

FieldView

Installation Guide

This document details the installation instructions for **FieldView**. Please read thoroughly to properly determine your system requirements and successfully install the software.

Overview of FieldView Installation, p. 3

Certified and Supported Platforms, p. 4

Installing FieldView (Basic Installation), p. 7

Linux & MAC, p. 7

Windows, p. 8

Python Installation (Optional), p. 10

Installing FieldView Servers (Optional), p. 11

Installing FieldView Parallel on Linux / Mac (Optional), p. 12

Installing FieldView Parallel on Windows (64-bit), p. 13

Testing & Configuring FieldView Servers, p. 14

Step 2 – Unpack & Install the server program(s), p. 17

Step 3 – Setup a Server Configuration File, p. 18

The fvrunsrv Wrapper Program, p. 24

Additional Comments Regarding FieldView Servers, p. 25

FieldView Licensing, p. 28

Concurrent User (Floating) Licensing, p. 28

Single System (Node-Locked) Licensing, p. 29

FieldView Parallel Licensing, p. 29

FieldView Batch-Only Licensing, p. 31

Using Batch-Only Licensing: Creation of transient animations, p. 32

FieldView Licensing for Virtual Environments, p. 33

Licensing via Server, p. 35

Obtaining a FieldView License, p. 38

Linux & MAC, p. 38

Windows, p. 38

FieldView License Installation Tasks on Linux, p. 39

Configuring a Concurrent User (Floating) License, p. 39

Configuring a Single System License, p. 40

Running the License Server on Linux, p. 41

fv_server Licensing Tool, p. 41

FieldView License Installation Tasks on Windows, p. 43

Configuring a Concurrent User (Floating) License, p. 43

Configuring a Single System License, p. 43

Running the License Server on Windows, p. 44

LMTOOLS, p. 46

FieldView and Other Software Using FLEXlm, p. 49

Using a Combined License File on Linux, p. 49

Using a Combined License File on Windows, p. 50

port@host Licensing, p. 51

Setting the `ILIGHT_LICENSE_FILE` environment variable, p. 51

Licensing Issues and Problems, p. 51

Post-Installation Tasks, p. 56

Configuring your machine (Linux only), p. 56

Modifying Your Print Script (Linux only), p. 56

Locating the FieldView Tutorial Files (Windows and Linux), p. 56

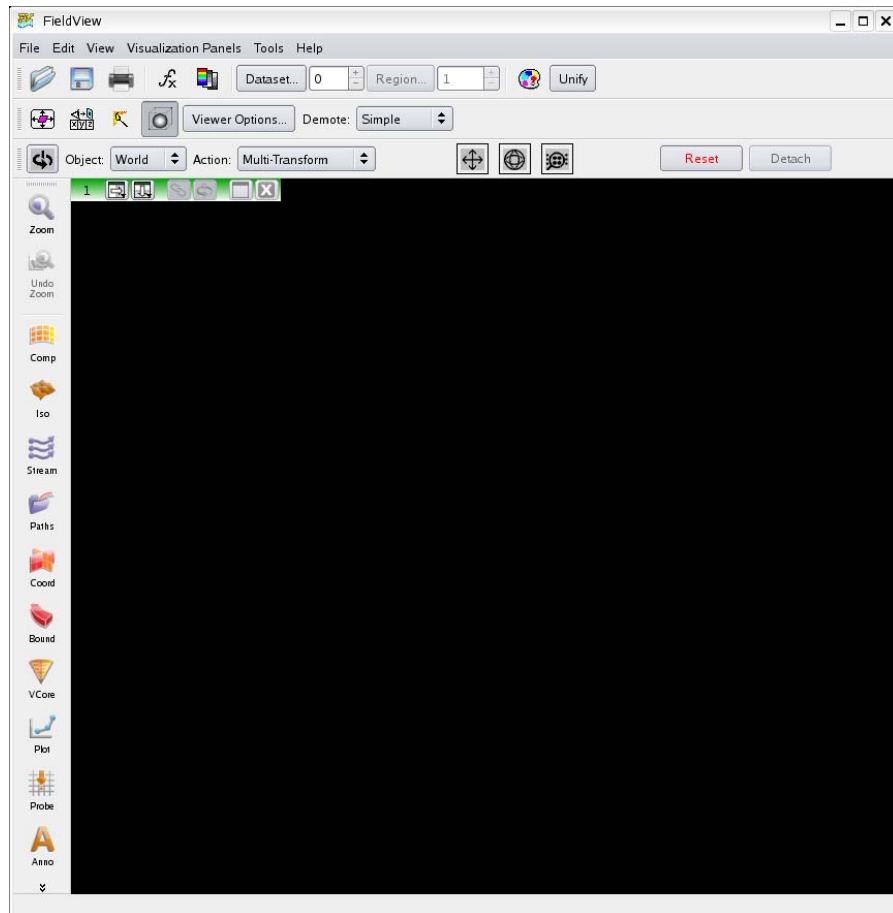
Locating the KIVA Translator File (Windows and Linux), p. 57

Overview of FieldView Installation

The installation of **FieldView** consists of the following general steps:

1. Review the supported platforms and system requirements
2. Install the **FieldView** executable and/or server programs
3. Contact **FieldView CFD** to obtain passwords (passwords@fieldviewcf.com)
4. Complete your license administration tasks
5. Complete your post installation tasks

Following a successful installation and startup, **FieldView** should look like this:



The rest of this **Installation Guide** provides detailed information on how to install and customize **FieldView** to meet your specific needs.

Certified and Supported Platforms

Certified platforms are defined as follows:

The **FieldView CFD** team has performed a successful, comprehensive test on this platform.

Supported platforms are defined as follows:

Users have successfully installed and used the product on this platform.

We strongly recommended that you check the table below to see whether **FieldView** will run properly on your operating system BEFORE you attempt to install it. All certified and supported platforms provided in the following table are valid on or before December 2020. See www.fieldviewcfd.com to find the most up to date platform coverage information. .

Certified & Supported FieldView Platforms					
System	Platform	OS	Client	Server	Parallel
APPLE MAC	INTEL	macOS 10.9 through 10.13 (64-bit)	FV 20 SUPPORTED		
		macOS 10.6 through 10.8 (64-bit)	FV 20 CERTIFIED		
System	Platform	OS	Client	Server	Parallel
Linux (64-bit)	AMD64/EM64T/Xeon-64	CentOS 6.5 (kernel 2.6)	FV 20 SUPPORTED		
		CentOS 6.9 (kernel 2.6)	FV 20 CERTIFIED		
		CentOS 6.10 (kernel 2.6)	FV 20 CERTIFIED		
		CentOS 7 (kernel 3.10)	FV 20 SUPPORTED		
		CentOS 8 (kernel 3.10)	FV 20 SUPPORTED		
		Red Hat Enterprise Linux 6.5 (kernel 2.6)	FV 20 SUPPORTED		

Linux (64-bit)	AMD64/EM64T/Xeon-64	Red Hat Enterprise Linux 6.7 (kernel 2.6)	FV 20 SUPPORTED		
		Red Hat Enterprise Linux 7.4 (kernel 3.10)	FV 20 SUPPORTED		
		Red Hat Enterprise Linux 8 (kernel 4.18)	FV 20 SUPPORTED		
Linux (64-bit)	AMD64/EM64T/Xeon-64	Ubuntu 20.04 (kernel 5.4)	FV 20 SUPPORTED		
		Ubuntu 18.04 (kernel 5.4)	FV 20 SUPPORTED		
Linux (64-bit)	AMD64/EM64T/Xeon-64	Red Hat Enterprise 5.5 (kernel 2.6)	Platform coverage dropped at FV20		
		CentOS 5.5 (kernel 2.6)	Platform coverage dropped at FV20		
System	Platform	OS	Client	Server	Parallel
Windows (64-bit)	AMD64/EM64T/Xeon-64	Windows 7 Pro SP1	FV 20 CERTIFIED		
		Windows 8	FV 20 SUPPORTED		
		Windows 8.1 Pro	FV 20 SUPPORTED		
		Windows 10	FV 20 CERTIFIED		
		Windows HPC Server 2008	FV 20 SUPPORTED		
		Windows HPC Server 2012	FV 20 SUPPORTED		
		Windows Server 2008 Standard	FV 20 SUPPORTED		
		Windows Server 2012 Standard	FV 20 SUPPORTED		
System	Platform	OS	Client	Server	Parallel
FUJITSU M5000,M9000	UltraSparc	Open Solaris 10 (64-bit)	Not avail.	Platform coverage dropped at FV18	
SUN	UltraSparc	Solaris 10 (64-bit)	Not avail.	Platform coverage dropped at FV18	

Linux (32-bit)	Pentium/ Xeon	Red Hat Enterprise 5.5 (kernel 2.6)	Platform coverage dropped at FV17
		CentOS 5.5 (kernel 2.6)	Platform coverage dropped at FV17
IBM	RS/6000	AIX 4.3 (32-bit)	Platform coverage dropped at FV17
Windows (32-bit)	Pentium/ Xeon	Windows Vista, 7, 8	Platform coverage dropped at FV16.1
SOLARIS	UltraSparc	7,8,9	Platform coverage dropped at FV13.2
SGI Altix	Itanium	ProPack3 SP6 (64-bit)	Platform coverage dropped at FV14

Installing FieldView (Basic Installation)

Installation of the **FieldView** Client (including packaged servers) will require about 1.9G of disk space. Separate packages containing only the **FieldView** Servers for a given platform are also provided, and these will require between 30M to 104M of disk space.

The general instructions presented here are expected to work properly on platforms running our certified or supported operating systems; however certain platform-specific issues may be encountered. If you encounter problems during installation please contact support@fieldviewcfd.com.

To download **FieldView** from the web, please log in to the **FieldView** Customer Center at my.fieldviewcfd.com and follow the installation instructions provided there.



WARNING: Do NOT install **FieldView** in a directory containing international (multibyte / UTF-8) characters.

Linux & MAC

To install **FieldView** from the downloaded archive, follow these instructions:

1. Navigate to the location where you wish to install **FieldView**. If there is a previously installed version of **FieldView** in this directory, you must either delete or rename the `fv` subdirectory. Do not attempt to merge different versions of the software together.

Important: Before deleting the directories you should back up any files in these directories that you may have modified or added. For example, you may have changed the print script: `fv/bin/fv_to_printer.sh`.

2. Run the shell script `installfv.sh`. This can be done by typing the command:

```
./installfv.sh
```

Linux users are prompted for the type of machine to install (i.e. the type of machine on which you will run **FieldView**) and the directory in which to install the software. This command installs the directory `/fv` at the directory level you specified. If you receive an error message during the install, please contact **FieldView CFD**.

The installation script creates a **FieldView** startup script in the subdirectory `/fv/bin` of the current directory. This startup script is called `fv`. The startup script contains an environment variable called `FV_HOME` that points to the subdirectory `fv` in the directory where **FieldView** is installed. There are a number of other files which use the explicit value of `FV_HOME` (the installation path), including python files, so the **FieldView** installation should never be moved. Instead, reinstall **FieldView**.

After **FieldView** has been installed, you will see the following prompt:

```
Do you want to install the floating license manager (answer y for yes)?
```

You should answer `y` to this question for single systems (node locked) and floating licenses installed on Linux versions of **FieldView**.

If you answer `y` to installing the floating license manager, you are asked which platform will run the license manager. The floating license manager does not need to run on the same machine, or even the same machine type, as **FieldView**. However, the machines must be on the same network.

If you have installed the floating license manager, the floating license script, called `fv_server`, is installed in the subdirectory `/fv/bin`. If you wish to change the location of this script, you must edit it, changing the environment variable `FV_HOME` in the script, to point to the directory where **FieldView** is installed.

If you are not installing optional components such as servers for different platforms, Python or **FieldView** Parallel, you may proceed directly to Licensing

Known Issues

When attempting to start **FieldView** on systems with newer or compact Linux distributions, you may encounter an error similar to this:

```
/opt/fv13/fv/bin/fvbin: error while loading shared libraries:  
libstdc++.so.5: cannot open shared object file: No such file or  
directory
```

To resolve this problem, you will need to install the appropriate GNU C++ run time libraries. For users running CentOS 6.2 or RHEL 6, installing `compat-libstdc++-33` is known to resolve this problem. For users running OpenSUSE 11.0, `compat-libstdc++` is known to resolve this problem.

Windows

You must install **FieldView** (and all software) as either Administrator, or as a user that has Administrator privileges, otherwise the **FieldView** installation will fail.

