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

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. Use Help.. FieldView Licensing to enter your license or activation code
4. Complete your post installation tasks

Following a successful installation and startup, **FieldView** should look like this:
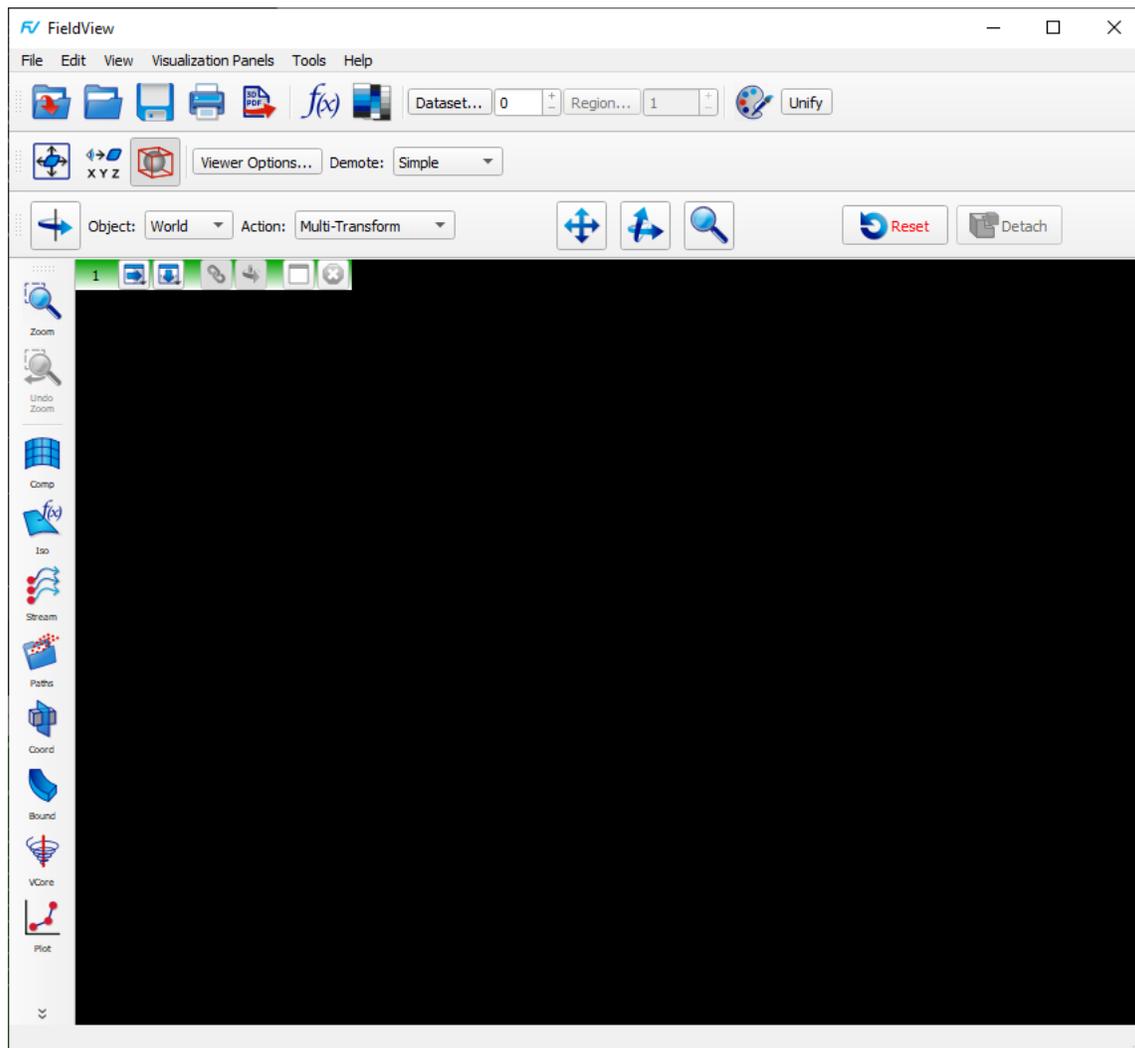
The rest of this **Installation Guide** provides detailed information on how to install and customize **FieldView** to meet your specific needs.

Certified and Supported Platforms

Certified platforms are defined as follows:

The **Tecplot** 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 2023. See **tecplot.com** to find the most up to date platform coverage information.

Certified & Supported FieldView Platforms			
System	**Platform**	**OS**	**Client, Server, Parallel**
APPLE MAC	INTEL	Big Sur	FV 2022 CERTIFIED
		Monterey	FV 2022 SUPPORTED
APPLE MAC	M1	Monterey	FV 2022 SUPPORTED (Rosetta 2)
System	**Platform**	**OS**	**Client, Server, Parallel**
LINUX	AMD64/ EM64T/ Xeon-64	CentOS 7	FV 2022 CERTIFIED
		CentOS 8	FV 2022 SUPPORTED
		Red Hat Enterprise Linux 6	FV 2022 SUPPORTED
		Red Hat Enterprise Linux 7	FV 2022 SUPPORTED
		Red Hat Enterprise Linux 8	FV 2022 SUPPORTED
		Debian 10	FV 2022 SUPPORTED
		Debian 11	FV 2022 SUPPORTED
		Ubuntu 18.04	FV 2022 SUPPORTED
		Ubuntu 20.04	FV 2022 SUPPORTED

System	Platform	OS	Client, Server, Parallel
WINDOWS	AMD64/ EM64T/ Xeon-64	Windows 8.1	FV 2022 SUPPORTED
		Windows 10	FV 2022 CERTIFIED
		Windows 11	FV 2022 SUPPORTED

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@tecplot.com**.

