

PRACTICAL GUIDE · CFD

OpenFOAM on a Linux server, working primarily from Windows

A practical guide to FileZilla, Notepad++, PowerShell, solver logs, and ParaView client/server.

Windows 10/11 · Linux SSH/SFTP · OpenFOAM 14 · ParaView 5.13.2

General-purpose operating guide, independent of any particular server or user

Edition: 16 October 2026 · Examples: OpenFOAM Foundation 14, ParaView 5.13.2, Windows 10/11, Linux with SSH/SFTP.

Goal. Prepare and edit cases on Windows, transfer them with FileZilla, run only the necessary commands from PowerShell, download and inspect logs on Windows, and visualize results using ParaView for Windows connected to Linux `pvserver`. You do not need to install OpenFOAM on Windows for this workflow.

1. Architecture: what runs where



Figure 1 · GUI applications and case preparation stay on Windows; computation and data reading take place on Linux.

Task	Windows PC	Linux server
Edit dictionaries in <code>system/</code> , <code>constant/</code> , <code>0/</code>	Notepad++ or VS Code	Holds the execution copy
Transfer cases, scripts and logs	FileZilla over SFTP	SSH service and case directories
Start meshing, checks and simulations	PowerShell → SSH	OpenFOAM, CPU/RAM and storage

Task	Windows PC	Linux server
Monitor computation	PowerShell for commands; FileZilla for log copies	<code>tail</code> , <code>grep</code> , growing log files
Examine logs	Notepad++, klogg , spreadsheet or Python	Optional text-processing commands
Visualize CFD data	ParaView GUI	pvserver , which reads simulation results

Replace three names in the examples: `USER` is your Linux account, `SERVER` is the reachable DNS name/IP address, and `myCase` is the case directory. All instructions assume authorized SSH access, normally on port 22. If the server uses a different SSH port, use `ssh -p PORT` and set the corresponding port in FileZilla.

2. One-time setup on the Windows PC

1. Install **FileZilla Client** (not FileZilla Server) from <https://filezilla-project.org/> .
2. Install **Notepad++** from <https://notepad-plus-plus.org/> . VS Code is an alternative, particularly for navigating many files.
3. Install **ParaView for Windows**, preferably the **same version** as Linux `pvserver` (5.13.2 in these examples): <https://www.paraview.org/download/> . Avoid mixing different client and server versions.
4. Check the OpenSSH client: open PowerShell and run `ssh -V` . If it is missing, enable the optional Windows feature **OpenSSH Client**. There is no need to install **OpenSSH Server** on your PC.
5. Optional but useful: **klogg**, a viewer for very large logs, from <https://github.com/variar/klogg> .

For example, create `C:\CFD\cases\myCase` and `C:\CFD\logs\myCase` . The first directory stores your working copy of the case configuration; the second stores downloaded logs. Establish one authoritative location for edits to avoid conflicting copies.