1. Run the program `fv20_win64_setup.exe` located in downloaded archive. For 64-bit systems download `fv20-win64.zip`
2. You are prompted for the directory to install the software. By default, it is installed (for 64-bit) in the directory:

C:\Program Files\FieldView CFD\FVWIN20

We *strongly recommend* that you accept this default location.

After installation is complete, a Start menu program group will exist which contains the **FieldView** icon, licensing tools, and links to several pdf documentation files.

If you are not installing optional components such as servers for different platforms, Python or **FieldView** Parallel, you may proceed directly to Licensing

Python Installation (Optional)

FieldView is designed to use Python versions 2.7 or 3.1. Python version 2.7 is included in the `'python'` subdirectory of the **FieldView** distribution for Linux and Windows. This will not conflict with any pre-existing Python installations, and Python is not required to run **FieldView**.

To run **FieldView** with Python 2.7, no further installation is required.

Selecting/Linking to your Python Version

If you choose to use a Python installation other than 2.7, it must be the other version we support (3.1), and you will need to do the following in order to have **FieldView** work correctly with Python:

1. Set the `PYTHONHOME` environment variable to allow **FieldView** to locate your Python installation.
2. Set the environment variable `FV_PYTHON_VER=3` (if using Python version 3), since version 2.7 is the default.

Note that OS X **FieldView** cannot run Python 3.

On Linux and Mac systems, to provide full flexibility, you have the option of redirecting **FieldView** to an alternative Python release by changing the link which points to the **FieldView** Python installation. This link is located at:

```
$FV_HOME/lib/<platform>/libfv_python_link.so
```

where `$FV_HOME` is the path for `'fv'` as installed on your system.

By default, **FieldView** uses Python 2.7 libraries. However, **FieldView** can also be configured to work with Python 3. Due to fundamental differences between Python 2.7 and 3, information on which Python interface library to use must be provided to **FieldView**. By setting `FV_PYTHON_VER` to 3, the Python 3 interface library is used.

To use **FieldView** with Python 3, Python 3 libraries also need to be present. Python 3 is currently not supplied with **FieldView**. Running Python 3 on a Linux platform requires the following steps:

Create the link `$FV_HOME/lib/<platform>/libfv_python3_link.so` pointing to the Python 3 library, i.e.:

```
ln -s /usr/local/python3/lib/libpython3.1.so  
$FV_HOME/lib/<platform>/libfv_python3_link.so
```

And, as noted above you will need to set the following environment variable before starting **FieldView**.

```
export FV_PYTHON_VER=3
```

On Windows systems, you will need to make sure that the `python31.dll` library is in your `PATH`. This should already be true since the standard Python installation for Windows platforms places the `python31.dll` in the `WINDOWS\system32` subdirectory.

Next, you will need to edit the `fv.bat` file found in the 'bin' subdirectory of the **FieldView** installation:

- In the line

```
SET PYTHONHOME=%FVINSTALL%\python
```

replace `%FVINSTALL%\python` with the path to the Python 3.1 installation on your system; by default this is `"C:\Python31"`
- After the `SET PYTHONHOME` line, add the following:

```
SET FV_PYTHON_VER=3
```

Note that using both Python 2.7 and 3 simultaneously within different **FieldView** processes is allowed.

Installing FieldView Servers (Optional)

Each client platform includes the corresponding **FieldView** server software for Client-server and Parallel operation. **FieldView** Server programs for remote and parallel operation are also available for any combination of **FieldView** Client and **FieldView** Server platforms. If you wish to run Client-Server between two different machine types (for example, a Windows desktop and a Linux server), then you will also need to download and install the server executable for your target server platform(s).



Note: The basic **FieldView** license includes an option to run **FieldView** Parallel at no additional cost on up to eight worker processes. For local parallel operation, the **FieldView** Parallel servers are automatically installed and no additional steps are required.

During the **FieldView** installation on Windows, archives for all **FieldView** Server platforms are installed in the servers subdirectory by default.

It is possible to install the server programs by downloading and uncompressing the appropriate platform archive to the desired location. For additional details, refer to [Step 2 – Unpack & Install the server program\(s\) on page 17](#). Further steps (described in [Step 3 – Setup a Server Configuration File on page 18](#)) are needed to make Client-Server operation functional.

Installing FieldView Parallel on Linux / Mac (Optional)

There are two important limitations to be aware of concerning **FieldView** Parallel. The first is that you can only read either PLOT3D, OVERFLOW-2, OpenFOAM or **FieldView** Unstructured FV-UNS datasets, or datasets with either single file, multigrid formats, or partitioned file formats that can be read using plugin (or toolkit) readers. ALL data can be read in parallel using PFPR; see [Partitioned File Parallel Reader \(PFPR\)](#) in the **Reference Manual** for more information. The second is that platform support is limited. Refer to [Certified and Supported Platforms on page 4](#).

To run **FieldView** Parallel, you will need to have the following:

1. A system with a supported OS capable of running a parallel application (multiprocessor, multicore)
2. A parallel compatible dataset
3. **FieldView** Parallel Server programs installed

The **FieldView** Parallel and Client-Server programs are of the order of 5MB to 8MB. If you install both the client-server and parallel servers, the total disk space requirements are less than 100MB per platform.

FieldView Parallel distributes and uses a specific version of Open MPI. Users that have a need to use a specific version of MPI may be interested in our premier product, **HPC FieldView**, which allows users to select different versions of MPI (for improved performance, etc.). Please contact **FieldView CFD** for information about **HPC FieldView** (<http://www.fieldviewcfd.com/en/contact>).

After the server programs have been installed, several additional steps must be performed to make **FieldView** Parallel (or **FieldView** Client-Server) operation functional. These steps are the same as those for installation of **FieldView** Client-Server operation. Please consult [Testing & Configuring FieldView Servers on page 14](#) for detailed instructions on how to configure your systems to support Client-Server and **FieldView** Parallel operation.

If you intend to run **FieldView** Parallel on a different system than the one where you have installed the **FieldView** Client (the standard **FieldView** installation) then you must copy the server files to all intended server machine(s).



Please be sure to access **FieldView** servers which match the version of the Client you are using. Unmatched Client-Server combinations are unsupported.

To install a server executable on a server machine, perform the following steps on your server machine:

1. Log in to the server system, and change directory to a location where you would like your **FieldView** Servers to reside. This should be a location visible from your **FieldView** Client installations and may require assistance from your system administrator.
2. Decide which file is correct for your server machine.
3. Uncompress the correct file for your server machine:

```
zcat fvsrv_type_tar.Z | tar xvf -
```

4. Copy the directory tree to its destination directory on the server machine.

Installing FieldView Parallel on Windows (64-bit)

The installed package name for the 64-bit Windows Servers is `fvsrv_win64.zip`, and is installed (as an option) during the **FieldView** for Windows installation process. This can be extracted to a location on your Windows Server. Note that the package is also available to customers as a software download from my.fieldviewcfd.com.

FieldView is installed on Windows platforms with support for shared memory parallel. Cluster parallel (invoked with the use of `-hosts [ANY | WINHPC]`) requires `job.exe`, which is provided for Windows Server platforms, and may require additional package installation such as HPC Pack 2008.

For **FieldView** to start Parallel servers automatically (i.e. autostart) on a remote Windows system, there must be a "remote shell daemon" on that remote Windows system. Microsoft does not support `sshd` or `rshd`. However, `sshd` can be added to a server OS by enabling the "Subsystem for UNIX-based Applications" then downloading additional software from:

<http://www.suacommunity.com/>

(`rshd` may be available as well). Alternatively, you may be able to use the "PsExec" to start remote processes on a server if you are only using Windows clients. PsExec is available from:

<http://www.sysinternals.com>

Note: You do NOT need either `sshd` or PsExec to run local Parallel on a Windows 64-bit system.

Testing & Configuring FieldView Servers

For local servers running on the same system as the client, no additional configuration steps are required. The "Server: Local" option will appear automatically on the Data Input menu.

After you have loaded the server programs, the following additional steps must be performed to make Client-Server operation functional:

1. Test the "relationship" between the client machine and the server machine to ensure that you can start the server process and that the server and client can communicate.
2. Copy the proper server `tar` file to the server machine(s). Unpack the server executable from the `tar` file.
3. Create and test a Server Configuration File that will govern the connection of client and server.

Hints for troubleshooting any problems with steps 1-3 are provided in the following section. All three of the steps listed above must be successfully completed for **FieldView** Client-Server or **FieldView** Parallel to operate properly.

Step 1 – Test the "connection"

We will first test the relationship between the client machine and the server machine. There are two mandatory conditions for **FieldView** Client-Server to work: (a) you must be able to log in from the client to the server in some way, and (b) you must be able to connect from the server to the client using TCP protocol on a specified port. If either of these conditions are not met, then you will *not* be able to use Client-Server. Therefore, it is highly recommended that you test these conditions prior to proceeding with any other steps of the installation. If any of the tests fail, you may need assistance from your system administrator to adjust your configuration.

The first test is to make sure that the client machine and the server machine can communicate with each other. The `ping` command is used for this. (In Windows, you will need to run `ping` from the command prompt).

Test	Command	Expected Result
1	<code>ping server_name</code> or <code>ping server_IP_address</code>	Indication of successful data transfer

If you are not successful with Test 1, you will not be able to use **FieldView** in Client-Server mode, since your client machine cannot make contact with the server. There may be firewalls or routers involved in the communication path which need to be adjusted in order for the two machines to communicate. Contact your system administrator for assistance.

The next test deals with the mechanism that **FieldView** will use to start server processes on the server machine. There are two basic ways to log in to the server machine from the client: via a Telnet-style connection or via `ssh/rsh/plink`. If you can log in via `ssh/rsh/plink`, then you can use Automatic Server start. If `ssh/rsh/plink` cannot be used, but you can log in via a Telnet-style connection, Automatic Server Start can not be used, but you can use Manual Server Start.

Regarding the choice of `ssh/rsh/plink`, the availability of either is dependent on the security policy that governs the client and server machine. In addition, `ssh/rsh/plink` require that files, databases, etc. be set up properly (usually by a system administrator) in order to function properly. Therefore, if you are unsure about any details of using either `ssh/rsh/plink`, it is highly recommended that you talk to a system administrator about your plans before proceeding.

Test procedure - On Linux, only one of 2A or 2B needs to be successful. On Windows, any one of the tests must succeed. (In Windows, you will need to run `rsh/ssh/plink` from the CMD prompt):

Test	Command	Expected Result
2A	<code>rsh server_name -l login_name date</code> Note: <code>server_name</code> can be replaced with the server IP address. Note: Use <code>remsh</code> instead of <code>rsh</code> on HP systems.	Today's date is printed
2B	<code>ssh server_name -l login_name date</code> Note: <code>server_name</code> can be replaced with the server IP address.	
2C	<code>plink.exe server_name -l login_name date</code> Note: <code>server_name</code> can be replaced with the server IP address.	

In any of the tests above, the date must be printed *without* a prompt for a password for the test to be a success. If you are successful with one of the tests above, you can use Automatic Server Start and proceed to Test 4 below. If you are not successful, then you may not be able to use Automatic Server Start. It is possible that there is a problem with the configuration of `ssh/rsh/plink` which can be solved by your system administrator.

In order to test a manual connection, you can use one of either `telnet`, `ssh` or `putty`:

Test	Command	Expected Result
3	<code>telnet server_name</code> or <code>telnet server_IP_address</code> You are prompted for <code>login_name</code> and <code>password</code>	You are logged in to the server machine

If this test passes, then you can use Manual Server Start.

The next test ensures that the client machine is visible from the server machine.

Test	Command	Expected Result
4	<pre>telnet <i>server_name</i> or telnet <i>server_IP_address</i> You are prompted for <i>login_name</i> and <i>password</i>. In the telnet window, type: ping <i>client_name</i> or ping <i>client_IP_address</i>)</pre>	Indication of successful data transfer

This test may fail if there is a router between your client and server. In this case, the IP address needed to contact the client may be different than the IP address that the client thinks it has. For example: A computer connected to a DSL line through a firewall/router may have an IP address of 10.0.0.2 (which is assigned by the router), but on the other side of the router only the router's IP address can be seen.

In this type of configuration, the router/firewall must be configured to allow an incoming connection on this port to be routed to the client machine. Contact your system administration for assistance.

The final test ensures that you can connect back from the server to the client.

Test	Command	Expected Result
5	<pre>On client machine: contest <i>port_number</i> On server machine: telnet <i>client_name</i> <i>port_number</i> Note: Use the IP address returned by contest</pre>	Successful connection message printed in telnet window.

