ask you to run as Administrator before launching the BOXER setup wizard.

Click `Next`.

Review the Licence Agreement (EULA)

If you agree, click the `I Agree` button to proceed.

Choose the install location

Choose the name of the folder where you want to install the BOXER client. Accept the default destination folder or click `Browse` to select a different one. If the folder does not exist the installer will create it.

After making your selection, click the `Next` button.

Choose the Start Menu folder

Choose the name where the installer puts shortcuts for the BOXER client in the Windows Start Menu.

After making your selection, click the `Install` button.

Note:

The BOXER Windows installer will also install, if not already present on the system, the following Microsoft system packages/libraries:

- Microsoft MPI
- DirectX9
- Visual Studio run-time libraries 2013 & 2015

Complete the installation

When the installation has successfully completed, the installer displays the `Completing the Boxer Setup Wizard` dialogue.

Click `Finish` to complete this installation.

There will now be 3 new icons on your destop. “BOXER”, “BOXER server” and “BOXER client”

- “BOXER” runs the meshing server and GUI client simultaneously on the local PC
- “BOXER server” runs the meshing server only – It awaits a connection from a local or remote client down the stated port (by default port 26937)
- “BOXER client” sarts the GUI client only. It requires connection to a local or remote meshing server, specified on the startup dialog. The meshing server should already have been started and be listening for a connection.

Desktop icons for “BOXER”, “BOXER server” and “BOXER client”

Installing the extended precision server

The extended precision server comes as a separate installer package. To install the extended precision server, follow the steps as described for installing BOXER, above. At the end of the installation process an icon called “BOXER server extended” will appear on the desktop.

Connecting Clicks and BOXER

The environment variable `CLICKS_SERVER` tells BOXER where to look for a licence.

By default (i.e. if not set on the system) then `CLICKS_SERVER` is `localhost:8343`.

Alternatively, for BOXER to look for a licence somewhere other than on the local machine, `CLICKS_SERVER` needs to be set to `IP_of_licence_server:8343` or `hostname_of_licence_server:8343`

Execution

Summary:

The terminology describing individual programs in the BOXER software is as follows.

- A *licence server*, or *Clicks server*, is a system-level service which serves licence keys to the meshing server process.
- The meshing server *boxerserver* is the core BOXER program. It can run in parallel within an MPI environment, either in *server* mode for a client GUI session, or in *script* mode for standalone batch meshing. The *boxerserver* process connects to the *licence server*.
- The GUI client *boxerclient* connects only to a meshing server (which needs to have been already started), and not directly to the licence server.

Note:

Windows & Linux meshing servers & GUI clients are interchangeable; A Windows version of one can connect to a Linux version of the other, and vice-versa.

Note:

In Windows, when running BOXER for the first time on a new machine, you will be prompted to add a firewall exception, *even when running the software only your local machine*. This is because, even though BOXER is only running locally, the MPI processes of it are all discrete process which need network connectivity amongst themselves – which requires the Windows firewall to permit it.

Clicks startup

In all circumstances before running BOXER, a Clicks licence server must be running for BOXER to connect to and the meshing server must know where to obtain a licence from (the address of the Clicks server)

Starting BOXER – the GUI and server at the same time

In Linux, for convenience there is a small utility for workstations that will start the BOXER client and server in one go. Simply typing

```
boxer
```

at the command line, will start a local OpenMPI session with N processes matching the local hardware, communicating with a local GUI client. The GUI will start directly.

In Windows, double-click the “BOXER” desktop icon. Double-clicking this icon calls a .bat script which will start the meshing server on a suitable number of CPU cores and then, subsequently, start the GUI client.

Caution:

By default, boxerclient connects to boxerserver via **port 26937**. If this port is in use by some other software on your system, the server will not start and return a PORT IN USE error in the terminal window.

To fix this – edit the file `<INSTALL_DIRECTORY>\bin\server.bat` and change port 26937 to another port (26936, say)

Basic script-based meshing using boxerserver

- In Linux, the BOXER meshing server can be executed in script mode in line with the generic command

  -
    ```
    mpirun-boxerserver -np N boxerserver --script $my_script.lua
    ```
- In Windows:

  -
    ```
    $path_to/mpiexec.exe -np N $path_to/boxerserver.exe --script
    $my_script.lua
    ```

In either case, `N` is the number of parallel processes to be started on the local machine, and `$my_script.lua` is some sort of suitable BOXER script.

In Linux, using the OpenMPI over a large parallel HPC resource, there are many ways to launch parallel `boxerserver` processes. Some of these are described in the output of `mpirun-boxerserver --help` and more details can be found on the Open MPI FAQ web page. For example, an obvious alternative to the command above would be:

```
mpirun-boxerserver -machinefile $my_nodes boxerserver --script $my_script.lua
```

where `$my_nodes` is a text file with the names of the compute nodes in a CFD cluster. All such nodes must be accessible to the user via `ssh` without a password being required.