3. FileZilla: connect with SFTP and transfer files

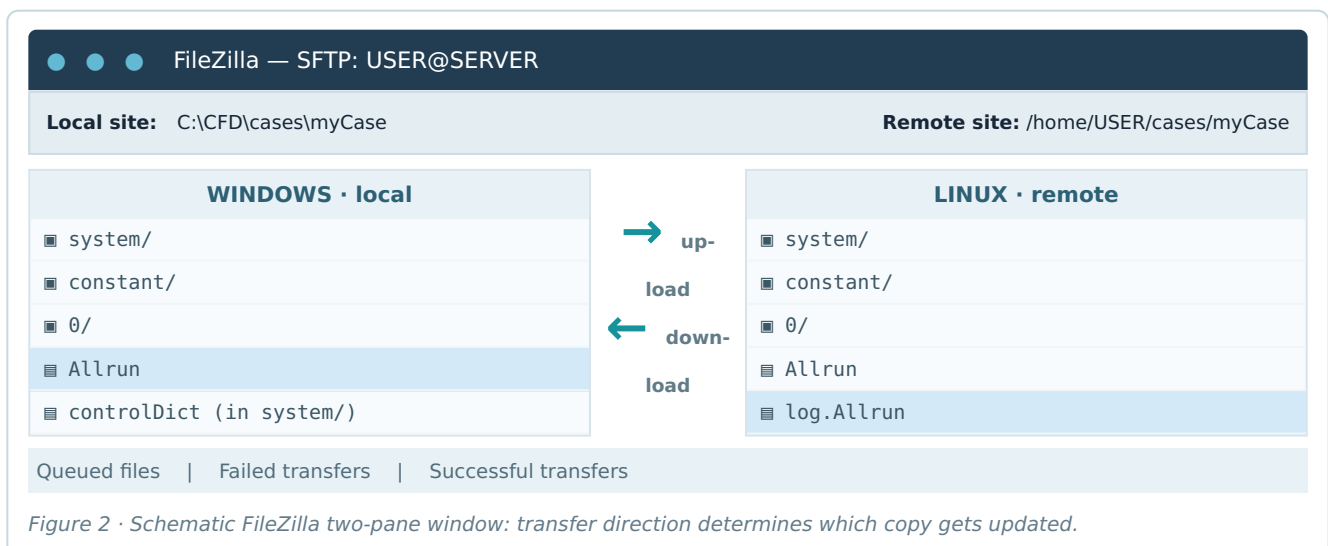


Figure 2 · Schematic FileZilla two-pane window: transfer direction determines which copy gets updated.

1. In FileZilla choose **File → Site Manager → New site**.
2. Set **Protocol** to **SFTP - SSH File Transfer Protocol** (**not** unencrypted FTP).
3. Set **Host** to `SERVER` and **Port** to `22`, unless configured otherwise.
4. Under **Logon Type**, select **Interactive / Ask for password** (depending on FileZilla version), or **Key file** if key-based authentication has been configured. Set **User** to `USER` .

- Click **Connect**. On first connection, compare the SSH host-key fingerprint with one supplied by your administrator. Do not blindly accept an unknown fingerprint.
- In the **left** pane, open `C:\CFD\cases\myCase`; in the **right** pane, navigate to `/home/USER/cases/myCase` (or the actual path assigned to you).
- Drag **left to right** only the files you want to upload. Drag **right to left** the logs you want to inspect. Review the **Failed transfers** and **Successful transfers** tabs.

Mind the transfer direction: uploads can overwrite server files, and downloads can overwrite local files. Back up dictionaries before substantial edits; when FileZilla reports a conflict, compare timestamps and file sizes before replacing anything. A log that is still being written is only a snapshot when downloaded, not the final log.

Suggested case layout:

```
myCase/
├── 0/                # initial and boundary conditions
├── constant/         # physical properties and mesh
├── system/           # controlDict, fvSchemes, fvSolution...
├── Allrun            # optional run script
└── log.*             # outputs from utilities and solvers
```

Do not automatically transfer large time directories (`0.1/`, `0.2/`, ...), `processor*/`, or the entire `postProcessing/` directory when using a client/server workflow: ParaView reads the data on Linux. FileZilla can also remember default local and remote directories in the site's advanced settings.