On the client machine, open a terminal window (or a CMD prompt window on Windows systems). Run the `contest` command, which you will find in the `bin` subdirectory of your **FieldView** installation directory. This command will use port number 12345 by default. If you are using a different port when running **FieldView**, supply it as a command-line argument to the `contest` program. For example, to test using port 20000, type the following command:

```
contest 20000
```

On the server machine, test the connection back to the client system using the `telnet` command. You must supply the same port number given to the `contest` program. The `telnet` command should use the client IP address printed by `contest` (the IP address after the client name). If you do not need Automatic Server Start and `telnet`

fails with the client IP address, you can try the client name instead. If no port number was given to `contest`, you *must* supply port number 12345. For example, to connect to the `contest` program on port 20000:

```
telnet client_IP_address 20000
```

If the connection is successful you will get the following message in the `telnet` window:

```
Test connection completed successfully!
```

```
Please disconnect your telnet session now.  
On many telnet clients you can do this by typing CTRL+]  
and entering "quit" at the prompt.
```

To conclude the test, `quit` the `telnet` session. The `contest` program will then shut down.



Note: If you do not `quit` the `telnet` program and instead `kill` the `contest` program, the supplied port may be left in use for 2 - 4 minutes.

If `telnet` just hangs while trying to connect, or if you get a message that the connection is refused, there may be a firewall running on the client system or between the two systems which is blocking connections on this port.

Any firewall will need to be configured to allow connections on your chosen port number in order for **FieldView** to operate in Client-Server mode. Contact your system administration for assistance.

If you are successful with this test, you may proceed to Step 2.

Step 2 – *Unpack & Install the server program(s)*

Server scripts and executables must be installed on the candidate server machine(s). Note that if a **FieldView** client has previously been installed on that server, our standard and parallel servers are automatically installed to permit "local" Client-Server operation, and these components are found in the `./bin` directory of that installation.

To install a **FieldView** server on a Linux server machine, perform the following steps:

1. Download the server package for your platform from fieldviewcfd.com. (Note that **FieldView** Windows installations provide servers for all platforms, and are optionally placed in the `servers` directory during installation.)

```
2. zcat <path_to_your_server_package>/fvsrv<platform>tar.Z | tar
   xvf -
```

Note: If you are installing the **FieldView** servers on a Linux cluster, you should install it on the head node.

Step 3 – Setup a Server Configuration File

Server Configuration Files (or “config files”) are used to help the **FieldView** client identify and connect to the server. In addition, there are features to help you manage multiple projects in different directories on servers. After **FieldView** reads the config files, it uses them to populate the GUI selections for server machines. The name of the config file (minus the `.srv` extension) is what appears on the server selection menu. We provide one default server configuration file for Linux installations. This is called `local.srv` and it will start a server with the name `local` on the same system as the client.

Each server needs at least one config file. When **FieldView** (the client) starts, it reads the server config files. They are read from one of two places in the following order: (1) a directory specified by the environment variable `FV_SERVER_CONFIG_DIR` or (2) `FV_HOME/sconfig`. If `FV_SERVER_CONFIG_DIR` is set, but the directory does not exist, **FieldView** will write a warning message to the console (Windows) or `xterm` (Linux), and then fallback to `FV_HOME/sconfig`. If `FV_HOME/sconfig` does not exist, which means this is not a standard **FieldView** installation, **FieldView** will write a warning message to the console (Windows) or `xterm` (Linux), and then fallback to non-Client-Server.

The contents of the server file are presented in the table below. There are a few rules worth noting:

- All lines are optional, with the specified default values if they are not present.
- The lines may appear in any order.
- The keywords are case-insensitive (although of course some of the values assigned by you to the keywords are not, such as file paths).
- Comment lines are allowed and must be preceded by an “!” character

Keyword	Meaning	Default Value
AutoStart	Server startup method - "true" means Automatic Start, "false" means Manual Start	"true"
ServerName	Identification of the server machine as either a <i>hostname</i> or an <i>IP address</i>	This applies only to automatic startup. An empty or missing server name means use the local server.

ServerType	Selects the Server Type from the following options: [standard shared_mem_parallel shmem cluster_parallel p4]	standard
ServerDirectory	Path to the directory containing the FieldView server executables	This path needs to be explicitly specified; there is no default.
NumProcs	Value passed to MPI for <code>-np</code> option. Value is ignored if the ServerType = standard	default is 4 NOTE: The FieldView "controller" server process is not counted, so the maximum allowable value is one more than the number of workers for which you are licensed.
MachineFile	MPI machine file name including full path as found on the server. This setting is recognized only if ServerType = cluster_parallel or p4. Does not apply to Windows.	\$FV_HOME/bin/ openmpi-1.6.2/ etc/openmpi- default-hostfile
PortForward	Status of whether port forward is to be used, specified as either true false. For port forwarding to work, RemoteShell = ssh or plink. Note: Compression will always be used for the forwarded connection.	false
ServerPort	Specifies the Server Port number for tunneling, For example, ssh -R ServerPort:ClientName:ClientPort	None.
ServerTimeout	Time in seconds that the server will wait for the client to accept the connection. For manual startup, it is the amount of time that you have after manually starting the server to click "Done" on the client pop-up that prompts for manual startup. A value of 0 means "use the default of 60".	60 seconds

UserName	User name that you want to use to log in to the server machine. This is only needed if you wish to log in as someone other than the user name that FieldView is running under on the client machine.	Name associated with effective user ID of client process. Note: This field is ignored for local server.
RemoteShell	The name of the remote shell to be used for Automatic Server Start, specified from [rsh ssh plink]. Note: If RemoteShell = plink, we will not send the -n option.	ssh
P4_RSHCOMMAND	Remote shell command for MPI for cluster parallel, specified from [rsh ssh plink]. Does not apply to Windows.	ssh (rsh was the former default)
ServerProgram	The complete path of the server program (fvsvr) on the server machine. This path should be constructed from the perspective the server machine. That is, the first "/" is the top level directory on the server machine.	This setting is deprecated - use the ServerDirectory setting instead. If this is blank or missing, use the local server.
StartDirectory	The initial path that FieldView's data browsers will open in.	Home directory of remote user on server OR current working directory if CWD is specified.
ClientName	This is the name of the client machine as seen from the server. This is in case the server has a different name for the client than the client has. Can be a <i>hostname</i> or an <i>IP address</i> .	Fully qualified name of client, as seen by client. Note: This field is ignored for local server.
UserDefinedReader	If your reader reads grid geometry and results from separate files, then add the word "Separate". Else, add the word "Combined" for readers which read all data from a single file. If your reader is for structured grids, add the word "Structured". Else, add the word "Unstructured" for unstructured grid readers.	Add the title of your user defined Plug-In Toolkit Reader exactly as it appears in your registration function.

PluginDirectory	If present, servers will use the specified directory for plugin readers and plugin user-defined functions instead of the default.	plugins subdirectory of ServerDirectory
------------------------	---	--

Manual servers have a command-line option:

```
-plugin_dir <pathname of directory containing server plugins>
```

If **PluginDirectory** is present in the server config file of a manual server, **FieldView** will suggest a manual server command line that includes the corresponding `-plugin_dir` option.

NOTE: Local servers, including local parallel servers, may not have an existing server config file. In order to use the server **PluginDirectory** feature, the user must create a server config file if one does not exist.

When a server configuration file is selected from the Data Files input menu, **FieldView** will fill in the correct arguments for the `fvrunsrv` wrapper program, based upon the above settings. When you select a dataset to be read, the `fvrunsrv` program launches the chosen server program (standard, shared memory or cluster), and echoes the current `fvrunsrv` configuration to the console window. This information is useful in case you encounter errors with your server configuration.

If the server is started manually, the port forwarding also has to be set up manually. In other words, you must manually start `ssh` with the appropriate port forwarding arguments, along with manually starting the (browsing) server to use the forwarded port. The PortForward section in the sever config for manual start will trigger a pop-up with the `ssh` port-forwarding command.

Let's consider several server config file examples:

Example 1: Local free parallel

This is a redundant example since Local basic parallel appears on the Data Files pull-down menu. The server config file (`.srv`) in this case would contain a single line. Since this is running locally, there is no need to specify a different **ServerDirectory**.

```
ServerType: shared_mem_parallel
```

There are several implicit defaults being used in this case. First, since we have not specified a **ServerDirectory**, we will run using the locally installed server programs. Next, since we have not specified a **NumProcs** value, we will default to four.

Example 2: Setting the Current Working Directory

Use 'CWD' for the StartDirectory keyword in the server config file:

```
StartDirectory: CWD
```

Use this (case insensitive) setting to start **FieldView** in your current directory when running **FieldView** in Client-Server or Parallel mode. This is particularly useful if you routinely navigate through many directory levels below your home directory. This setting will take precedence, overriding the saved preferences behavior in **FieldView** (13 or later) which stores and returns you to the last directory visited by default. Within a **FieldView** session, if you navigate away from your current working directory, this is remembered. This server configuration setting only applies to the initial view in your file browser.

Example 3: Remote licensed parallel

In this example, we need to specify the server name, `my_remote_system`, since it is remote and therefore different from the client. We must also explicitly specify where the server programs are located on the remote system. Since we want to run parallel, we must also explicitly choose either the `shmem` (shared memory) or `p4` (cluster) option - in this example, we have chosen to run cluster or `p4` parallel. Finally, we need to pick the number of processes since we have a `pfv32` **FieldView** Parallel license available to us. Names in the example below are notional - you will need to substitute correct information for your `ServerName` and `ServerDirectory`.

```
ServerName: my_remote_system
ServerDirectory: /remote_path/fv/bin
ServerType: cluster_parallel
P4_RSHCOMMAND: ssh
NumProcs: 32
```

In the example above, although we have chosen `cluster_parallel`, we have not specified a machine file. In this instance, we will use the default machine file `openmpi-default-hostfile`, which is typically empty and allows all nodes on the cluster to be utilized.

Example 4: Windows Client running Parallel on Linux

Here, we are trying to read a dataset which is sitting on a Linux system, `my_LINUX_box`, using a **FieldView** client on a Windows system, `my_WINDOWS_box`. The server config file needs to be on the Windows system. We need to specify the name of the server, location of the parallel server files on that system, and also specify a start up directory which lets us browse to the location of the file we want to read. Finally, we want to use `plink` to set up our connection from the Windows system to the Linux system.

```

ServerName: my_LINUX_box
ServerType: shared_mem_parallel
ServerDirectory: /on_LINUX/fv/bin
StartDirectory: /on_LINUX/my_data
NumProcs: 16
!
! -- a comment line -----
!
ClientName: my_WINDOWS_box
RemoteShell: plink

```

In this example, we have explicitly specified all required arguments. No default settings are used. Also note the use of the exclamation mark, '!' to indicate the start of a comment line. As you develop server config files to meet your needs, we recommend the use of comments to help to document your efforts.

Example 5: Simple Port Forwarding

In this example, we look at a scenario in which port tunneling is required for communication between a client system, `my_desktop_LINUX`, and a secure server system, `my_secure_LINUX`. The server config file needs to specify both the client name and the server name. We will also specify the directory on our secure system where we want to start browsing. Lastly, we will simply specify that we want to make use of port forwarding.

```

ServerName: my_secure_LINUX
ServerType: shared_mem_parallel
ServerDirectory: /on_secure_LINUX/fv/bin
StartDirectory: /on_secure_LINUX/my_data
NumProcs: 16
!
! -- another comment line -----
!
ClientName: my_desktop_LINUX
UserName: my_name
PortForward: true

```

Port forwarding is only supported if `RemoteShell` is set to either `ssh` or `plink`. In this example, we have not explicitly set `RemoteShell` to `ssh` - this is not a requirement since `ssh` is the default. When the server config file is read, and you browse to the `StartDirectory` and/or attempt to read data, the following server command is executed by **FieldView**:

```
ssh -C my_secure_LINUX -l my_name -R 26010:my_desktop_LINUX:26010
```

```
'/on_secure_LINUX/fv/bin/fvrunsrv -np 16 localhost -port 26010' &
```

Here the port number, 26010, has been specified using the environment variable:

```
FV_DEFAULT_PORT = 26010
```

The fvrunsrv Wrapper Program

The `fvrunsrv` wrapper simplifies the start-up of **FieldView** servers, especially in parallel modes where the program `mpirun` and its arguments must be used. The wrapper generates the command line syntax required to launch servers based on the simplified argument set shown below. The `fvrunsrv` command is also used internally when the server configuration fields `ServerDirectory:` or `ServerType:` in Server Configuration (`.srv`) files are used. This command has the general syntax:

```
fvrunsrv [-np N] [-hosts host1[:N1],host2[:N2],...]
          [-machinefile <file>]
          [-p4_rshcommand <command>] client_hostname [-port NN]
```

where:

- np If present, specifies the number of processes, N.
 Without `-hosts`, **FieldView** runs share memory parallel.
 With `-hosts`, **FieldView** runs cluster parallel.
 If not present, **FieldView** runs standard client-server.
- hosts Specifies the list of hostnames, with an optional number of processes for each host.
- machinefile Specifies the machine filename (follows `MPI` syntax). This file must be visible from the server.
- p4_rshcommand Specifies the command used by `MPI` to start worker processes. If not present, this defaults to `ssh`.
- port Specifies the **FieldView** server port number, NN.
 This port can be tunneled through your Remote Shell connection.