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

Linux

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 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 **Tecplot, Inc**.

NOTE: FieldView can optionally be modified to use an MPI (for parallel servers) which is different than what is provided with the installation. Please refer to **Replacing FieldView's MPI (Optional) on page 9**

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.

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 `fv2022_win64_setup.exe`

2. You are prompted for the directory to install the software. By default, it is installed in the directory:

   ```
   C:\Program Files\Tecplot\FieldView 2022
   ```

   We *strongly recommend* that you accept this default location. Installing to a directory name using UTF-8 / multi-byte characters is not supported.

After installation is complete, a Start menu program group will exist which contains the **FieldView** icon, a link to Copyrights and to several PDF documentation files.

macOS

The installation process for FieldView on mac systems is standard for macOS, where you will be presented with the following prompt when opening the .dmg file:
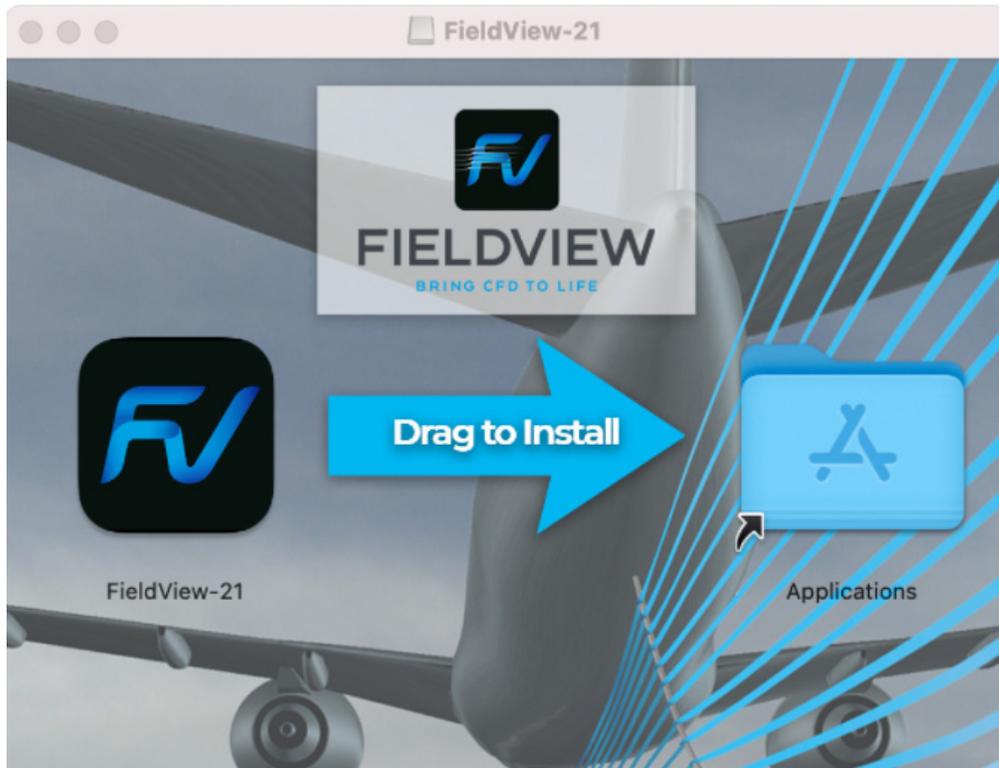
Replacing FieldView's MPI (Optional)

FieldView and all of its components come ready to run in parallel with an included version of MPI (Open MPI 1.6.2). If you'd like **FieldView** to use a different version of MPI, **Tecplot Inc.** has developed an easy solution with myFVMPI.

FieldView installations provide **myFVMPI** for HPC users who want to run a particular version of MPI which has been tuned for performance optimization on their HPC system. All they have to do is follow the simple steps below for building and installing myFVMPI with their MPI. There is no need to rebuild any other component of **FieldView**. Once it is done, **FieldView** will run the **FieldView** engines in parallel using that particular MPI instead of the default Open MPI 1.6.2.

Building myFVMPI

Refitting FieldView with an alternate MPI is accomplished by creating a new version of the installed mpirun script and the libmyFVMPI.so library in the ./plugins/lib directory, backing up each with the ".orig' file extension.

NOTE: Using myFVMPI requires licensing for at least **FieldView** Parallel 32.

1. Make `$FV_HOME/myFVMPI` the current directory.

2. Set up your environment appropriately for using your MPI. In particular, the `mpicc` command should be in your path and usable for compiling MPI programs. The `mpiexec` command should be in your path and usable for running MPI programs.

3. Examine the value of `MPICC_CFLAGS` in the file `Makefile`. Change this value as needed for compatibility with your `mpicc` command.