Tip:

You can see the command-line options for `boxerserver` by typing `boxerserver --help`.

Starting boxerserver and boxerclient separately

First start the BOXER server

- In Linux `mpirun-boxerserver -np N boxerserver --server`
- In Windows `mpiexec.exe -np N boxerserver.exe --server` (similar to the script based command, above) Or double click the BOXER server icon, or the BOXER server extended icon (if the extended precision server is installed):

Caution:

By default, boxerclient connects to boxerserver via **port 26937**. If this port is in use by some other software on your system, the server will not start and return a PORT IN USE error in the terminal window.

To fix this – edit the file `<INSTALL_DIRECTORY>\bin\server.bat` and change port 26937 to another port (26936, say)

In either case, `N` is the number of parallel processes to be started on the server machine.

This command will obtain a licence form the licence server, start the meshing server process in the MPI scope specified, and listen for incoming commands from the client GUI process.

The GUI client is started after the server:

- In Linux from another terminal window using the `boxerclient` command

- In Windows double-clicking the desktop “BOXER client” icon.

This opens a popup window with two fields:

- *Host*. The target machine on which the server is running. Typically `localhost` (for a workstation), or a DNS entry or IP address for a meshing cluster.
- *Port*. The client/server communication port. By default, `26937`.

The GUI client will then connect to the meshing server, and the interactive BOXER session will start.

Note:

The necessary `server` and `port` options for the GUI may be specified via the following command-line/bat file options :

`--server` *hostname* : If present, skip connection dialog and attempt to connect to a BOXER server at the given hostname. The port, if unspecified, will default to the last port used when connecting.

`--port` *number* : If present, skip connection dialog and attempt to connect to a BOXER server on the given port number. The host to which the connection attempt is made will default to the last known host, and can be specified with the `--server` argument.

Note:

The default port for communication between client and server is `26937`. This can be changed easily, for example in case of running more than one server and client on a single machine, by using the `--port` option :

`mpirun-boxerserver -np 2 boxerserver --server --port` **5-digit port number**

with the port number given to the subsequent `boxerclient` set to match.

Note:

The default graphic driver used by BOXER is OpenGL.

Users with suitable hardware can use the following, better drivers, via command-line options:

- OpenGL2 driver. Start the `boxerclient` GUI with `--ogl2-driver`.
- DirectX9 driver (Windows). Start the `boxerclient` GUI with `--dx9-driver`. You may also wish to edit (as Administrator) the `boxer.bat` startup script to include this driver option.

Attempting to use a driver which is unavailable or not fully supported by your local system hardware, will result in unpredictable behaviour.

When no compatible OpenGL graphics card or driver is present on the hardware running the BOXER GUI, a basic graphic driver option may provide a solution based on legacy two dimensional graphic libraries.

In both Windows and Linux, this option is passed to `boxerclient` at the command line by using `--2d-driver`.

Specifying this lower-capability graphics option will result in a GUI without some of the high performance features of an OpenGL environment - such as lack of anti-aliasing, and degradation in the smooth rotation of the graphical view.

Licence-specific functionality - CAD import

Some BOXER functionality is available under certain licence-key configurations, i.e. requires a specific key to run. Licensed functionality may or may not be required by the user at run-time under different circumstances; e.g. a `boxerserver` user command may or may not lead to a GUI session (the `--server` option) and/or may or may not require CAD read capability.

If licensed, then CAD reading capability is available to the user by default when starting `boxerserver` in server mode, i.e. with the option `--server`. The CAD capability can be explicitly turned *off* for a session by using `boxerserver --server --no-cad`.

Conversely, the default for using BOXER in scripted-only mode (i.e. `boxerserver  --script`) is to have CAD capability turned *off*. In this case, if licensed then CAD access can be enabled by using `boxerserver --script --cad`.

Extended precision

Meshing and geometry precision for an increased range of length scales

Calculations are routinely undertaken using single precision arithmetic. However, where large variation in length scales is required, double precision arithmetic can be necessary, to provide the ability to create meshes and level-sets with much finer granularity (i.e. greater range in length scale).

In Linux the extended precision meshing server is started using the option `--extended` or `-x` directly after "`boxerserver`" at the run-time command line.

In Windows the extended precision meshing server is started by double clicking the `BOXER server extended` icon (if installed)

In normal precision BOXER the maximum level of octree sub division is 21 (i.e. the shortest octree element edge length is L/(2^21), where L is the longest octree element edge length

In normal precision BOXER the maximum level of octree sub division is 38 (i.e. the shortest octree element edge length is L/(2^38), where L is the longest octree element edge length

Note:

Enabling extended precision can cause a run-time slow down of about 30-40% and there is an increased memory requirement, due to the increased computational overhead of double-precision operations.