4. Notepad++: edit OpenFOAM dictionaries and Linux scripts safely

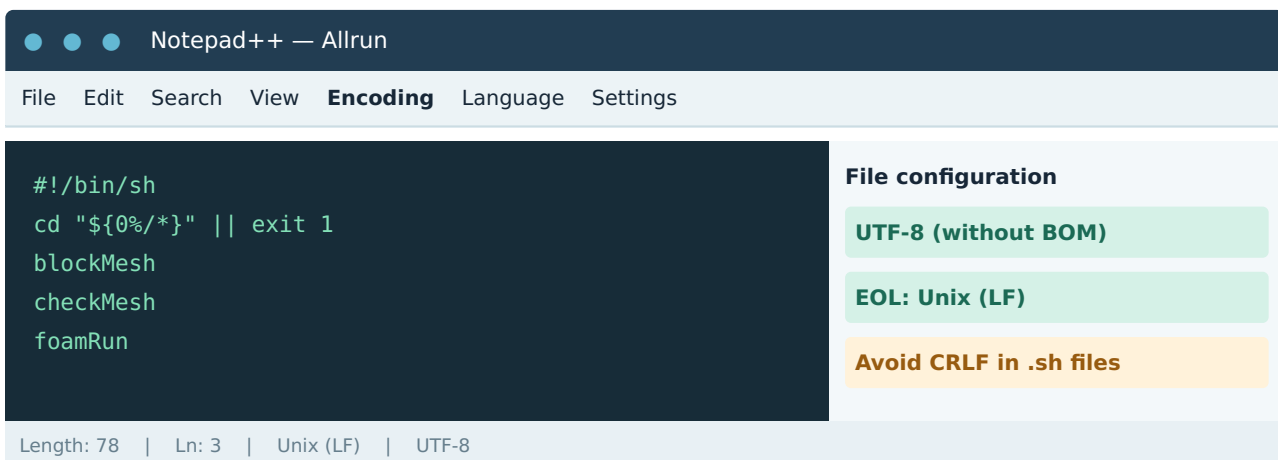


Figure 3 · Schematic Notepad++ screen: encoding and line endings matter for Linux scripts, not just their text.

OpenFOAM files are generally **plain text**, even when they have no filename extension. Notepad++ works well with `controlDict`, `fvSchemes`, `fvSolution`, `blockMeshDict`, field files under `0/`, and `Allrun/Allclean` scripts.

- Select **Encoding** → **UTF-8** without BOM. Avoid UTF-16 and UTF-8 with BOM for shell scripts and configuration files.
- Select **Edit** → **EOL Conversion** → **Unix (LF)**, especially for `Allrun` and `.sh` files. In **Settings** → **Preferences** → **New Document**, you can choose Unix line endings and UTF-8 as defaults.
- Use **Language** → **C/C++** for approximate syntax highlighting of dictionaries, or **Shell** for scripts; highlighting does **not** validate OpenFOAM syntax.
- Preserve **case-sensitive** filenames: Linux distinguishes `controlDict` from `controldict`. Do not let the editor append `.txt` to names such as `fvSolution`.

- Avoid modifying server-side dictionaries while a simulation is using them unless you understand the consequences. Prepare a revision, transfer it, and **then** restart the appropriate procedure in a controlled way.

Example excerpt from `system/controlDict`:

```
application    foamRun;
startFrom      startTime;
startTime      0;
endTime        100;
deltaT         1;
writeControl    timeStep;
writeInterval  10;
```

This excerpt is **illustrative only**: do not replace the whole `controlDict` of a real case with it. In OpenFOAM 14, `foamRun` uses a solver module selected from the case configuration. Read the dictionaries already present before changing the run setup.

Common error: `./Allrun: /bin/sh^M: bad interpreter`. This almost always indicates Windows CRLF line endings in a shell script. Convert to LF in Notepad++ and upload it again. If unsure, check `file Allrun` on Linux.

5. PowerShell: log in to Linux and navigate directories

PowerShell acts as your **remote terminal**. After running `ssh`, you are at a prompt for the Linux server's shell: subsequent commands are **Linux commands**, not PowerShell commands.

```
# From Windows PowerShell:
ssh USER@SERVER
# SSH on a port other than 22:
ssh -p 2222 USER@SERVER
```

```
# After logging in to Linux:
whoami          # current account
pwd             # current directory
ls -lh          # readable listing with file sizes
ls -la          # also show hidden files
cd ~/cases      # go to your cases directory
cd myCase       # enter your case
cd ..           # parent directory
cd ~            # return to your home directory
mkdir -p ~/cases/myCase # create a directory if necessary
find . -maxdepth 2 -type f # list files to depth 2
exit            # close SSH; return to PowerShell
```

If `foamVersion` is not found, the OpenFOAM environment may not have been loaded into the shell. For a Foundation installation under `/opt/OpenFOAM-14`:

```
source /opt/OpenFOAM-14/etc/bashrc
foamVersion
command -v blockMesh
command -v foamRun
```

The installation path is an **example**, not a promise: the actual location depends on your server. Some servers load the environment automatically at login. Use `command -v` and the installed programs' `--help` where appropriate to check what is available.