The presence of the `-hosts` option implies that **FieldView** will run cluster (`p4`) **FieldView** parallel. This selection also requires that the `-np` option must be set. Using the arguments from the `-hosts` option, a temporary machines file is created. The `-machinefile` option is an alternate way to specify machines for use with cluster **FieldView** Parallel. If the `-hosts ANY` option is used, cluster **FieldView** parallel is run with the default machinefile. For Windows servers, use the `-hosts ANY` option or the `-`

`hosts` WINHPC option to specify cluster operation. The Windows HPC job scheduler will select the specified number of processors from the idle processors in the cluster.

Some simple examples follow below:

```
fvrunsrv                Runs standard FieldView client-server, local.

fvrunsrv -np 4          Runs FieldView shared mem parallel server, local, 4 processes.

fvrunsrv -np 4 -hosts myhostA,myhostB:2
                        Runs cluster (p4) FieldView Parallel, remote, 4 processes, and
                        builds a temporary machine file with:
                        myhostA
                        myhostB:2
```

The `fvrunsrv` program can also be used in Server Configuration (`.srv`) files using the deprecated field `ServerProgram:`.

While most of these arguments can be specified conveniently in the server config file, this simplification offers a specific benefit to those who need to run in **FieldView** Client-Server or **FieldView** Parallel in the manual mode.

For Windows users, the same syntax for the `fvrunsrv` program is used, with the following differences. First, a warning is issued if the `-machinefile` or `-p4_rshcommand` options are used - these options for running cluster parallel are not available on Windows. To run a cluster **FieldView** parallel job on Windows, use:

```
-hosts ANY (WINHPC is also an acceptable argument)
```

This will submit a cluster **FieldView** parallel job to the Windows HPC Server, which chooses the desired number of processors from the cluster automatically. At present, we are unable to allow you to choose specific server machines from within the Windows HPC cluster.

FieldView Parallel can only be run using the 64-bit servers.

The `fvrunsrv` program will run the 64-bit servers if the local machine is capable of running them, and they have been installed. It is highly recommended that all parallel servers be installed on all platforms which can run them.

Additional Comments Regarding **FieldView** Servers

An environment variable, `FV_DEBUG_AUTOSTART`, echoes the server command line that **FieldView** is trying to execute when `AutoStart: true` is set in your server config file. You

will also see the server command line echoed when selecting a file to be read from your file browser.

The environment variable `FV_DEBUG_STARTUP` helps debug **FieldView** problems with clients and servers on non-Windows systems that are caused by limited system resources. If this environment variable is set, then **FieldView** prints the following information to the console on startup. `RLIMIT_DATA` is the maximum allocated data allowed for this process, in bytes. `RLIMIT_AS` is the maximum address space (virtual memory size), in bytes. `RLIMIT_STACK` is the maximum size of the process "stack", in bytes. `RLIMIT_CPU` is the maximum amount of CPU time allowed for this process, in seconds. The "hard" limit is the important number. On startup, with or without this environment variable, **FieldView** tries to set the soft limit equal to the hard limit (the soft limit is often smaller); if this fails, a message is printed to the console. The "ulimit" shell command can also print these limits.

When a local server is started, the same information is printed about the server (which should be identical to the client, since it is local). To get information about a non-local server, the environment variable needs to be set for the server. This is easy for a manual server. For a non-manual remote server, the environment variable can be set on the command line in the server config file. This is also supported for server-only platforms (Solaris, IBM AIX).

Console output from `FV_DEBUG_STARTUP`

```
-----
RLIMIT_DATA soft limit: NNNN
RLIMIT_DATA hard limit: NNNN
RLIMIT_AS soft limit: NNNN
RLIMIT_AS hard limit: NNNN
RLIMIT_STACK soft limit: NNNN
RLIMIT_STACK hard limit: NNNN
RLIMIT_CPU soft limit: NNNN
RLIMIT_CPU hard limit: NNNN
sysname: Linux (or Darwin or SunOS or AIX)
nodename: [name of the machine running FV]
release: [OS/kernel release]
version: [OS/kernel version]
machine: [machine type, such as x86_64]
```

For Linux only, we also print the output of "cat /proc/meminfo", which is about 30 lines of information about physical memory (RAM) and virtual memory (RAM plus swap).

An additional environment variable, `FV_NO_SCONFIG_ECHO` has been added to turn off the server config print to the console window. Setting this variable makes the console output consistent with previous **FieldView** releases.

Based on customer feedback and our own testing, we have set the `ReadWriteTimeOut` default to be 0 - this changes from the previous default of 120 seconds. This is a necessary change since the `ReadWriteTimeout` mechanism does not work correctly with port forwarding. Server time-outs are part of the client-server start-up handshake. The benefit to users is that they will no longer need to type the time-out on the command line for manually starting servers. If these time-outs are still specified, they are ignored and a warning message is issued to state that the time-outs should be specified in the server config file.

In order to run **FieldView** in a standard (client-server) or parallel (shared memory or cluster) automated mode of operation, we strongly recommended that passwordless `ssh` is implemented. Please also note that passwordless connections are a requirement between the nodes of a cluster in order for `p4` parallel to work.

At the time of this release, starting a Win64 cluster parallel server automatically from a remote system (one outside the cluster) does not work because the `WINHPC "job"` command does not allow us to specify where the controller server will run. If this is attempted, you will see an error message generated by Windows which is similar to:

```
"Aborting: Access denied by node 'compute-0-XX'  
This node is a resource managed by the Microsoft HPC Scheduler  
and mpiexec was attempting to use it without a scheduled job."
```

You can still run **FieldView** Parallel on a Windows 64-bit cluster as a local server, or by manually starting a remote server.

At present, **FieldView** Parallel has the requirement that the controller server must be run on the head node of the cluster in order for it to be seen remotely.

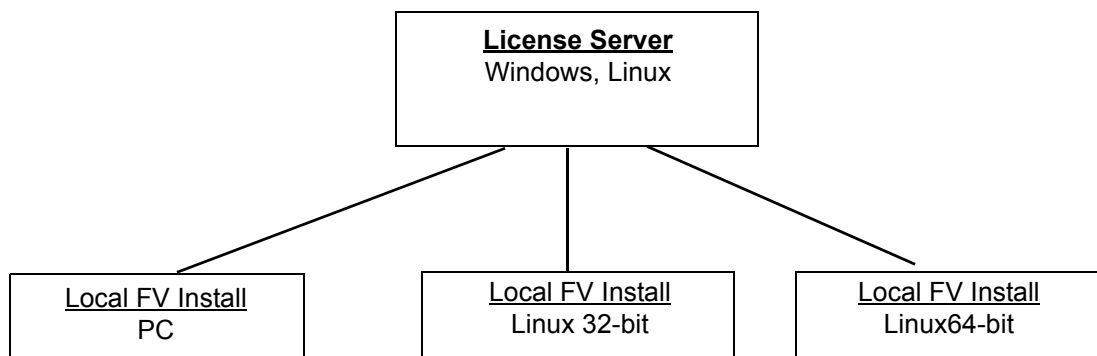
FieldView Licensing

Before your software can be enabled, the product must be properly licensed. **FieldView** uses either Concurrent User (also called ‘floating’) licensing, the most flexible licensing method, or Single System (Node-Locked) licensing. **FieldView** can also be licensed for virtual environments subject to certain restrictions. A valid licensing key will need to be issued by **FieldView CFD**. If you try to run this application without performing these steps, **FieldView** will exit with a message indicating that your software has not been properly licensed.

Concurrent User (Floating) Licensing

This method of licensing is the most flexible and powerful. The way that this licensing method works is that one physical machine operates as the license server. Licenses are checked out and checked back in and **FieldView** can be run on any platform that it has been installed on as long as licenses are available. With this method of licensing, the license manager is installed on the license server. Then, **FieldView** is installed on any machines that it will be run on, choosing the appropriate machine type when asked during the installation script. The installation setup can be done one of two ways:

The **License Server** is on same network as local machines but does not have to have directories mounted. The local machines only have to be able to ping the license server.



The **Local** machines have **FieldView** installed. The `/data` directory of the local **FieldView** installation typically provides a license file configured to access **FieldView** licenses from the license servers.

Figure 1: License Retrieval Architecture

With this option, **FieldView** is installed on any desired platform *locally*. When **FieldView** is run, it looks at the *local* copy of the `license.dat` file to determine which machine is

acting as the license server. It then contacts the license server for a license. If one is available, it is checked out to that machine. The directories where **FieldView** license utilities are installed on the license server do *not* have to be NFS-mounted by the local machines running it, but they all have to be on the same network. **FieldView** can be run on the License Server, but this is not required.

Single System (Node-Locked) Licensing

With this method of licensing, **FieldView** can only run on the machine in which it is licensed. See [Obtaining a FieldView License on page 38](#) for instructions on how to obtain a license and [Configuring a Single System License on page 40](#) for installation instructions.

FieldView Parallel Licensing

You can run **FieldView** servers with up to eight parallel worker processes and one controller process with a standard **FieldView** license at no extra cost.

If you would like to increase your current **FieldView** Parallel capacity you will need a special license. **FieldView** Parallel licenses are available for specific numbers of processors up to 64 and beyond 64 (HPC option).

As a specific illustration, here is what a properly configured license file would look like for a standard **FieldView** installation for **FieldView** 20 with provision for 3 concurrent users, including support for **DataGuide**TM (*dg*) and standard support for up to 8 Parallel worker processes (*pfv8*):

```
# This is the FIELDVIEW license file.
```

```
DAEMON ilight
```

```
SERVER my_system 181933454c13 1602
INCREMENT pfv8 ilight 20.000 dd-mmm-yyyy 3 1E228D6A09EAF081AFBD \
VENDOR_STRING="fv every dg +333f96373" ISSUED=dd-mmm-yyyy \
SIGN="0038 D8A8 40AD 8DA4 6AD7 0125 2D85 DB00 2A5E EC38 BC17 \
FDE9 1B40 F1BA A6E5"
```

For **FieldView** Parallel, the license file contains a string, *pfvN*, where *N* specifies the maximum number of available parallel processes. In the example above, the license file permits operation of **FieldView** Parallel for up to 8 processes (*pfv8*).

To illustrate the operational difference between **FieldView** Parallel licensing and **FieldView** licensing, we will take a step back to consider **FieldView** Client-Server. For this mode of operation, **FieldView** Servers are not licensed. It is possible to run multiple **FieldView** Servers with a single **FieldView** Client. In [Figure 2](#), two separate Server processes are used to read two different datasets. There is no limit to the number of Client-Server processes which can be run with **FieldView**. So, if you have a single user

license, it is possible to run as many Client-Server processes as you wish, subject to the practical limitations of your hardware (i.e. RAM, graphics).