4. Run the following command to build myFVMPI:

        make

5. Optionally, run the following command to test myFVMPI:

        make test

    This will execute the `pmandel` example program (using two processes on the local machine) and display an image created by this program to confirm the test is passed, as illustrated in **Figure 1**.
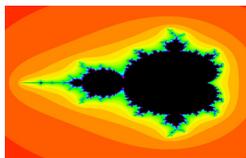

Figure 1: Image displayed to confirm the myFVMPI test is passed

Installing and Using myFVMPI

1. Run the following command to replace the default **FieldView** myFVMPI library and `mpirun` script with those you built using your own MPI:

    ```
    make install
    ```

    The first time you run '`make install`', it will backup the default **FieldView** myFVMPI library and `mpirun` script as they are updated.

    *Note that myFVMPI has been tested with several MPI implementations. Scripts **mpirun** and **fvrunsrv** are tuned for OpenMPI 1.6.2. Regarding the MPI parameters found in FieldView's scripts:*

    *   *For OpenMPI 3, the option --oversubscribe must be added to last line of the mpirun script. Without that, parallel servers will fail with the following message:*
        ```
        There are not enough slots available in the system to
        satisfy the 5 slots that were requested by the application:
        <FVpath>/bin/fvsrv_par
        ```
    *   *For MPICH and MPI Intel, the --mca arguments as provided must be removed from the script fvrunsrv, and equivalent settings should be used for your MPI. For more information on MPI options please refer to the README.txt in the myFVMPI directory.*

2. You should now be able to run **FieldView** and use the Parallel engine with your own MPI via the myFVMPI library and see FieldView report the MPI which is being used upon startup, e.g.:
    ```
    myFVMPI was built with: <path>/openmpi-<version>/bin/mpicc
    Note: The system on which this application is currently
    running may have MPI installed in a different location.
    ```

3. If you want to perform this operation again for another MPI, run the following command to remove the myFVMPI library and mpirun script you previously built:

    ```
    make clean
    ```

    ...and go to 3. of **Building myFVMPI**.

Restoring the default **FieldView** myFVMPI library and mpirun script

1. Run the following command to restore the default **FieldView** myFVMPI library and `mpirun` script (from the copies backed up by '`make install`'):

    ```
    make uninstall
    ```

Python Installation (Optional)

FieldView is designed to use Python versions 2.7 or 3.1. Python is not required to run **FieldView**. In order to use Python with FieldView, some initial set up is required.

Linux: Running **FieldView** using Python 2.7 or 3.1 (**without tkinter** module for panel support), requires a link which points FieldView to your installed Python.For example, if python is installed in /usr/lib64::

```
ln -s /usr/lib64/libpython2.7.so <FV Install Dir>/fv/lib/linux-
_amd64/libfv_python_link.so
```

Running **FieldView** using Python 2.7 or 3.1 **with tkinter** support:

- Download source code package for Python from python.org, build against Tcl and Tk libraries which are provided within FieldView installations. For example, if using Python 2.7:
    1. `./configure --prefix=/<2.7 Python Installation Dir> \`
       `--with-tcltk-includes=-I/<FV Install Dir>/fv/lib/tcltk-\`
       `includes --with-tcltk-libs==-L/"FV Install Dir"/fv/lib\`
       `/linux_amd64/lib --enable-shared`
    2. `make`
    3. `make install`
    4. Create symbolic link:
       `ln -s /<2.7 Python Installation Dir>/lib/libpython2.7.so \`
       ` <FV Install Dir>/fv/lib/linux_amd64/libfv_python_link.so`

  • Then run FieldView normally.

  Note: Environment variable **FV_PYTHON_VER=3** must be set to use FieldView with Python 3.1.

Windows:

- Download 2.7 or 3.1 (only) 64bit package from python.org and install
- Set environment variable **FV_PYTHON_VER=3** if using Python 3.
- Then run FieldView normally

macOS:

Python 2.7 is the version installed on current macOS platforms, and no additional steps are required to use that version with **FieldView,** and version 3 Python is not supported by FieldView on this platform.

Installing FieldView Servers on remote systems (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 separately for installation on remote systems for use as a back end.

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.

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 18**. Further steps (described in **Step 3 – Set up a Server Configuration File on page 19**) are needed to make Client-Server operation functional.

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. (Note that FieldView Client installations provide parallel servers.)

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 provides and uses a specific version of Open MPI but allows users to select different versions of MPI (for improved performance, etc.).

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 13** 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 fv_type_tar.Z | tar xvf -
```

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

FieldView provides a specific version of Open MPI. If an alternate version or implementation of MPI is preferred, please refer to **Replacing FieldView's MPI (Optional) on page 9.** Note that using myFVMPI requires licensing for at least **FieldView** Parallel 32.

Testing & Configuring FieldView Servers

For local servers running on the same system as the client, no additional configuration steps are required. The "**Local**" serial and parallel server options will appear automatically on the Data Input menu. To read data on **remote** servers, you can run the server(s) manually, as Manual Server Startup on page 34 in the Users Guide, or create 'AutoStart" Server Configuration files (.) for your FieldView Client. Depending on your network, the configuration file needed may be as simple as the following:

```
! The '!' character at beginning of line is for comments.
ServerName: <hostname or IP>
Server Directory: <folder or directory containing command fvrunsrv>
! Port forwarding may not be necessary, depending on your network.
Port Forward: true
! See a full list of keywords and their meanings in the following pages.
```

```
! Additional settings may be required, depending on your network, and
! Fieldview server type (ie. serial or parallel)
```

FieldView will look for Server Configuration files in the *sconfig* folder of your installation, but can be redirected to another location with the environment variable FV_SERVER_-CONFIG_DIR.

Testing server connectivity in your environment

The following three additional steps can be performed to ensure **remote** FieldView Client-Server operation is functional:

Step 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.

Step 2.   Copy the proper server `tar` file to the server machine(s). Unpack the server executable from the `tar` file. (On servers which have the FieldView Client installed, this step will not be necessary, since serial and parallel servers are already installed.)

Step 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. If all three of the steps listed above are completed successfully, **FieldView** Client-Server or **FieldView** Parallel should be expected 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	`ping server_name` or `ping server_IP_address`	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. Secure remote shell programs *ssh* or *plink (Putty)* can be used

to log in to the server machine from the client. If you are unsure about the details of using either these protocols, it is highly recommended that you talk to a system administrator about your plans before proceeding, as each Customer site is different in terms of security policies which govern the client and server systems.

Test procedure - This test must succeed. (With Windows, you will need to run your *secure remote shell* command using the CMD prompt):

Test	Command	Expected Result
2	`[ssh	plink]` *server_name* `-l` *login_name date* Note: *server_name* can be replaced with the server IP address.	Today's date is printed

The above test must print the date **without** a prompt for a password for the test to be a success. If successful, you can use Automatic Server Start (`AutoStart:`) in a *Server Configuration File* (.), and proceed to Test 4 below. If you are not successful, then you may not be able to use Automatic Server Start. See your System Administrator to have **passwordless login** authentication set up for your user.

If you need a password to log in at the server, you can *still* use FieldView Client-Server *in* **Manual Server Startup** mode, as described under the heading of the Users Guide, given that you can execute the following two steps.

Now connect to the server, and test connectivity from the server to your client system, to ensure the FieldView Server will be able to communicate back to the FieldView Client:

Test	Command	Expected Result
3	`[ssh	plink]` *server_name* or *server_IP_address* `-l` *login_name* You are prompted for *password*	You are logged in to the server machine

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

Test	Command	Expected Result
