

Email: techinfo@fieldp.com

URL: <http://www.fieldp.com>

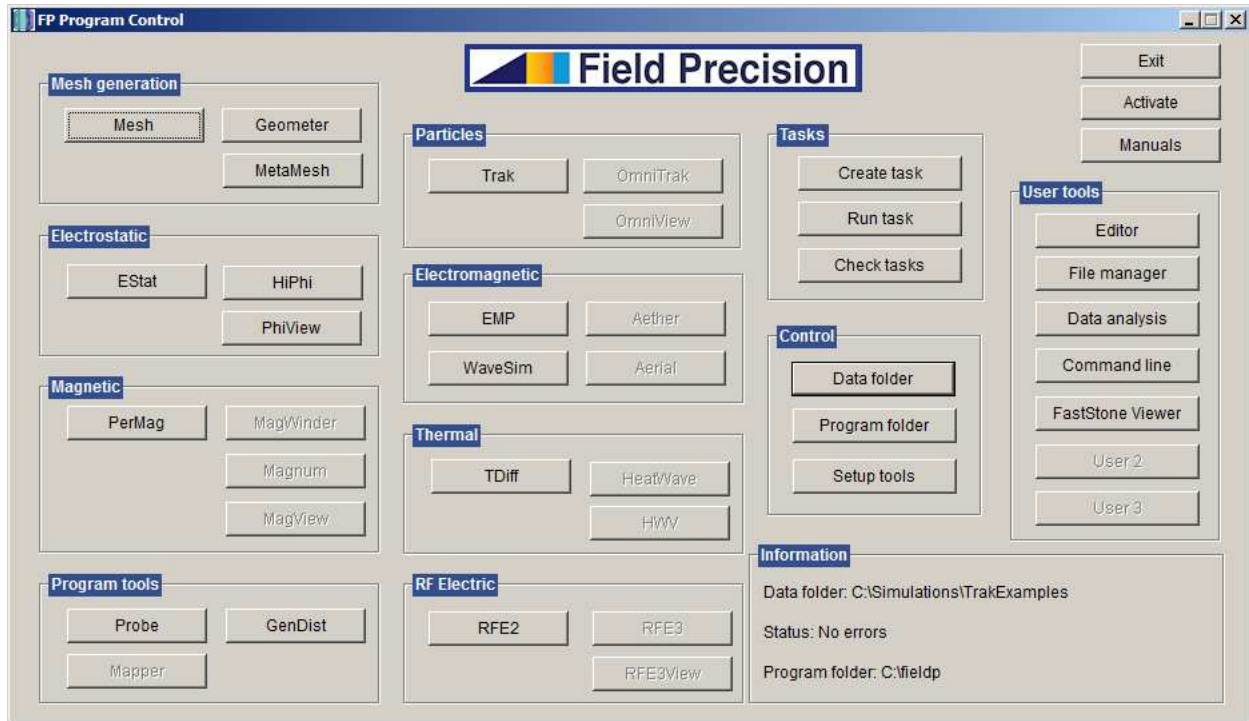


Figure 1: **FP Controller** display

General use

The first step to make calculations with **TriComp** (2D) or **Amaze** (3D) programs is to run **FP Controller**. The utility helps you organize your work in two ways:

- Only one shortcut is required on your desktop, minimizing clutter.
- All programs open in a specified data directory, eliminating redundant trips through the directory tree.

The installer for the package you purchased has created a shortcut to `fpcontrol.exe` on the Windows desktop and in the start menu. Figure 1 shows the initial program display. The controls are divided into twelve functional groups:

- Mesh generation
- Electrostatic solutions
- Magnetic-field solutions
- Charged-particle beam solutions
- Electromagnetic solutions
- Thermal solutions
- RF electric field solutions

- Program tools
- Tasks (automated calculations)
- Control settings
- Information display

The first seven groups reflect the activities of a **TriComp** or **Amaze** calculation: first create a mesh and then proceed to the physical solution and analysis. There are also buttons at the top-right to exit **FP Controller**, to activate the software license on your computer and to display instruction manuals for installed programs.