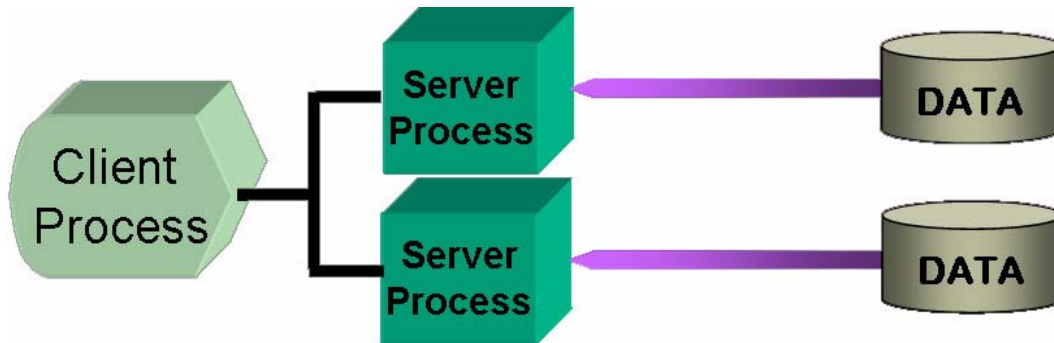


Figure 2: FV Client using two of an unlimited number of Server Processes

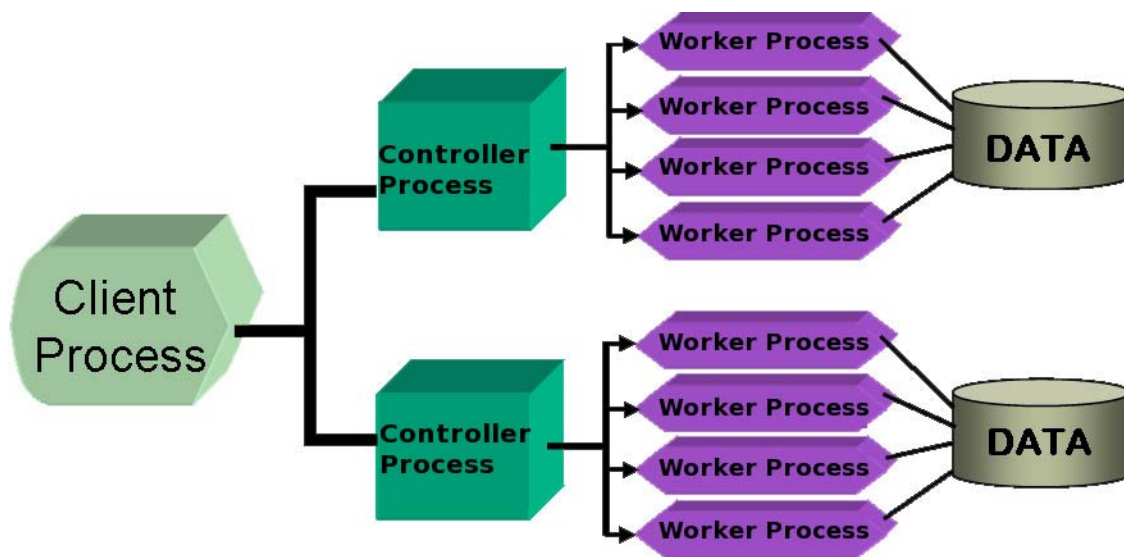


Figure 3: FV Parallel with Licensed Controller Processes

For the case of **FieldView** Parallel, following what is done for Client-Server, it is possible to run an unlimited number of server 'Controller' processes. What is licensed with the **FieldView** Parallel license is the total number of Worker Processes which can be run; the Controller processes are not counted towards this limit. In the example above (see [Figure 3](#)), we illustrate how it is possible to read two different datasets using a total of eight (8) worker processes, applying four (4) worker processes to each dataset. To do this, a **FieldView** Parallel license for 8 processes would be required.

FieldView Batch-Only Licensing

This licensing option lets you run several instances of **FieldView** in a low cost, batch-only mode of operation. Running this way makes concurrent postprocessing possible.

The key limitation is that a GUI cannot be used in batch mode, so **FieldView** must be run from a script. With a batch-only license you can run **FieldView** from a command line as follows:

```
fv -fvb -s my_script.scr

fv -fvb -fvx my_program.fvx
```

In the first example above, **FieldView** is run in batch mode, executing the commands in the **FieldView** script file called `my_script.scr`. In the second example, an **FVX** program called `my_program.fvx` is run in batch mode. The command line option, `-fvb`, is used to specify two things. First, that **FieldView** will be run in batch mode, and, second that it will use a batch only license for this run. The above examples could also be executed as follows:

```
fv -batch -s my_script.scr

fv -batch -fvx my_program.fvx
```

The key difference between the `-batch` switch and the `-fvb` switch is that in the former case, a normal interactive **FieldView** license will be used to run your job. When you plan to run in batch mode, we recommend that you use a batch-only license.

Batch-only licenses are available for parallel operation, for increasing numbers of processors, defined as being up to 8, up to 16, up to 32 and up to 64. For batch-only cases in which more than 64 processes are required, special licensing options are available.

Depending on your licensing options you will have one or more of the following increment lines present in your **FieldView** `license.dat` file:

```
INCREMENT fvb8 ilight 14.000 dd-mmm-yyyy 3 9E124DEA8867477EDFE6 \
VENDOR_STRING="fv every dg +w2w7288485" ISSUED=dd-mmm-yyyy \
SIGN="0060 7A23 4082 47CE 9DD1 D12E FC85 E300 C07E AA09 BA12 \
6FFF 86C9 3A5F F86C"
```

```
INCREMENT fvb32 ilight 14.000 dd-mmm-yyyy 3 9E124DEA8867477EDFE6 \
VENDOR_STRING="fv every dg +w2w7288485" ISSUED=dd-mmm-yyyy \
SIGN="0060 7A23 4082 47CE 9DD1 D12E FC85 E300 C07E AA09 BA12 \
6FFF 86C9 3A5F F86C"
```

```
INCREMENT hpcfvb32 ilight 14.000 dd-mmm-yyyy 3 9E124DEA8867477EDFE6 \
VENDOR_STRING="fv every dg +w2w7288485" ISSUED=dd-mmm-yyyy \
SIGN="0060 7A23 4082 47CE 9DD1 D12E FC85 E300 C07E AA09 BA12 \
6FFF 86C9 3A5F F86C"
```

The number of worker processes which can be run will be signified by the number designation in the `fvb##` string.

Using Batch-Only Licensing: Creation of transient animations

Batch-only licensing is well suited for creating transient animations. Concurrent operation can be used to break the range of transient data into sub-ranges, each of which can be processed individually. Suppose we want to calculate streaklines, based on 1200 seed locations. If we were to run the transient sweep for all seeds using only one instance of **FieldView**, the time to complete this (notional) task would be 600 minutes (see [Figure 4](#) at right). However, if we split the problem into four separate rakes, and run the calculations over these time-steps simultaneously, the overall time saved is a factor of four. To run four instances of **FieldView** simultaneously, you can use four batch-only licenses. For each **FieldView** run, the same script would be executed; the only difference would be the time range being studied.

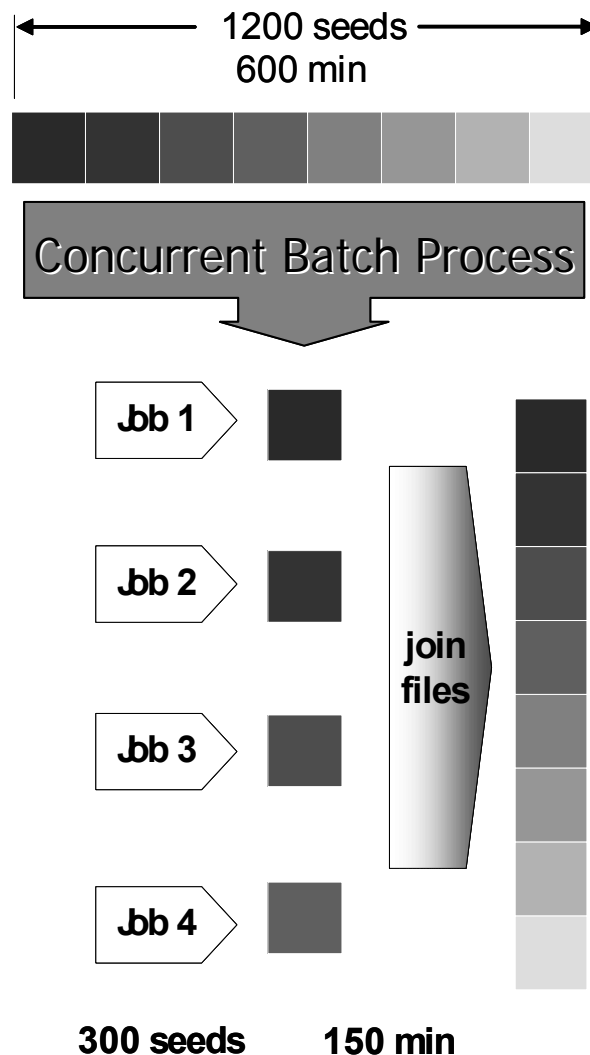


Figure 4: Concurrent Batch Operation

The **FieldView** scripts needed to carry out this concurrent set of operations would look something like this:

```
! streak_A.scr
RESTART ALL prep_streak
```



```

RESTART STREAM rake01

SWEEP TIME LOOP 1 1 rake01.fvp

! streak_B.scr
RESTART ALL prep_streak
RESTART STREAM rake02

SWEEP TIME LOOP 1 1 rake02.fvp

! streak_C.scr
RESTART ALL prep_streak
RESTART STREAM rake03

SWEEP TIME LOOP 1 1 rake03.fvp

! streak_D.scr
RESTART ALL prep_streak
RESTART STREAM rake01

SWEEP TIME LOOP 1 1 rake04.fvp

```

Within each of the four scripts above, a general restart (`prep_streak`) is used to compose the scene and create any surfaces and/or titles of interest. Since we will be calculating four separate streakline rakes, we have four separate streamline (`STREAM`) restarts. The streakline rakes are calculated over all timesteps using the `SWEEP` command. From a command line, using batch-only licensing, you would type the following:

```

fv -fvb -s streak_A.scr -port none &
fv -fvb -s streak_B.scr -port none &
fv -fvb -s streak_C.scr -port none &
fv -fvb -s streak_D.scr -port none

```

In the above example, note that the command line switch, `-port none`, is used to let concurrent instances of **FieldView** run without each of them attempting to use the same default port (12345) for operation. Alternately, individual port numbers may be explicitly specified. Since all of the restarts in this example are attempting to read the same time-step initially, it may be a good idea to delay the start of each instance of **FieldView** using a `SLEEP` command.

FieldView Licensing for Virtual Environments

We appreciate that many of you are working with "virtual" operating systems to ease system administration tasks and to lower your operating costs. Virtual systems however

present a problem since they lack the hardware security that standard FlexLM needs to operate. To meet your needs as this working environment gains popularity, **FieldView CFD** has implemented Flexera's new handling methods for virtual systems.

The primary difference between licensing on physical versus virtual systems is illustrated in **Figure 5** below. **FieldView** supports running a license server within a Linux or Windows virtual machine running on a VMware ESX or VMware ESXi hypervisor.

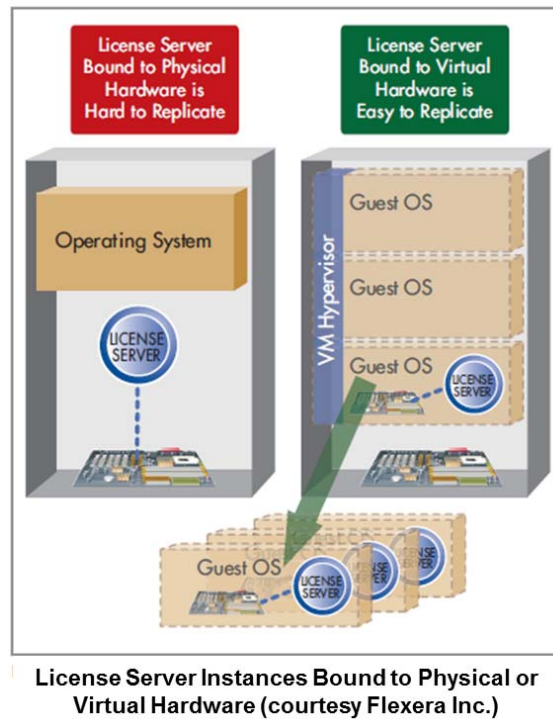


Figure 5: Physical versus Virtual Environments

For virtual environments, virtualization supported by FlexNet is illustrated in **Figure 6** on the following page.