Clicksinfo

`clicksinfo` is a lightweight utility that can be run on any machine connected to a Clicks server (or not). It polls the Clicks licence server (by default, identified by the environment variable CLICKS_SERVER) via a TCP/IP connection and returns information about the currently available Clicks licences, how many seats and processes are in use and by whom, and the status of any other licence files (valid, expired or invalid). If `clicksinfo` cannot connect to the licence server, the reason is given, and the program exits.

On Linux machines, as a regular user, type `clicksinfo` at the command line. The default licence server and port can be overridden by typing

```
clicksinfo <servername>
```

Or

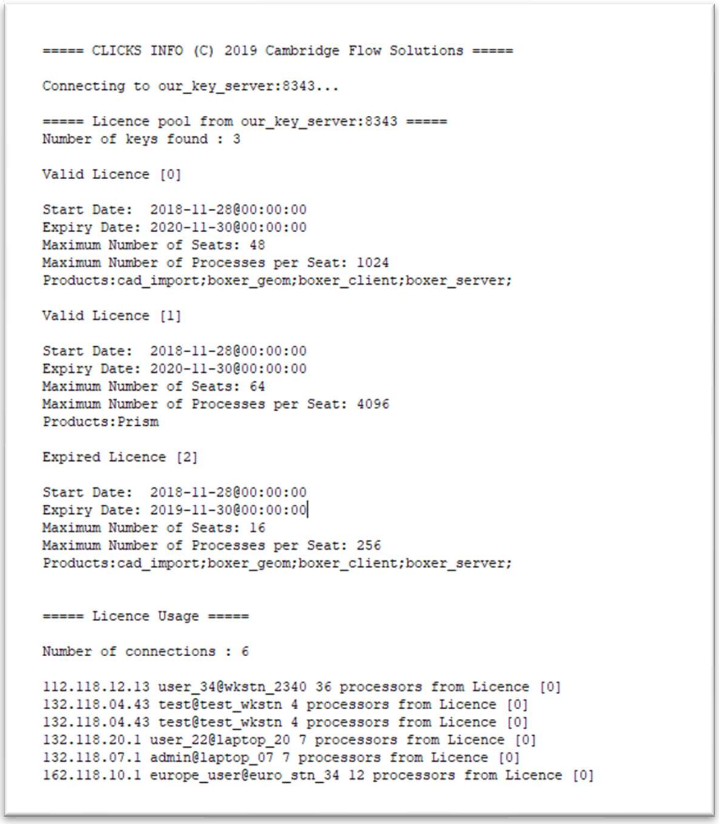
```
clicksinfo <servername>:<port>
```

Text will be sent to the std output, and the program returns the command prompt.

On Windows machines, as a regular user, double click the `clicksinfo` icon

.

A command window will appear with text output. Press any key on the keyboard to close the window and end the program:



```
===== CLICKS INFO (C) 2019 Cambridge Flow Solutions =====

Connecting to our_key_server:8343...

===== Licence pool from our_key_server:8343 =====
Number of keys found : 3

Valid Licence [0]

Start Date:  2018-11-28@00:00:00
Expiry Date: 2020-11-30@00:00:00
Maximum Number of Seats: 48
Maximum Number of Processes per Seat: 1024
Products:cad_import;boxer_geom;boxer_client;boxer_server;

Valid Licence [1]

Start Date:  2018-11-28@00:00:00
Expiry Date: 2020-11-30@00:00:00
Maximum Number of Seats: 64
Maximum Number of Processes per Seat: 4096
Products:Prism

Expired Licence [2]

Start Date:  2018-11-28@00:00:00
Expiry Date: 2019-11-30@00:00:00
Maximum Number of Seats: 16
Maximum Number of Processes per Seat: 256
Products:cad_import;boxer_geom;boxer_client;boxer_server;


===== Licence Usage =====

Number of connections : 6

112.118.12.13 user_34@wkstn_2340 36 processors from Licence [0]
132.118.04.43 test@test_wkstn 4 processors from Licence [0]
132.118.04.43 test@test_wkstn 4 processors from Licence [0]
132.118.20.1 user_22@laptop_20 7 processors from Licence [0]
132.118.07.1 admin@laptop_07 7 processors from Licence [0]
162.118.10.1 europe_user@euro_stn_34 12 processors from Licence [0]
```

Example output from clicksinfo

No other CFS software products need to be installed on a machine for `clicksinfo` to run – it is a completely stand-alone utility.