6. Run a case without becoming a shell expert

First run and mesh checks

```
cd ~/cases/myCase
ls -l Allrun system/controlDict
sed -n '1,100p' Allrun          # READ the script before running it
blockMesh 2>&1 | tee log.blockMesh
checkMesh 2>&1 | tee log.checkMesh
```

Run `blockMesh` **only if** the case uses a `blockMeshDict` and its mesh is meant to be generated with this utility. `Allrun` may already handle meshing; avoid running unnecessary or duplicate steps, and follow the case's prescribed sequence. `checkMesh` examines an existing mesh.

Run the `Allrun` batch script

```
cd ~/cases/myCase
chmod u+x Allrun          # if executable permission is missing
./Allrun 2>&1 | tee log.Allrun # show output and save it to a log
```

Not every case has an `Allrun`: it is a convention, not a requirement. Some cases use a different script, or run `foamRun` directly.

Keep the simulation running if your connection drops: `tmux` (recommended)

```
tmux new -s cfd_myCase
cd ~/cases/myCase
./Allrun 2>&1 | tee log.Allrun
```

To **detach without stopping the job**, press `Ctrl+B`, release, then `D`. To reconnect after a new SSH login:

```
tmux ls
tmux attach -t cfd_myCase
```

`Ctrl+C` **interrupts** the active command, so do not use it as a way to disconnect. Typing `exit` in an ordinary shell may end the session: detach from `tmux` first. If the case has no run script, launch the appropriate executable (`foamRun` in many Foundation 14 cases) only after checking its configuration.

Without `tmux` (less interactive): `nohup ./Allrun > log.Allrun 2>&1 < /dev/null &` is an option if `tmux` is unavailable and the script supports non-interactive execution.

7. Monitor execution and logs from the terminal

```
cd ~/cases/myCase
ls -lh log*
tail -n 40 log.Allrun          # last 40 lines
tail -f log.Allrun             # follow new lines (Ctrl+C stops tail only)
grep -n -E 'FOAM FATAL|Error|error|Courant|Solving for|Time =' log.Allrun
ps -u "$USER" -o pid,etime,cmd  # processes owned by your account
```

A process listed by `ps` may still be stalled or producing an invalid solution; it is **not** proof of convergence. Likewise, an `End` message is not by itself validation of physical results. Many cases have individual logs such as `log.blockMesh`, `log.checkMesh`, or `log.foamRun`. Inspect the files the actual workflow produces.

Typical OpenFOAM logs contain simulation time (`Time =`), field residuals (`Solving for ... Initial residual ... Final residual ...`), `Courant Number`, elapsed/execution time, and error messages. The values and acceptance criteria depend on the model: **small residuals alone do not establish physical correctness**.

8. Download logs with FileZilla and analyze them on Windows

1. In FileZilla, navigate in the **right** pane to the **remote** case directory. Open `C:\CFD\logs\myCase` in the left pane.
2. Look for `log.Allrun`, `log.foamRun`, `log.checkMesh`, or other `log.*` files and select those you need.
3. Drag the files **right to left**, or choose **Download**. Check the transfer queue for errors.
4. If you want to archive a run, rename **only the Windows copy**, for example `log.foam-Run_2026-10-16_runA.txt`.
5. Open small or medium logs in Notepad++. For very long logs use **klogg**, which supports fast searching, regular expressions, highlighting, and navigation between matches.

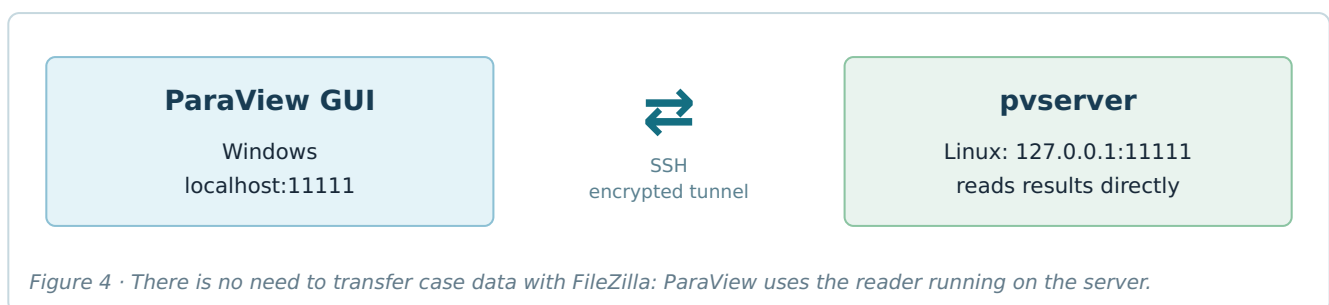
Useful search terms in klogg or Notepad++: `FOAM FATAL`, `Floating point exception`, `Courant Number`, `Time =`, `Initial residual`, `Final residual`, `ExecutionTime`, `End`.