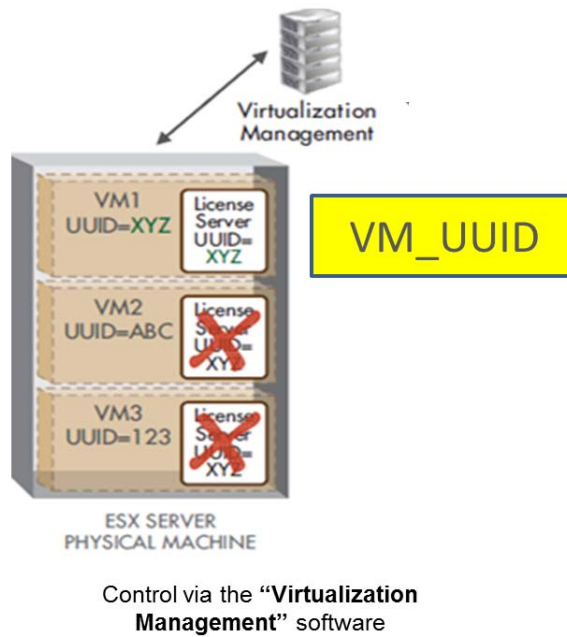


Figure 6: Virtualization supported by FlexNet

Important Notes

If you are using a virtual system for license management, you will need additional software and specific passwords from **FieldView CFD**. To obtain additional documentation for **FieldView** licensing for VMs, please contact support@fieldviewcfd.com.

If you need to run your license server bound to a virtual machine (see **VM_UUID** in [Figure 6](#)), you will need to sign a special binary license agreement before **FieldView** passwords can be issued.

Virtual support is limited to Linux and Windows systems only and is not available for MAC (license server only).

Note that FlexNet no longer supports binding a virtual system to a remote physical one (a method known as lmbind).

Licensing via Server

This feature is included with **FieldView Parallel 16** and above, and allows users to run **FieldView** when only the **FieldView Server** has access to a license.

In this section, "**FieldView Server**" refers to the process that communicates with the **FieldView Client** when reading and post-processing data on a remote system, and "license manager" the process that controls the number of **FieldView** processes that can run at a given time (sometimes called "license server" in other sections of our documentation).

As explained in the [FieldView Licensing](#) section, **FieldView** can check licenses by connecting to a license manager. As a reminder, only the **FieldView Client** needs a license to run. **FieldView Servers** are not licensed. In some configurations, one may need to launch the **FieldView Client** on a system from which the license manager is not visible, while it is for the **FieldView Server**. [Figure 7](#) below illustrates such an example, in which a user is connecting to a cluster through a secured connection: the license manager is visible from the **FieldView Server**, running for instance on the cluster's head node, but is not visible to the **FieldView Client**.

This Licensing via Server option provides a solution to this problem.



Figure 7: Remote user (Client) and License Manager on secure network

Licensing via Server requires a working Client-Server setup. Only a standard Client-Server configuration file can be used, not a parallel one. For more on this, see the section [Testing & Configuring FieldView Servers](#).

To start **FieldView** using Licensing via Server, use the "`-lvs`" command line switch when starting the **FieldView Client**, as detailed below:

```
fv -lvs server_config_file:port@host
```

Where

- `server_config_file` is the name of a valid Client-Server configuration file (see [Step 3 – Setup a Server Configuration File](#) for more details)
- `port` is the TCP port number for the license manager. This port number is specified in the `SERVER` line of the license file.
- `host` is the hostname of the system on which the license manager is running. Alternatively, this could be replaced by its IP address.

The license manager does not have to be running on the same system as the **FieldView** Server; the license only needs to be "visible" from the **FieldView** Server.

Licensing via Server works in Client-Server autostart or manual mode.

The Client-Server connection used for accessing a license does not have to be used for reading data after that. Simulation results can be read locally, in Client-Server mode or in parallel, using the same Server or a different one.

FieldView issues the following console messages:

```
FieldView found a license via a manually started server.  
FieldView found a license via a server on <serverhostname>.
```

Limitations

Licensing via Server is only supported for `pfv16` (and over) licenses.

Licensing via Server is unsupported for batch-only (`fvb`) licenses.

Licensing via Server is unsupported for Solaris and IBM **FieldView** Servers.

Licensing via Server can only be used with a server configuration file using "ServerType" standard, and not any of the parallel types.

Obtaining a FieldView License

FieldView 20 requires FieldView 19 passwords. To obtain **FieldView 19** passwords, please contact Customer Service (passwords@fieldviewcfd.com), following the instructions provided below.

Linux & MAC

After installation of **FieldView** on your Linux machine, you need to provide information about your machine to Customer Service so that a license can be issued. This system information is easily provided by using the `fv_server` utility, which was installed along with the **FieldView** software. This utility can be found on the `/bin` directory of the **FieldView** installation area. Change to this directory and then run the following command:

```
./fv_server id
```

The relevant output of this command will look something like this:

```
...  
The FLEXnet host ID of this machine is "690f23e5"  
FLEXlm server name = wing
```

This information needs to be sent to Customer Service (passwords@fieldviewcfd.com) along with user and company information. Once this information is received and processed, a license will be generated and sent to you.

Windows

After installation of **FieldView** on your Windows system, you need to provide information to Customer Service so that a license can be issued. This system information is easily provided by using the **Get Host ID** utility, which is installed along with the **FieldView** software. This utility can be found on the Start menu as follows: Start → Programs → **FieldView 20** → Get Host ID. Running this utility will bring up a text file with the requisite information, as shown in **Figure 8**.

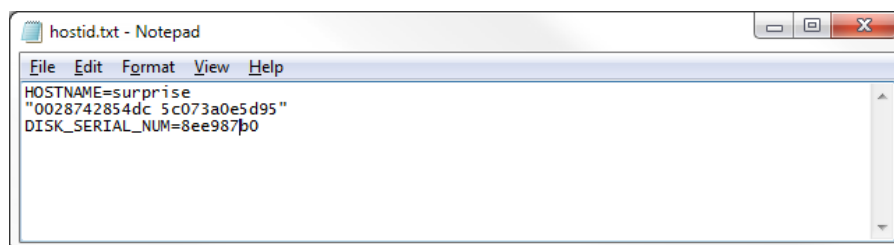


Figure 8: Sample Text File with Windows system information

Save this file to a convenient location on your system, copy the file contents into an email message and send it to **FieldView** Customer Service (passwords@fieldviewcf.com). Also, please be sure to state which operating system you are using to host licensing (e.g., Windows Server 2012, Windows 10, etc.). Once this information is received and processed, a license will be generated and sent to you.

License Expiration Warning

FieldView has been enhanced with a License Expiration Warning message that alerts users when their password is about to expire and provides contact information for license renewal.

Within 30 days of license expiration, **FieldView** will issue a warning message upon startup. A pop-up window will appear in the GUI, as illustrated in **Figure 9**.



Figure 9: FieldView License Expiration Warning

A message will also appear in the console window on the command line, e.g.,

```
Warning: This license will expire on 8-jul-2018.
```

This reminder benefits all **FieldView** users by giving them advance warning to renew their licenses. Users who have acquired **FieldView** directly through **FieldView CFD** are provided with a live link to easily request a new password, while users who have acquired **FieldView** through a local distributor or sales representative are referred to their local **FieldView** representative.

FieldView License Installation Tasks on Linux

Configuring a Concurrent User (Floating) License

The installation of the license key is simple, but can be prone to cut and paste errors, and unprintable characters copied from Outlook are not handled by the licensing utility.

(Passwords can also be provided to you as a file attachment to your email.) Please read these instructions carefully.

1. Change your current working directory to the directory where **FieldView** is installed, and then into the subdirectory `/data`.
2. Open the following file for edit: `license.dat`.
3. Add the `SERVER` line provided (by email or fax) by Customer Service.
4. Make sure the installed `DAEMON` line is correct, and *not* commented out. (The comment symbol is “#”).
5. Add the `INCREMENT` information (which may span several lines and be sure not to delete the continuation character “\” at the end of all but the last line) in your license file.
6. Save the license file. See [Licensing Issues and Problems on page 51](#) if you have any questions about the required content of the file.
7. To test the `license.dat` file, you should stop and then start the server. To do this, change the directory to the **FieldView** `/bin` directory, then run the following commands:

```
./fv_server stop
./fv_server start
./fv_server status
```

NOTE: The following error will result if your operating system is lacking LSB (Linux Standard Base) compatibility:

```
lmgrd: No such file or directory
```

Flexera recommends resolving this by creating a symbolic link in `/lib64`

```
ln -s ld-linux-x86-64.so.2 ld-lsb-x86-64.so.3
```

If you do not get the **FieldView** license server up and running, please refer to [Licensing Issues and Problems on page 51](#). Contact support@fieldviewcfd.com (**FieldView CFD** Support) if this section does not help resolve the problem.

Configuring a Single System License

Running the license manager is required, however **FieldView** can only be launched on the system running the license manager.

Once you have obtained the license from **FieldView CFD** install the `license.dat` file in the data directory of the **FieldView** installation. **FieldView** will read the `license.dat` in the data directory upon start-up. If the license is not valid, you may see this error printed to the command prompt:

```
valid floating license not found
```

Although the license is a single system license, “floating” license will be included in the error message. This can be ignored, however, if you do receive this error please contact support@fieldviewcfd.com for technical assistance.

Running the License Server on Linux

The license server handles the checking out and checking back in of **FieldView** licenses. It keeps track of the number of licenses available and hands out licenses to various users as they are requested until the licensing limit is reached. When running the install script (`installfv.sh`), and after the **FieldView** installation, you will be prompted to install the floating license manager; we recommend that you answer `yes`. The script will then add the `fv_server` utility to the `/bin` directory, create a `/flexlm` directory containing platform specific daemons, and add all other necessary files.

It is not necessary to have **FieldView** run on the same system where the license server is installed (although it can be), since its primary responsibility is to handle licensing requests. If you do not wish to run **FieldView** on the license server, then you can avoid the installation of **FieldView** altogether. In this case, you should run the installation script *for the licensing only*, called `installcd_flexlm.sh`. When you run this shell, it will prompt you for the name of the directory where **FieldView** is installed. Since there is no **FieldView** installation, merely enter in the directory where you want the **FieldView** licensing to be installed. The script will notice that there is not a **FieldView** installation already in the directory that you specified (which is normally the case). This is expected, and the prompt should be answered `yes`. The script will then create a directory called `fv` in the directory you specified, with the sub-directories `/bin` (containing the `fv_server` utility), and `/flexlm` (which contains platform specific daemons). To finalize this task, you will need to create `/data` and place an empty `license.dat` file at this location.

The directory containing the license server installation does *not* have to be ‘mounted’ to the file system of machines that will be used to run **FieldView**. The only requirement is that the machine that is acting as the license server can be ‘pinged’ (the Linux `ping` command) by any machine that will be running **FieldView**.

`fv_server` Licensing Tool

We provide an easy-to-use ‘wrapper’ shell tool to all of the needed FLEXlm utilities called `fv_server`. This tool provides system information which can start and stop the server, as well as provide diagnostic information for Customer Service.

```
./fv_server
```

The set of valid arguments are:

```
id, start, status, stop, reread, diagnose, version
```

The status of the license server is output by running the following command in `FieldView_installation_directory/bin`:

```
./fv_server status
```

If the server is up and running and there are no errors in the license file, the output will contain the following information:

```
...
Flexible License Manager status on Fri 12/4/2020 08:34

License server status: 1602@server_name
  License file(s) on server_name: FieldView_installation_directory/
  data/license.dat:

  server_name: license server UP (MASTER) v11.16

Vendor daemon status (on server_name):

  ilight: UP v11.16
Feature usage info:

Users of fv: (Total of 1 license issued; Total of 0 licenses in
use)
```

Some instances of errors and the resulting `status` output are given in the section [Licensing Issues and Problems on page 51](#).

Diagnostic information which can be used to resolve licensing problems can be generated using:

```
./fv_server diagnose
```

The information provided when there are *no* problems is as follows:

```
...
FLEXnet diagnostics on Fri 12/4/2020 08:34

-----
License file: FieldView_installation_directory/data/license.dat
-----
"fv" v19.000, vendor: ilight
  License server: server_name
  floating license starts: 4-Dec-2020, expires: 31-dec-2020
```

This license can be checked out

Some instances of errors and the resulting `diagnose` output are given in [Licensing Issues and Problems on page 51](#). The output of the `diagnose` command should be provided to Customer Support if they request it to help them debug any problems you may be having with running **FieldView**.

FieldView License Installation Tasks on Windows

Configuring a Concurrent User (Floating) License

The installation of the license key is simple, but can be prone to cut and paste errors, and unprintable characters copied from Outlook are not handled by the licensing utility. (Passwords can also be provided to you as a file attachment to your email.) Please read these instructions carefully.

1. Navigate to the directory where **FieldView** was installed, and then into the subdirectory `/data`.
2. Open the following file for edit: `license.dat`. You will need administrator privileges to edit this file.
3. Add the `SERVER` line provided (by email or fax) by Customer Service.
4. Make sure the installed `DAEMON` line is correct, and *not* commented out. (The comment symbol is "#").
5. Add the `INCREMENT` information (which may span several lines and use the continuation character "\ " at the end of all but the last line) to the license file.
6. Save the license file.