4	`While logged in with ssh or plink as per test 3, above, execute:` `ping` *client_name* or `ping` *client_IP_address*`)`	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	On client machine: `contest` *port_number* On server machine: `telnet` *client_name port_number* Note: Use the IP address returned by `contest`	Successful connection message printed in `telnet` window.

Details: 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 **tecplot.com** and unpackage the compressed tar file:

    ```
    cat <path_to_your_server_package>/fv<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 – Set up 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 `.` extension) is what appears on the server selection menu. We provide one default server configuration file for Linux installations. This is called `local.` 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 *hostname* or an *IP address*	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 `-np` 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 `[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 `[ssh|plink]`. Does not apply to Windows.	`ssh` (`rsh` was the former default)
ServerProgram	The complete path of the server program (`fv`) 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 *hostname* or an *IP address*.	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` or `plink` 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 parallel

The server config file (`.`) 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 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 (`.`) 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 (`.`) 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.

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 neces-

sary 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 Miccrosoft 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

Entering Your License

The first time you launch FieldView after a fresh installation, you will be prompted to enter your license information:
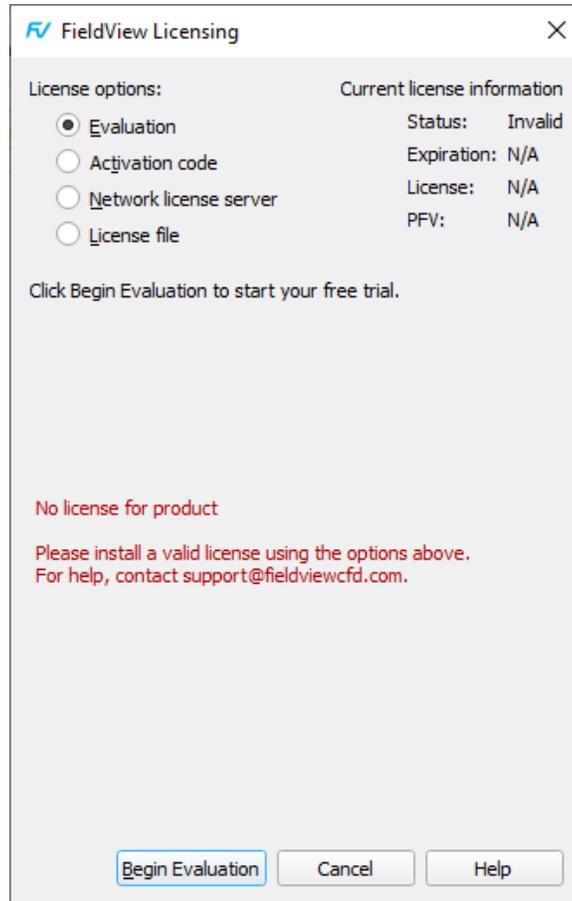
You will also see this dialog if FieldView cannot validate your license information (for example, because your evaluation license has expired, or because the network license server is not available). Additionally, you may change your license information at any time from the **Help** menu.

SYSTEM ADMINISTRATORS

Tecplot, Inc.products store their license configuration file in a user and platform specific location by default.

Windows: C:\Users\<username>\AppData\Local\Tecplot

Linux: ~/.local/share/Tecplot

Mac: ~/Library/Application Support/Tecplot

The above directories are specific to the individual workstation users. If you want all users of the computer to use the same license configuration for a given Tecplot, Inc. product, you may locate the license file (.lic) in a more general location during or after initial license setup so that users can select the *License File* option on the License Dialog at startup. This will automatically create a copy of `fieldview.lic` in the users home directory, described above.

The license file can also be placed in the top level of the FieldView installation itself to allow the application to run regardless of which user or host is launching that particular installation of FieldView. This is especially useful for network licensing.

In addition, network licenses can be accessed with an environment variable, or with more general content in the file `fieldview.lic`, as noted in the port@host Licensing section of this document.)