Example regular expression for solver-status lines: `Solving for|Initial residual|FOAM FATAL|Courant Number`.

To **plot residuals**, extract a table with two columns (iteration/time and residual) using a short Python script, or use existing data produced by OpenFOAM function objects if the case provides them. Do not expect a raw solver log to import into Excel as a consistent rectangular table: it requires parsing first.

Practical rule: FileZilla is not a remote `tail -f`. For live monitoring, use PowerShell + SSH; for more detailed analysis, download a **copy** of the log. Repeated downloads of a growing log can yield incomplete snapshots, and you may have to refresh the local file manually.

9. ParaView in client/server mode: how it works



The **ParaView GUI on Windows** controls a **pvserver process on Linux**. `pvserver` opens meshes and results in the Linux filesystem, performs data reading and filtering, and sends the necessary information to the Windows client. You do not need to download every solution field. Once connected, **File → Open** browses the **server's files**, not your local `C:` drive. ParaView's documentation warns that `pvserver` does not authenticate clients and that its protocol is not encrypted by default. Use an **SSH tunnel** rather than exposing TCP port 11111 directly.

Recommended method: one SSH terminal with a tunnel

Step A. From Windows PowerShell, provided TCP port 11111 is available locally:

```
ssh -L 11111:127.0.0.1:11111 USER@SERVER
```

This command opens an SSH terminal and simultaneously forwards Windows `localhost:11111` to the server's `127.0.0.1:11111`. **Keep this PowerShell window open** throughout the visualization session.

Step B. In the Linux shell opened by that SSH connection:

```
command -v pvserver
pvserver --server-port=11111 --bind-address=127.0.0.1
```

`--bind-address` is available in ParaView 5.13.x and limits listening to the server's loopback interface. If `pvserver` is not on `PATH`, locate its installed executable (for example `/opt/paraview-5.13.2/bin/pvserver`) and use that path. Wait for a message similar to `Waiting for client...`.

Step C. In ParaView for Windows select File → Connect → Add Server:

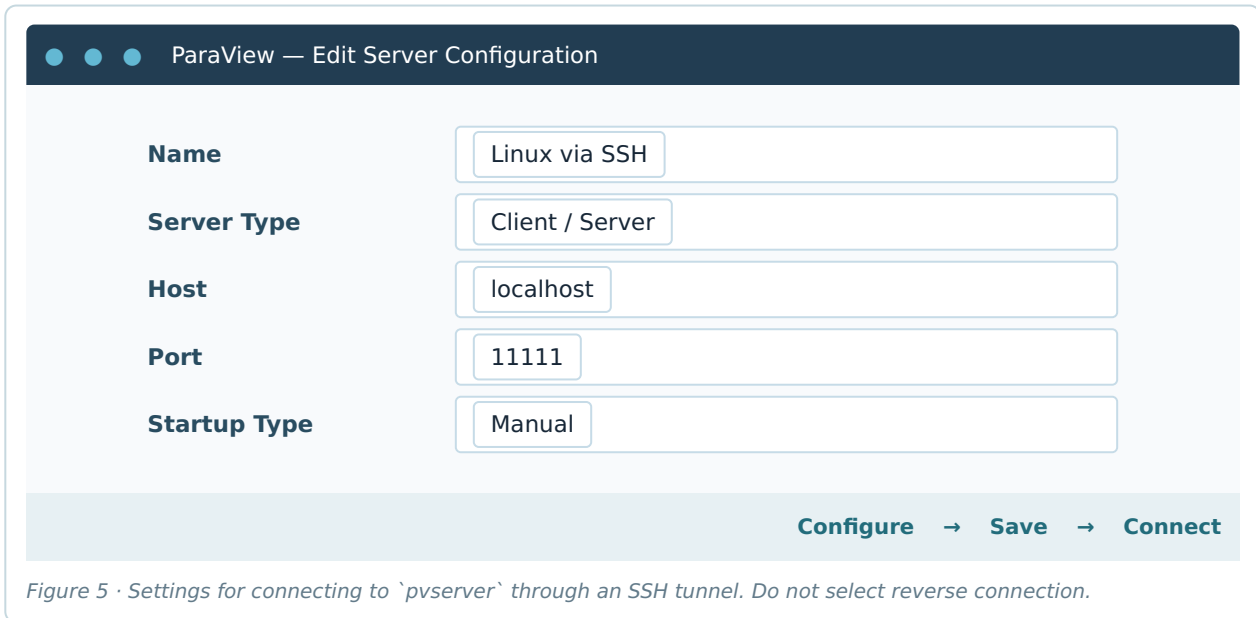


Figure 5 · Settings for connecting to `pvserver` through an SSH tunnel. Do not select reverse connection.

Setting	Value
Name	Linux via SSH
Server Type	Client / Server (not reverse connection)
Host	localhost
Port	11111
Startup Type	Manual

Save the settings and click **Connect**. A successful connection appears in Pipeline Browser as something like `cs://localhost:11111` rather than the local-only `builtin` server.

Open an OpenFOAM case on the server

In a Linux shell, from the case directory:

```
cd ~/cases/myCase
touch myCase.foam
```

`touch` creates an **empty placeholder file**, not a copy of the results. With ParaView connected to the server, choose **File → Open** and browse to the Linux directory `/home/USER/cases/myCase`. Select `myCase.foam`, choose the **OpenFOAM** reader, and press **Apply**. Then select fields, regions or boundary patches and the timestep to display. Reader availability depends on the components installed with `pvserver`.

If the simulation is decomposed into `processor0/`, `processor1/`, etc., check whether the reader supports **Decomposed Case** mode; alternatively use `reconstructPar` when appropriate. Do not reconstruct a large case just to open it without first checking the reader's options.

Finish the session

Choose **Disconnect** in ParaView, then stop `pvserver` in the Linux terminal with `Ctrl+C` if it is still running; type `exit` to close SSH. Once the tunnel closes, the Windows GUI cannot reconnect until it is reopened.

If the window is empty or OpenGL errors occur: installing OpenFOAM on Windows is not the solution. Check whether `pvserver` has an appropriate graphics backend (X11, EGL, or software rendering, depending on its build and server), inspect its terminal messages, and consult your administrator about the configured rendering mode. Having no active Linux display does not automatically mean every `pvserver` build can render without additional configuration.

10. End-to-end practice: Windows to simulation results

1. Open `C:\CFD\cases\myCase\system\controlDict` in Notepad++; make an agreed-upon change and save with **UTF-8 without BOM** and **LF** line endings.
2. Connect FileZilla via SFTP, compare the files, and upload **only** the edited `controlDict` into the correct remote `system/` directory.
3. From PowerShell run `ssh USER@SERVER`, then `cd ~/cases/myCase`. Check the location with `pwd` and `ls system/controlDict`.
4. Read `Allrun` before running it. Start a `tmux` session, launch the procedure, and save `log.Allrun`.
5. Watch the log with `tail -f` (in another SSH session) and check for critical errors.
6. Download the log with FileZilla to `C:\CFD\logs\myCase`; open it in klogg and search for errors and residual information.
7. When computation finishes, start a **new** SSH session with port forwarding, launch `pvserver`, connect ParaView for Windows, and open the `.foam` file directly from the Linux filesystem.
8. Save images, tables, and analysis notes on Windows. The CFD results stay on the server unless you explicitly export files.