Configuring a Single System License

Once you have obtained the license from **FieldView CFD**, perform these steps first:

1. From the Start menu, select Start → Programs → **FieldView 20** → FLEXIm License File. This brings up the license file `license.dat` in Notepad for editing.
2. Cut and paste (if you received the license information by email), or *carefully* copy the information from the fax (`SERVER` line and `INCREMENT` lines) into the `license.dat` file. Specifically, add the `SERVER` line before the `DAEMON` line. Comment out the `DAEMON` line "DAEMON `ilight`". Add the `INCREMENT` lines provided by **FieldView CFD** after the `DAEMON` line.

3. Save the `license.dat` file when you are done editing it. See [Licensing Issues and Problems on page 51](#) if you have any questions about the required content of the file.

Running the License Server on Windows

The set-up of the Concurrent User license requires that the server machine be configured with the **LMTOOLS** utility so that the server can be started, stopped, and otherwise managed (see [Figure 10](#)).

The **LMTOOLS** utility is installed by default for all versions of Windows. It is installed in the `/bin` directory under **FieldView** main install location. The **LMTOOLS** utility is run as follows: Start → Programs → **FieldView 20** → LMTOOLS.

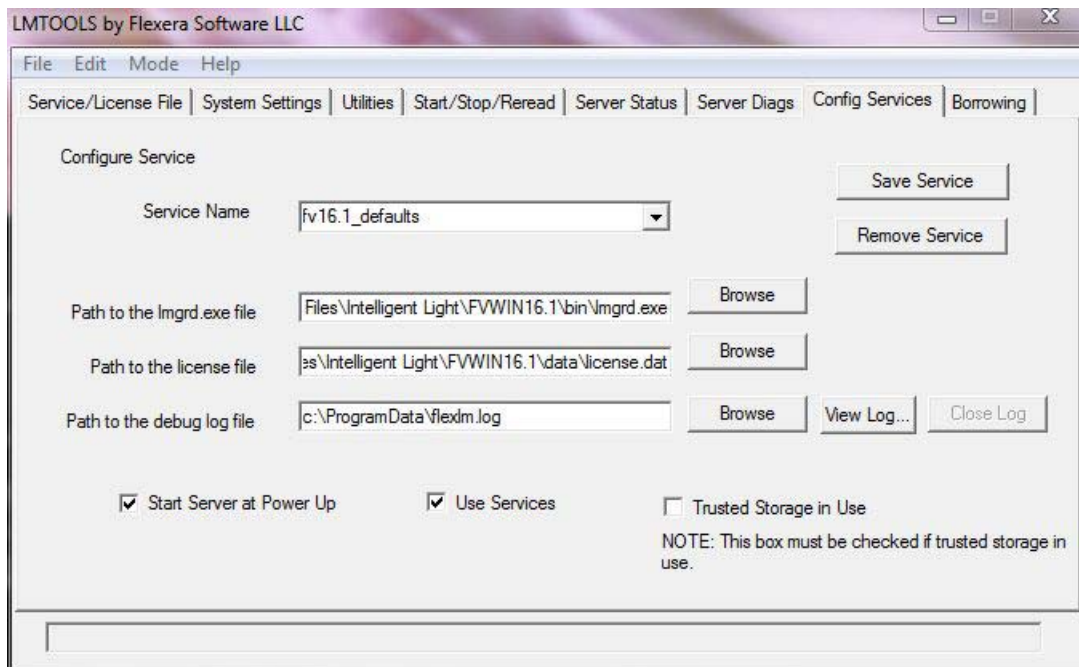


Figure 10: FLEXlm Config Services Tab

To set up a Windows license server, follow these steps:

1. From the Start menu, select Start → Programs → **FieldView 20** → LMTOOLS
2. Press the **Service/License File** tab. Select the **Configuration using Services** radio button. Please note that when this is done, the content for some of the tabs will change. A list of your existing FLEXlm services (if any) will be displayed.

3. Press the Config Services tab (see [Figure 10](#)). If you already have a **FieldView** Service Name setup, you will need to update this for **FieldView 20**. To update the setup, proceed to step 4. If you do not have an existing **FieldView** service, you will have to create one. To do this, in the Service Name field, type in the phrase `FieldView License` (DO NOT press the Enter key after this).
4. Press the **Browse** button next to the field **Path to the Imgrd.exe file**. Browse to the `\bin` directory of the **FieldView 20** installation area. By default, this is

```
C:\Program Files\FieldView CFD\FVWIN20\bin
```

When you find the file, select it and press the **Open** button on the browser. This will place the full path for this file into the empty field to the left of the Browse button.

5. Press the **Browse** button next to **Path to the license file**. Browse to the `\data` directory of the **FieldView 20** installation area. Since you were last in the `\bin` directory, merely go up one level and then into the `\data` directory. In the **Files of type** drop-down field, select **License Files (*.dat)**. When you find the file, select it and press the **Open** button on the browser. This will place the full path for this file into the empty field to the left of the Browse button. Be aware that at this stage, the `license.dat` file is merely a template.
6. Set **Path to the debug log file** to `"flexlm.log"` anywhere in the path `%SystemDrive%\ProgramData` (typically `C:\ProgramData`). System security prevents this file from being written to arbitrary locations.
7. Check the **Use Services** button.
8. Check the **Start Server at Power-Up** button (*optional*). This will cause the server to be restarted if your machine is rebooted. Otherwise, after reboot, you will have to manually restart the server.
9. Press the **Save Service** button (located at the top right part of the tab) to save the setup.
10. Press the **Service/License File** tab. Select the **Configuration using Services** radio button.
11. Press the **Start/Stop/Reread** tab. Select **FieldView License** in the list of services. Press the **Start Server** button.
12. Try to run the **FieldView** software using the Start menu program group entry for **FieldView 20**.

If you do not get the **FieldView** license server up and running, please refer to [Licensing Issues and Problems on page 51](#). Contact support@fieldviewcfd.com (**FieldView CFD** Support) if this section does not help resolve the problem.

FieldView should read the `licence.dat` in the data directory upon start-up. If the license is not valid, you may see this error printed to the command prompt:

```
valid floating license not found
```

If you do receive this error please contact support@fieldviewcfd.com for technical assistance.

The license server handles the checking out and checking back in of **FieldView** licenses. It keeps track of the number of licenses available and hands out licenses to various users as they are requested until the licensing limit is reached.

FieldView must be installed, even if it is not going to be run, on the machine acting as the license server. It is possible, though not necessary, to run **FieldView** on the license server, since the primary responsibility of the license server is to handle licensing requests.

The directory containing the license server installation does *not* have to be 'mounted' to the file system of machines that will be used to run **FieldView**. The only requirement is that the machine that is acting as the license server can be 'pinged' (the `ping` command) by any machine that will be running **FieldView**.

LMTOOLS

LMTOOLS has been provided to run, manage and diagnose problems with the license manager. This tool's functionality is accessible from tabs, some of which are described below.

System Settings Tab

This tab provides the Hostid Settings as well as the Time Settings information. The fields necessary to issue the license are under **Hostid Settings**. Specifically, they are Host-name, IP Address, Ethernet Address, and Disk Volume Serial Number.

Server Status Tab

Pressing the **Perform Status Enquiry** button will tell you whether the license server is UP (running) or not. If not, it will give diagnostic error messages. This output is useful to Customer Care personnel at **FieldView CFD**. It can be sent to us by highlighting the text in the message window, copying it with CTRL+C, and pasting it to your email message.

If the license manager is up and running, the message printed in the window should look something like this (with the machine in this example named `surprise`):

```

-----
Status
-----
Flexible License Manager status on Fri 12/4/2020 08:34

[Detecting lmgrd processes...]
License server status: 7788@surprise
  License file(s) on surprise: C:\Program Files\FieldView
CFD\FVWIN20\data\license.dat:

    surprise: license server UP (MASTER) v11.16

Vendor daemon status (on surprise):

    ilight: UP v11.16

Feature usage info:

Users of fv: (Total of 1 license available)

```

If the license manager is not running due to some error in the `license.dat` file, the message printed in the window will look something like this (with the machine in this example named `surprise`):

```

-----
Status
-----
Flexible License Manager status on Fri 12/4/2020 08:34

[Detecting lmgrd processes...]
License server status: 7788@surprise
  License file(s) on surprise: C:\Program Files\FieldView
CFD\FVWIN20\data\license.dat:

lmgrd is not running: Cannot connect to license server system.
  The license server manager (lmgrd) has not been started yet,
  the wrong port@host or license file is being used, or the
  port or hostname in the license file has been changed.
Server name: 192.168.1.101
License path: 7788@surprise;
FlexNet Licensing error:-15,10. System Error: 10061 "WinSock:
Connection refused"
For further information, refer to the FlexNet Licensing
documentation, available at "www.flexerasoftware.com".

```

See [Licensing Issues and Problems, p. 51](#) for other types of error messages and their meanings.

Start/Stop/Reread Tab

When the server is started by pressing the **Start Server** button, two processes are started on the machine: `ilight.exe` and `lmgrd.exe`. These processes are the daemons that control the licensing. **FieldView** cannot be run if these processes are not running. The server (and associated daemons) can be shut down from this tab by pressing the **Stop Server** button. If a license server to a system of machines is shut down (i.e., turned off), all **FieldView** licenses will become unavailable.

Server Diagnostics Tab

This tab gives additional information about the license manager and the licenses in use. When the Perform Diagnostics button is pressed, a message similar to what is shown below will be displayed in the text window:

```

-----
Diagnostics
-----
FLEXlm diagnostics on Fri 12/4/2020 08:34

-----
License file: C:\Program Files\FieldView CFD\FVWIN20
\data\license.dat
-----
"pfv16" v19.000, vendor: ilight
  License server: surprise
  floating license starts: 4-Dec-2020, expires 31-dec-2020

This license can be checked out
-----

```

The message “This license can be checked out” may be misleading, as the diagnostic and status tools do *not* report usage. That is, if you have 4 floating licenses and 2 are being used, it will report that there are 4 available. If 4 licenses are in use and a 5th user tries to run **FieldView**, a pop-up window with the message “All floating licenses are in use” will be displayed.

Utilities Tab

This tab has two functions. One is to set or manage the vendor name. The other is a feature that provides the FLEXlm version given the file (`lmgrd.exe`). Using the **Find Version** feature provides output similar to below:

```

-----
Version
-----
FlexNet Licensing v11.16.x.x build 116574 (ipv6) x64_n6 (lmgr.lib),
Copyright (c) 1988-20XX Flexera Software LLC. All Rights Reserved.

```


FieldView and Other Software Using FLEXlm

You can run multiple products that require FLEXlm management on the same server. However, all software products must be served using a single running instance of `lmgrd`. In this case, a combined license file needs to be used, as all products need to use the same FLEXlm daemon, but you can only start one server. This one server will handle all licensing. Each product will have its own vendor daemon running (for **FieldView**, this is the daemon called `ilight`), but all will use a common `lmgrd`.

Using a Combined License File on Linux

In order to use a combined license file, a single `lmgrd` must be used for all products. This is assured if the setup is created so that all products use the same FLEXlm daemon (`lmgrd`). What this means is that all Service Name setups should be pointing to the same file through the same path. FLEXlm documentation states that the most recent version of `lmgrd` should be used for all software.

In order to determine the version of `lmgrd` a given software is using, either contact their support personnel or run the following command in a Linux shell in the directory containing that software's copy of `lmgrd`:

```
./lmgrd -v
```

You may get output similar to this listing:

```
lmgrd v11.4.100.0 build 50818 x64_se9 - Copyright (c) 1988-2007  
Macrovision Europe Ltd. and/or Macrovision Corporation. All Rights  
Reserved.
```

This particular output says that the version is 11.4.100.0. **FieldView** is currently running v11.16. Note that this version of FlexNet supports Internet Protocol version 6 (IPv6). You should run this command for each product using FLEXlm and use the `lmgrd` that is the most recent version.

If you buy software that uses a newer version of `lmgrd` than **FieldView** does, you should do the following:

1. Stop the license server.
2. Edit the global `license.dat` file and change the path on the `DAEMON` line to the newest daemon.
3. Copy the global `license.dat` file to each software's license file installation area, or use a link to prevent the need to do this. With a link to the global `license.dat` file, you only have to update one (1) file. This will also avoid the possibility of the various license files getting out-of-sync, which will cause server problems.