Note: Each Tecplot, Inc. product has its own license configuration file. To prevent users from editing the license configuration, you may then change the permissions on the license file to be writable only by an administrator or root user.

How you install your license information depends on what type of license you have.

If you wish to evaluate the product before purchasing, you may obtain an **Evaluation** license with a single click. See **Evaluation License Setup**.

If you have an Activation Code for a **Single-System** license and have Internet access, you enter the activation code. See **Single-System License Setup Using An Activation Code**.

If you have a License File for a **Single-System** license, you select the license file. See **Single-System License Setup Using A License File**.

If you have a **Network** license, you specify the server name and port number of the RLM license server on your network. See **Network License Setup**.

Evaluation License Setup
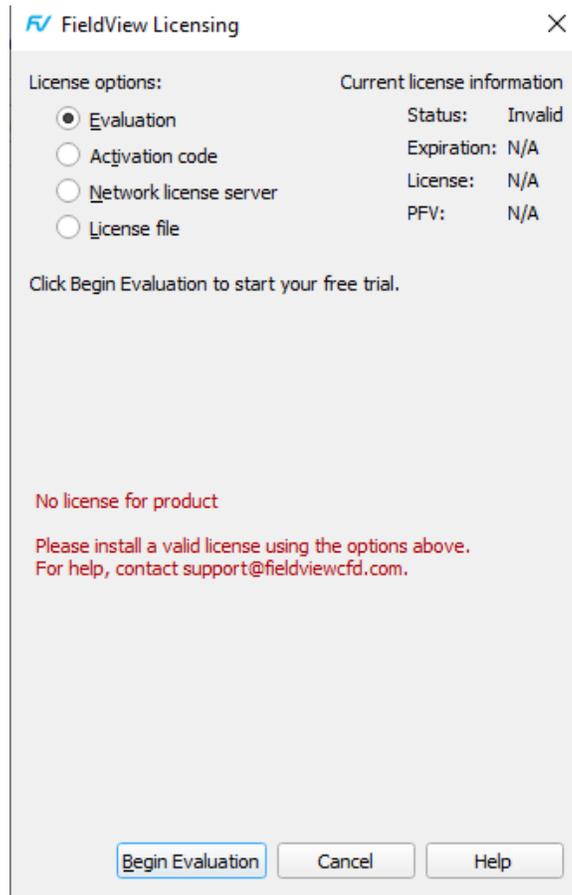
To obtain a time-limited evaluation license key so you can try the product before purchasing, make sure the "Evaluation" license type is selected in the Licensing dialog, then click **Begin Evaluation**. The product evaluation period begins. You will see a notice on the product's Welcome Screen indicating when your evaluation license will expire.

This procedure requires an active Internet connection. If for some reason you cannot fully evaluate FieldView on an Internet-connected computer, or if you need additional time to complete your evaluation, contact FieldView Support for a time-limited Single-System license file, which may then be installed using the instructions in **Single-System License Setup Using A License File**.

Single-System License Setup Using An Activation Code

If you received an activation code for your product, allowing it to be activated over the Internet, follow the instructions below. If you received a license key file, or if the workstation you have installed the product on does not have Internet access, see **Single-System License Setup Using A License File**


To activate the product, simply enter or paste your activation code in the field provided. Then click **Activate**. The license key corresponding to your activation code is downloaded to your computer, and FieldView uses that license

Single-System License Setup Using A License File

You will need the license file sent to you by Tecplot, Inc. as an e-mail attachment. Please save it to a file on your computer in an easy-to-find location, such as the Desktop.
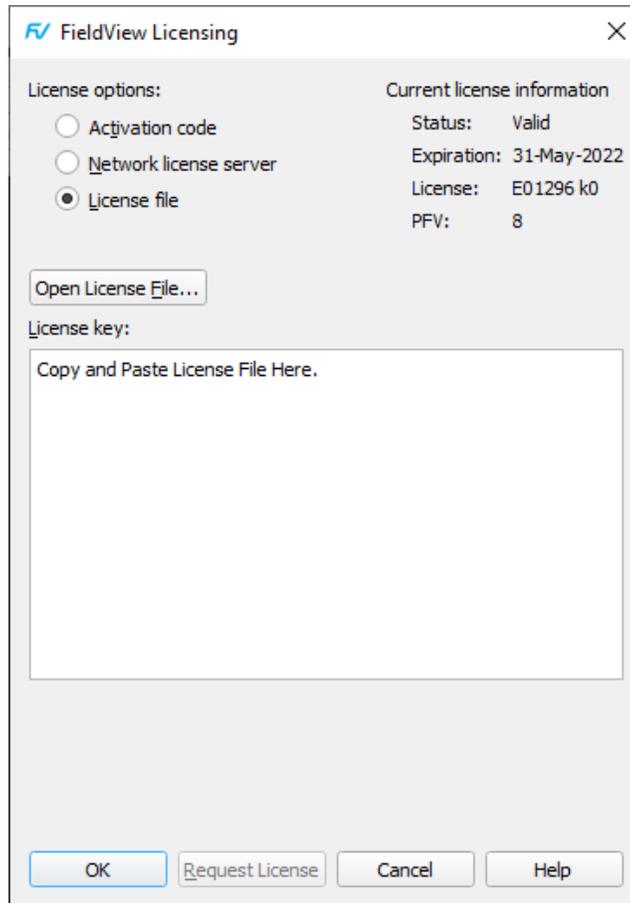
> If you do not have your license file, or have lost it, click *Request License* in the Licensing dialog for instructions on how to obtain it.
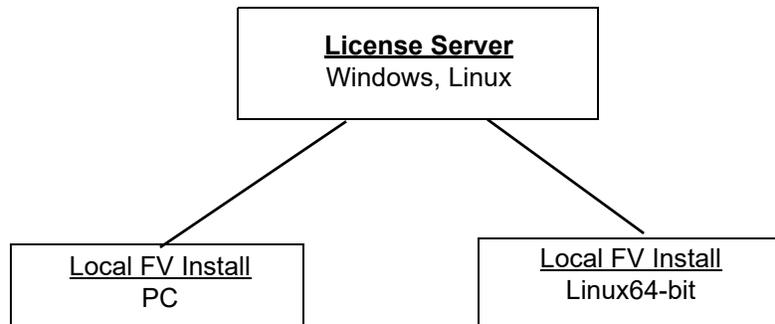
1. Make sure the "License File" option is selected in the Licensing dialog.

2. Click Open License File. A file browser appears. Select the license file e-mailed to you, then click Open. Alternatively, you can Copy and Paste the contents of the license file to the available field. The message "Status: Valid" appears in the Current License Information panel at the upper right.

3. Click OK to save the license information.

FieldView now uses the chosen license.

Note: The file dialog will show a default location for the License file. For multiple users, the file can be located in a general location
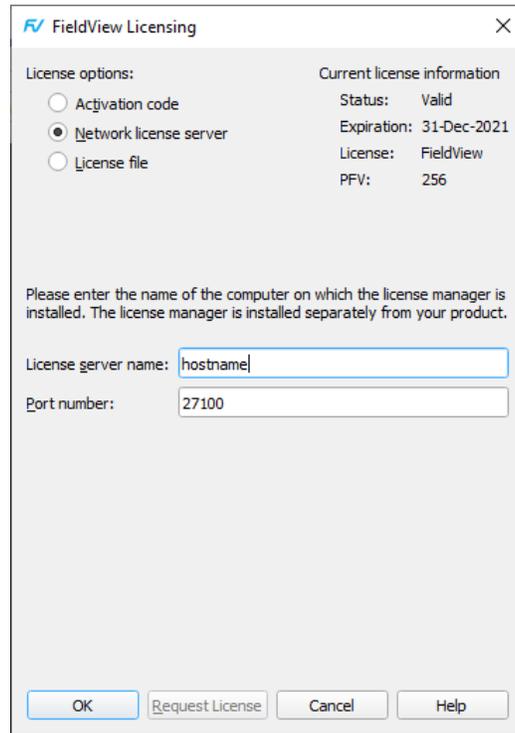
Network License Setup

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.