11. Common problems and diagnostics

Symptom	What to check / do
<code>ssh: command not found</code> on Windows	Enable the Windows OpenSSH Client feature; reopen PowerShell.
SSH/SFTP <code>Permission denied</code>	Check username, password/key, permissions, and the host-key fingerprint.
FileZilla opens the wrong directory	Check the full remote path and permissions; <code>/</code> is not the same as your home directory.
<code>foamRun / blockMesh: command not found</code>	Load the OpenFOAM environment or check the installation and <code>PATH</code> .
<code>./Allrun: Permission denied</code>	Inspect <code>ls -l Allrun</code> and, where appropriate, use <code>chmod u+x Allrun</code> .
<code>bad interpreter: /bin/sh^M</code>	Change <code>CRLF</code> to <code>LF</code> and upload the script again.
SSH disconnects and the simulation stops	Run in <code>tmux</code> , and detach before closing the terminal.
ParaView cannot connect	Is <code>pvserver</code> running? Is the SSH tunnel open? Do the ports match? Are versions compatible?
<code>Address already in use</code>	Port 11111 is in use on Windows or Linux; release it or configure matching alternative ports.
<code>.foam</code> file is missing	Run <code>touch myCase.foam</code> in the correct directory and check that you are browsing server-side files.

Symptom	What to check / do
Missing fields or reader failure	Check case format, mesh, decomposed data, and the reader available to <code>pvserver</code> .
Black view / OpenGL error	Check rendering capabilities and <code>pvserver</code> messages instead of randomly changing X11/GPU settings.

12. Printable quick-reference sheet

```

WINDOWS / POWERSHELL
ssh USER@SERVER
ssh -L 1111:127.0.0.1:1111 USER@SERVER # ParaView SSH tunnel

LINUX / CASE
pwd ; ls -lh ; cd ~/cases/myCase
foamVersion
sed -n '1,100p' Allrun
blockMesh ; checkMesh # ONLY if the case needs them
tmux new -s cfd_myCase
./Allrun 2>&1 | tee log.Allrun
# Detach: Ctrl+B, then D
# Reattach: tmux attach -t cfd_myCase
tail -n 40 log.Allrun
tail -f log.Allrun

LINUX / PARAVIEW (IN THE SSH SESSION WITH PORT FORWARDING)
pvserver --server-port=1111 --bind-address=127.0.0.1
cd ~/cases/myCase ; touch myCase.foam

WINDOWS / PARAVIEW
File > Connect > Add Server > Client/Server
Host localhost ; Port 1111 ; Startup Manual
File > Open > /home/USER/cases/myCase/myCase.foam

```

13. Official resources and further reading

- OpenFOAM Foundation, **OpenFOAM 14** and getting started: <https://openfoam.org/version/14/> and <https://openfoam.org/download/14-ubuntu/>.
- ParaView, **Remote and Parallel Visualization, 5.13.2** manual: <https://docs.paraview.org/en/v5.13.2/Reference-Manual/parallelDataVisualization.html>.
- ParaView, **Loading Data, 5.13.2** manual: <https://docs.paraview.org/en/v5.13.2/UsersGuide/dataIngestion.html>.
- ParaView, 5.13 release notes (`--bind-address`): https://www.paraview.org/paraview-docs/v5.13.2/cxx/md__builds_gitlab-kitware-sciviz-ci_Documentation_release_ParaView-5_813_80.html.
- FileZilla Wiki, **Site Manager** and SFTP authentication: https://wiki.filezilla-project.org/Site_Manager.
- Notepad++ User Manual, **New Document / encoding / line endings**: <https://npp-user-manual.org/docs/preferences/>.
- Microsoft Learn, **OpenSSH on Windows**: https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse.
- klogg, **log viewer**: <https://github.com/variarklogg>.

Reference screenshots available online: FileZilla SFTP (https://www.ionos.com/help/fileadmin/images/hosting/ftp/2938/EN/2938_EN_transfer_files_filezilla_linux_05.png), Notepad++ Unix LF menu (<https://i.sstatic.net/sndcy.jpg>), ParaView server configuration in the Kitware guide (https://docs.paraview.org/en/v5.11.1/_images/serverconfig.png). Screenshots may show versions or interface arrangements different from those currently installed.

About the illustrations: The figures in this guide are **schematic reconstructions** of the important controls, not screenshots of any specific computer. Interfaces may vary with software version or language; the reference links above also lead to original application screenshots.