4. Start the server.
5. Check to make sure that all products with FLEXIm licensing now run.

When this is accomplished, and you have multiple products using FLEXIm, you should have one (1) copy of the process `lmgrd` running, plus one (1) vendor daemon for each product running as a process. The **FieldView** vendor daemon is called `ilight`.

Using a Combined License File on Windows

In order to use a combined license file, a single `lmgrd.exe` must be used for all products. This is assured if the setup is created so that all products use the same FLEXIm daemon (`lmgrd.exe`). What this means is that all Service Name setups should be pointing to the same file through the same path. FLEXIm states that the most recent version of `lmgrd.exe` should be used for all software.

In order to determine the version of `lmgrd.exe` a given software is using, either contact their support personnel or run the following command in an MS-DOS shell in the directory containing that software's copy of `lmgrd.exe`:

```
C:\Program Files\FieldView CFD\FVWIN20\bin\lmgrd.exe -v
```

You should get output similar to this:

```
lmgrd v11.16.x.x build 176543 x64_n6 - Copyright (c) 1988-20XX  
Flexera Software LLC. All Rights Reserved..
```

This particular output says that the version is `v11.16.x.x`. You should run this command for each product using FLEXIm and use the `lmgrd.exe` that is the most recent version. The `DAEMON` line in the `license.dat` file should point to the location of `ilight.exe`, which is in the `/bin` directory under the main **FieldView** install location.

If you buy software that uses a newer version of `lmgrd.exe` than that of **FieldView**, you should do the following:

1. From the Start menu, select Start → Programs → **FieldView 20** → LMTOOLS.
2. Select **Configuration using Services** on the Service/License File tab. Press the Configure Services tab.
3. Select **FieldView License Manager** in the Service Name field.
4. Press the **Browse** button next to the field **Path to the lmgrd.exe file**. Browse to the directory containing the *most recent* version of `lmgrd.exe`. When you find the file, select it and press the **Open** button on the browser. This will place the full path for this file into the empty field to the right of the Browse button.

5. Select the **Start/Stop/Reread** tab and start the server by pressing the **Start Server** button.

When this is accomplished, and you have multiple products using FLEXIm, you should have one (1) copy of the process `lmgrd.exe` running, plus one (1) vendor daemon for each product running as a process. The **FieldView** vendor daemon is called `ilight.exe`.

port@host Licensing

Typical licensing for **FieldView** involves providing a `license.dat` file for each user workstation. This file contains instructions for FLEXIm on how to license **FieldView** and the location of the server machine on which the `ilight` licensing daemon is running. Large, multi-national customers who have many user workstations and many **FieldView** licenses would like to avoid having to modify all of the `license.dat` files each time a new **FieldView** version is installed. **FieldView** licensing supports the FLEXIm `port@host` construct to refer to the server machine directly, rather than by having to copy the license file to each machine that runs **FieldView**.

With this feature, a `license.dat` file is not required on each user workstation. Instead, an environment variable called `ILIGHT_LICENSE_FILE` is set to point to the server by providing the port number over which FLEXIm communications occur and the name or IP address of the server machine running the `ilight` daemon.

Getting the Port number

Default installations of **FieldView** use one of two port numbers for FLEXIm:

Linux Systems = 1602

Windows Systems = 7788

If your installation has been customized, a different port number may have been selected and you will have to find out what the setting is.

Setting the `ILIGHT_LICENSE_FILE` environment variable

Use whatever method that you normally use for setting environment variables on your system. The name `ILIGHT_LICENSE_FILE` must be all upper case letters. Note that Flexera also provides a more general environment variable, `LM_LICENSE_FILE`, but this may already be in use for other vendors. In that case, the environment variable can be set for multiple vendors by appending the **FieldView** setting to the list. Sequential arguments in environment variables are separated by colons (`:`) on Linux and semi-colons (`;`) on Windows.

Licensing Issues and Problems

When I start the license server and run the command `fv_server status`, I get an error message similar to the following:

```
License server status (License file: /usr/fv/data/license.dat):

(machine): Cannot connect to license server (-15,12:146)
Connection refused

Vendor daemon status (on machine):

ilight: Cannot connect to license server (-15,12:146) Connection
refused

Feature usage info:

Users of fv: Cannot get users of fv: No such feature exists (-
5,222:146) Connection refused
```

This may be due to an error in the `SERVER` line provided to you. Run the command `fv_server id` again and confirm that the FLEXlm server name of your machine matches that in the `license.dat` file. Send this to support@fieldviewcfd.com.



Note: If `fv_server` script cannot be found in the `/bin` subdirectory of the directory where **FieldView** is installed, it probably means that the user answered no to the question Do you want to install the floating license manager (answer `y` for yes)? while running the `install.sh` script.

When I start the license server and run the command `fv_server status`, I get an error message similar to the following:

```
License server status (License file: /usr/fv/data/license.dat):

(machine): license server UP (MASTER)

Vendor daemon status (on machine):

ilight: Cannot connect to license server (-15,12:146) Connection
refused

Feature usage info:

Users of fv: Cannot get users of fv: No such feature exists (-
5,222:146) Connection refused
```

This may be the result of an error in the `INCREMENT` line provided to you. Run the command `fv_server id` again and confirm that the FLEXlm host ID of your machine matches that in the `license.dat` file. Send this to support@fieldviewcfd.com.

My license server went down and when it was brought back up, FieldView would not run.

When a license server goes down, the daemons which controls licensing will stop. If the license server stops issuing licenses for any reason, execute the command `fv_server stop`, make sure the processes called `ilight` and `lmgrd` are *not* running (through the `ps -elf |grep ilight`, etc. command). If they are running, stop them. Then issue the command `fv_server start`. Check the status with the command `fv_server status`. If the license server is not up, or is but you cannot run **FieldView** anyway, please contact support@fieldviewcfd.com.

I started the server but when I try to run FieldView, I get a message similar to the following:

```
client: Local checkout filter rejected request (-73,125)
```

This is usually due to an error in the `INCREMENT` line. Specifically, make sure that there is a *space* before the continuation characters (“\”) on lines that contain them and that there are *no* spaces *after* these continuation characters. See [Licensing Issues and Problems on page 51](#).

Does running FLEXlm require root permissions?

No. The FLEXlm documentation specifically states that there is no part of FLEXlm, `lmgrd`, vendor daemon or application, that requires root permissions. In fact, it is strongly recommended that you do not run the license server (`lmgrd`) as root, since root processes can introduce security risks. If `lmgrd` must be started from the root user (for example, in a system boot script), FLEXlm recommends that you use the `su` command to run `lmgrd` as a non-privileged user.

```
su username -c "/path/lmgrd -c path/license.dat -l /path/log"
```

where `username` is a non-privileged user, and ‘`path`’ are the correct paths to `lmgrd`, `license.dat`, and debug log file. You will have to ensure that the vendor daemons listed in `/path-to-license/license.dat` have execute permissions for `username`. The paths to all the vendor daemons in the license file are listed on each `DAEMON` line.

Does FieldView support redundant license servers?

Yes. Using a redundant license is not our standard procedure and is handled on a case-by-case basis. The requirement is that server information for 3 (three) different machines be given to us. All three servers need to be up to start the server and two of the three need to be up for the license server to operate. That is, any *one* server machine may fail without affecting licensing. However, if *two* of the machines fail, the license server will go down. The first of the three will be considered the controller. These require FLEXlm to be installed. There is a *separate* FLEXlm install script (called `installcd_flexlm.sh`). Follow the directions for how to install the FLEXlm license

server *only* (see [page 41](#)). Once you have the licensing software installed, the command

```
./fv_server id
```

is run from the `/bin` directory of the installation area for all three machines and this information is provided to **FieldView CFD** and a redundant license is requested (so that there is no confusion). A redundant license will then be issued. This has 3 `SERVER` lines instead of only one.

I edited the license file and ran an `Imreread` command, but the changes do not seem to have taken effect?

When the `license.dat` file is changed (edited), you need to stop and re-start the server. An `Imreread` will *not* always accomplish the desired effect. Run the following commands in the `/bin` directory of the **FieldView** installation:

```
./fv_server stop
```

```
./fv_server start
```

I stopped a FieldView session, and then tried to run FieldView again. Why did it say there were no licenses available?

It takes a few minutes for the license to become available again. This delay may be as long as 10 minutes. Please try again in a few minutes.

Do I have to use a hosts file so that my local installations of FieldView can contact the license server?

No. Instead of the `SERVER` hostname, the IP address can be used in the `license.dat` files. If this is done, *all* copies of the `license.dat` file, even the one on the `SERVER`, must be changed. In this case, the `SERVER` line might look like this, assuming the machine `wing` has the IP address `192.168.2.27`:

```
SERVER 192.168.2.27 690f23e5 1602
```

Can I run FieldView at home using my license from work?

It depends.

Assuming that you copied the `license.dat` file from your license server at work (see previous question), then there are two situations where this will work:

1. The server (at work) is accessible via the Internet. This is not very common, but it is the easiest way to work at home.

2. You can `telnet` to a machine at work, and then port the `DISPLAY` to your home system. In this set-up, you will be running the Linux version of **FieldView** at work and the panels and images would be sent across the network connecting your home system to the work machine. This is difficult because you need to know what your home `DISPLAY` is known by to the outside world. That is, when you set the `DISPLAY` environment variable on the work machine to display **FieldView** on your home system, you need to provide an IP address or machine name that would be recognized by the machine at work. You *also* need an Xserver running on your home system. Running in this way is *strongly discouraged* and will be extremely slow.



Important Note: If you lose or drop the connection to the remote license server, **FieldView** will detect that it has lost the connection to the license server (this may take 10 - 15 min). A warning will be printed telling you to save your work. Five (5) minutes later, if a connection cannot be re-established, **FieldView** will exit.

If you cannot reach the license server, another option exists. Please refer to [Licensing via Server](#) for information.

Post-Installation Tasks

Configuring your machine (Linux only)

If you want to be able to run **FieldView** by typing just `fv` (rather than the entire path of the startup script), you can add the directory containing the startup script to your command search path in your `.login`, `.cshrc`, or `.profile` files as appropriate. This step should be repeated by every **FieldView** user. This is also needed in order to use the MPEG converter.

You also need to make sure the X environment variable `DISPLAY` is set. For users of the C shell (`/bin/csh`), the following should appear in the `.login` or `.cshrc` files:

```
setenv DISPLAY :0
```

For users of the Bourne shell (`/bin/sh`) or Korn shell (`/bin/ksh`), the following should appear in the `.profile` file:

```
DISPLAY=:0  
export DISPLAY
```

If you are running locally, and the `DISPLAY` environment variable is set to something other than `:0`, performance may be seriously degraded. The above commands are correct unless you are logging in remotely. In that case, you will need to reset the `DISPLAY` variable to reference the remote display.

Modifying Your Print Script (Linux only)

In the `fv/bin` subdirectory, there is a shell script called `fv_to_printer.sh.sample`. In order to instruct **FieldView** to send your file directly to the printer, you must edit this shell script, and rename it to: `fv_to_printer.sh`. The supplied script can be used to send the PostScript file to any output device. Example print commands are shown in the script.

Locating the **FieldView** Tutorial Files (Windows and Linux)

In order to work through the tutorials contained in the **User's Guide**, we recommend that you obtain copies of the tutorial data and associated restarts. These files can be either downloaded from the **FieldView** Customer Center at my.fieldviewcfd.com (see Tutorials) or copied from the tutorial directory from the **FieldView** DVD distribution to a local disk. A broader explanation of the location, sizes and content of this tutorial directory can be found in the section called Tutorial Introduction found in the **User's Guide**. These files are optional, and do not need to be installed in order to run **FieldView**. To copy them from the DVD:

1. Insert the **FieldView** DVD-ROM.

2. Change to the tutorial subdirectory on the DVD.
3. You may copy any of these files from the DVD to your hard drive, or you may read the data into **FieldView** directly from the DVD.

Locating the KIVA Translator File (Windows and Linux)

KIVA data cannot be read directly into **FieldView**. The output of this solver has to be translated into a format that **FieldView** can read. The necessary translator, source code (where needed) and documentation has been provided on the **FieldView** DVD-ROM. Copy the translator directory for KIVA to your hard drive. This file is optional, and does not need to be installed in order to run **FieldView**. To install the translator:

1. Insert the **FieldView** DVD-ROM.
2. Change to the translators subdirectory on the DVD-ROM.
3. Copy the directory to your hard-drive.