```
┌──────────────────────────┐
│     License Server       │
│     Windows, Linux       │
└──────────────────────────┘
```

```
┌─────────────────────┐          ┌─────────────────────┐
│   Local FV Install  │          │   Local FV Install  │
│        PC           │          │     Linux64-bit     │
└─────────────────────┘          └─────────────────────┘
```

The **Local** machines have **FieldView** installed. The User directory of the local FieldView installation typically provides a license file configured to access FieldView licenses from the license server, and was created either manually, or with an activation code.

Figure 2: 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 `fieldview.lic` 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.

To use a network license, you or your system administrator must first install the Reprise License Manager (RLM) and the license key for your product on a network server. (See the installation instructions included with the RLM download, available (on the FieldView License Manager download page.) Once this has been accomplished, follow these steps:

1. Make sure the "Network" license type is selected in the Licensing dialog.
2. Enter the server name (you may use a hostname or IP address) and port number of the license server in the fields provided.
3. Click **OK** to save the license information.

FieldView will now use a license obtained from the license server.

If the product is unable to obtain a license key, an error message will appear, and you should verify that you have specified the correct network license information.

RLM_ROAM

You can initiate roaming from the command line by setting the RLM_ROAM environment variable to the number of days you wish to roam, then starting FieldView from that command line session. The roaming session expires at the end of the day after the number of days specified (for example, if it is noon Tuesday, and you set RLM_ROAM to 2, the roaming license expires at the end of the day Thursday). RLM_ROAM may also be set to the special value **today** to have the roaming license expire at the end of the current day (the value **0** means that no changes to roaming will be made).

If RLM_ROAM is set permanently (for example, using the System Properties in Windows), the license roaming will be refreshed each time you run FieldView while connected to the network. If RLM_ROAM is 1, for example, each time you start FieldView, the roaming license is refreshed to expire at midnight the following day. This allows you to easily take your computer home with you any night and continue using FieldView while disconnected from the license server, without needing to explicitly roam each day.

You may end a roaming session early by setting RLM_ROAM to -1 before launching FieldView.

> If you are having trouble ending your roaming early (a common cause is upgrading FieldView while roaming), try setting RLM_ROAM to -100 before launching FieldView. The roaming license will be forcibly removed from your workstation, and FieldView will begin requesting a regular license from a license server at each startup. At this point you can obtain a fresh roaming license, if desired. The reservation for your previous roaming license will, however, continue to be held on the license server that issued it until it expires.

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** 2022 with provision for 3 concurrent users, including support for **DataGuide**^{TM} (dg) and standard support for up to 8 Parallel worker processes (pfv8):

```
HOST computername hostid 27100
LICENSE teclmd fv yyyy.mm dd-mmm-yyyy 3 hostid=Any
customer=<number>  options=8 _ck=741e7266d6 sig="c
2N251d=E2AHbpk7pRIFU2v6g6CAxKP*0igJBP59o+v=c0By9*fjC8wa~sh~B0"
```

Where <number> is the FieldView License Identifier for the given Customer

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 3**, 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).
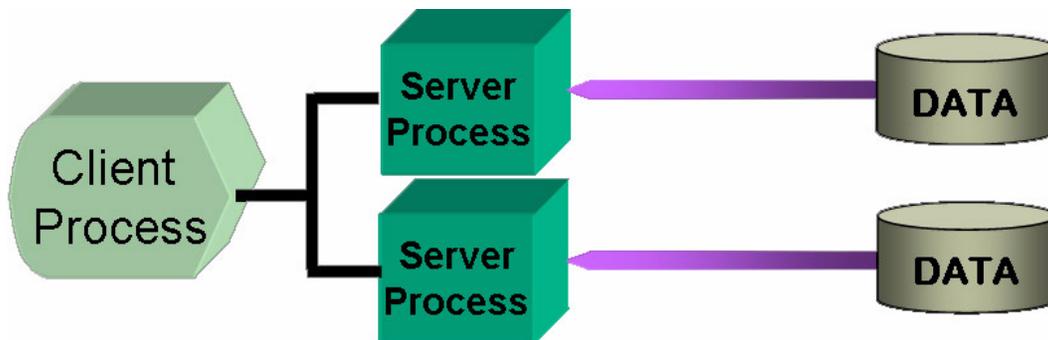

Figure 3: FV Client using two of an unlimited number of Server Processes
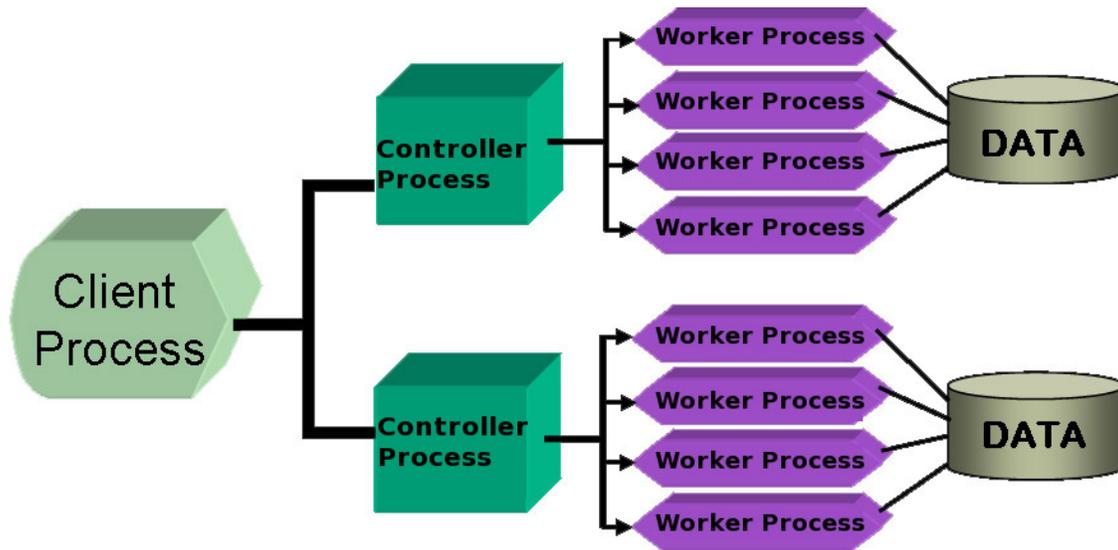
Figure 4: FV Parallel with Licensed Controller Processes

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 4**), 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 limitations are that a GUI cannot be used in batch mode, so **FieldView** must be run from a script, and that you FieldView will use *software graphics* rather than accelerated graphics. With a batch-only license you can run **FieldView** from a command line for either type of scripting, 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 attempt to 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 with accelerated graphics (if available) 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 64 and beyond 64 (HPC option). Depending on your licensing options you will have one or more of the following `LICENSE` lines present in your **FieldView** `fieldview.lic` file, and the **batch-only** option will be shown as in the following example:

```
HOST computername hostid 27100
ISV teclmd teclmd teclmd.opt 27101
LICENSE teclmd fv yyyy.mm dd-mmm-yyyy 3 hostid=Any
customer=nnnnn options="8 batch-only" _ck=741e459b05 sig="c2W
25g3U*q11YagwyfbRvbC0ScKnCdcbd08n0dmrf2gYfN8wC1cZLemIdtgIWAjA"
```

Where <number> is the FieldView License Identifier for the given Customer

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 5** 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.
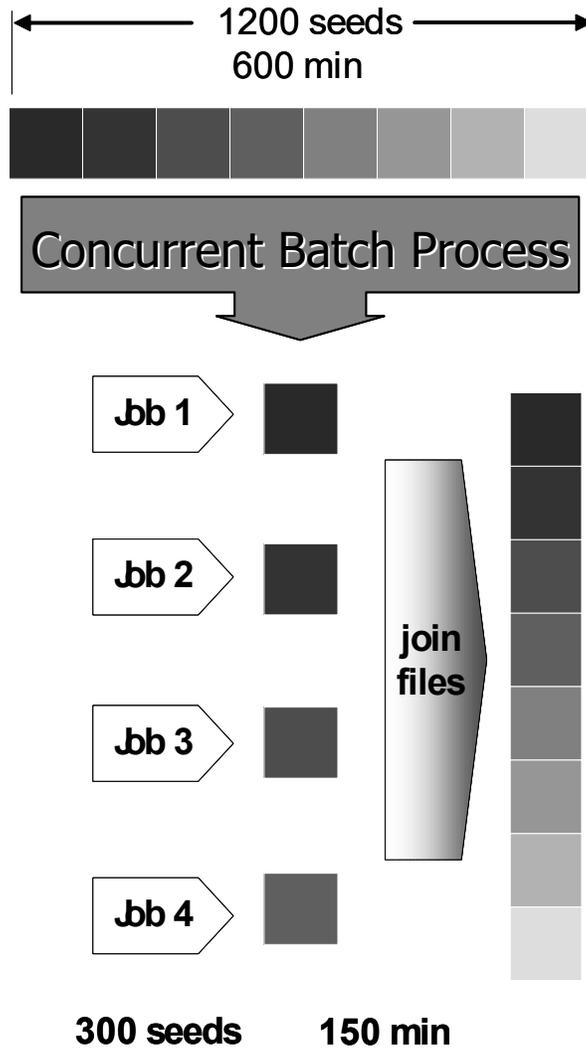
The **FieldView** scripts needed to carry out this concurrent set of operations would look something like this:

Figure 5: Concurrent Batch Operation

```
! 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, or a range of ports (eg. `-port 12345:12445`) may be explicitly specified. Since all of the restarts in this example are attempting to read the same timestep initially, it may be a good idea to delay the start of each instance of **FieldView** using a `SLEEP` command.

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 6**.

Figure 6: 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 **Tecplot, Inc.** 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 and Other Software Using RLM

You can run multiple products that require RLM license management on the same server. This can be done by having multiple license files (.lic) in the same location for the license server, or by combining licenses (and VENDOR lines) in the same license file.

Note that current FieldView products require use of at least version RLM 11 License Management Software.

port@host Licensing

Typically, the procedure for licensing **FieldView** is described in the above sections, but large, multi-national customers who have many user workstations and many **FieldView** licenses can entirely avoid use of the Graphical User Interface (GUI) for setting up access to a **network license.** This is generally done in one of two ways.

1. **License File -** The installed license file `fieldview.lic` can be installed such that the file contains only the single line shown below, and the port corresponds to the port being used for RLM in the server's copy of the license file.
    ```
    HOST <hostname or IP> 0 <port>
    ```

2. **Environment Variable -** An environment variable can be set on the system running FieldView, which again, points to the port and server used for licensing, for example:
   ```
   teclmd_LICENSE=<port>@<hostname or IP>
   ```

   Note that this will not interfere with other vendors using the more general RLM_LICENSE environment variable on the same system.

To enable FieldView to search for licensing at more than one server, sequential arguments in environment variables are separated by colons (:) on Linux and semi-colons (;) on Windows. Note that FieldView considers the environment variable as well as the fieldview.lic files found in the locations noted above as it attempts to check out a license.

Redundant License Servers

Using a redundant license is not our standard procedure and is handled on a case-by-case basis, so as you obtain your network license(s) from us, please advise us of your need for this redundancy. Note that the *triad* license configuration available with previous versions of FieldView has been replaced with a different implementation of license server redundancy.

> 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.

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.

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.tecplot.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.

    4. Change to the tutorial subdirectory on the DVD.

5. 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.