Note that some of the command buttons in Fig. 1 are not highlighted. The controller activates buttons only if it detects the corresponding program in the directories `c:\fieldp\tricom` or `c:\fieldp\amaze`. Buttons for **Mesh** and **MetaMesh** are always active, while the status of controls in the solution sections depends on the packages you have purchased. If no buttons are active, **FPController** is not pointing to the directory where you installed the software. In this case, click the **Program folder** button. In the dialog move to the directory `c:\fieldp` and click *OK*. For a valid setup, buttons for all the installed programs will become active.

It is a good practice to collect input and output files for related solutions in a specific data directory. Use the **Data folder** button to set the current location. Programs that are opened after the change will read and write to the directory. (Note that the setting does not affect previously-opened programs.) The **Information** area at the bottom of the dialog shows: 1) the current data directory, 2) the last operation performed and 3) the program directory. Table 1 summarizes the functions of the **TriComp** and **AMaze** programs represented by buttons.

User tools and instruction manuals

You may often use additional programs working with the **TriComp** and **AMaze** programs and other technical software. You can set up **FP Controller** to launch your favorite utilities. To define a text editor, press the **Set tools** button in the **Control** section (Fig. 1) to open the dialog of Fig. 2. Click the **Editor** button, then navigate to the appropriate directory and select the program. When you exit the dialog, the **Editor** button in the **User tools** section becomes active. You can follow a similar procedure to define two other utilities. For the **File manager** option, select a program like **Windows Explorer**. Use the **Data analysis** button for a spreadsheet, plotting program or mathematical analysis software. The **Command** button opens a terminal window if you want to run programs from the command prompt or under the control of your own batch files. There are all three additional buttons in the **Control** section for any other function you want. In the **Set tools** dialog, type in a title for the button and then choose the program.

You can open any of the instruction manuals supplied with our software packages from the **FP Controller** main window. Click the **Manuals** button to display the popup menu shown in Fig. 3. Then, click any item to open the document in your default PDF viewer. Note that the menu includes entries for all the **TriComp** and **Amaze** programs. **FP Controller** issues a message if a document is not available.

Table 1: Solution program functions

TriComp	
EStat	Electrostatics with dielectrics or conductors.
PerMag	Magnetostatics with coils, permanent magnets and soft magnetic materials.
Trak	Charged-particle transport and gun design.
RFE2	RF electric fields for RF heating and biomedical applications.
TDiff	Static and dynamic thermal transport in solid materials.
WaveSim	Frequency-domain electromagnetic solutions.
EMP	Time-domain electromagnetic solutions.
AMaze	
HiPhi	Electrostatics with dielectrics or conductors.
PhiView	Post-processor for HiPhi .
Magnum	Magnetostatics with coils, iron and permanent magnets.
MagView	Post-processor for Magnum .
OmniTrak	Charged-particle transport and electron/ion gun design.
OmniView	Post-processor for OmniTrak .
Aether	Time-domain and frequency-domain electromagnetic radiation.
Aerial	Post-processor for Aether .
HeatWave	Static and dynamic thermal transport in solid materials.
HWV	Post-processor for HeatWave .
RFE3	RF electric fields for RF heating and biomedical applications.
RFE3View	Post-processor for RFE3 .
Utility	
MagWinder	Created drive coil definitions for Magnum and Aether .
Probe	Utility to plot history records from TDiff , HeatWave , EMP and Aether .
GenDist	Utility to plot and to analyze distributions from Trak , OmniTrak and GamBet .

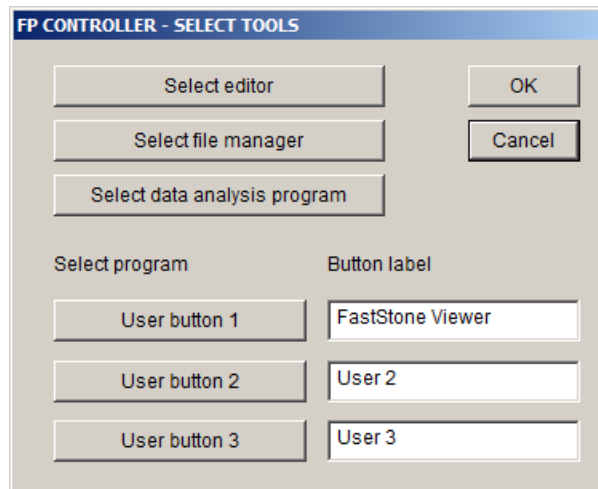


Figure 2: Dialog to set external tools.

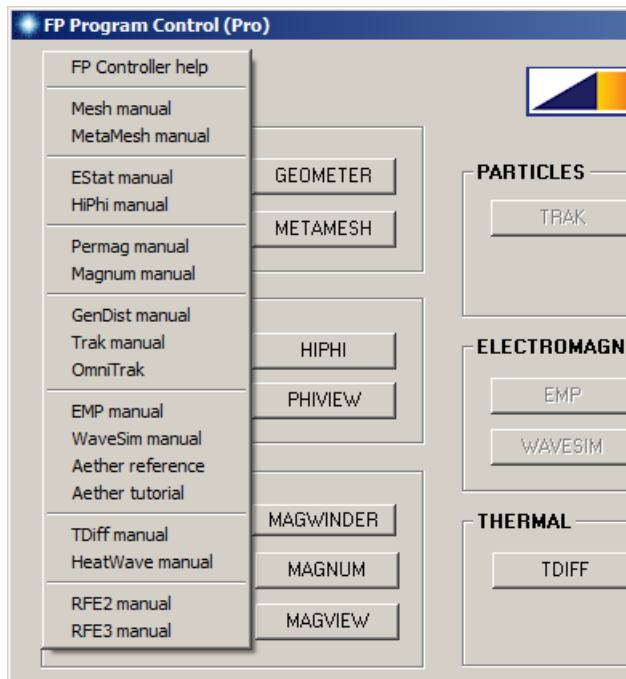


Figure 3: Program interface with the manual menu displayed.

Defining tasks for automatic operation

TriComp and **AMaze** programs are optimized for multicore personal computers. Solution programs support parallel operation. A second feature is the capability to run multiple independent calculations simultaneously. All solution programs can run in the background if launched from a batch file. Background operation is automatic and faster than running in a window. The **Create task**, **Run task** and **Check tasks** commands in the **Task** group make it easy to use batch files. With the commands you can 1) quickly define multistep calculations (*tasks*) in an interactive dialog, 2) launch simultaneous tasks in the background and 3) find out which tasks are running.

The **Create task** button calls up the dialog of Figure 4. In the **Task prefix** box, supply a name **fprefix** that indicates the function of the task. The task information will be stored in a batch file **fprefix.bat** created in the current **FP Controller** working directory. Each row represents an operation (batch file command). The first column defines the action. Clicking on a cell brings up a menu that includes all programs capable of background operation that are installed on your computer. In addition, several relatively safe standard batch commands are included (**erase**, **copy**, **move**, **rename** and **rem**). All commands operate on a file (**FileIn** column). The batch commands **copy**, **move** and **rename** require a second file name (**FileOut** column). You can type file names in the cells. By default, the files are in the **FP Controller** working directory, but you can include path information if the files are in other directories. Alternatively, you can click in a cell and then pick the **Select file** command to use the standard Windows dialog for choosing files anywhere on the computer. The pick-file dialog displays only files with appropriate suffixes (*e.g.*, *.hin for **HiPhi**).

Click the **OK** button when the sequence is complete to create the batch file. Here is an example, contents of the file **electrode01.bat**:

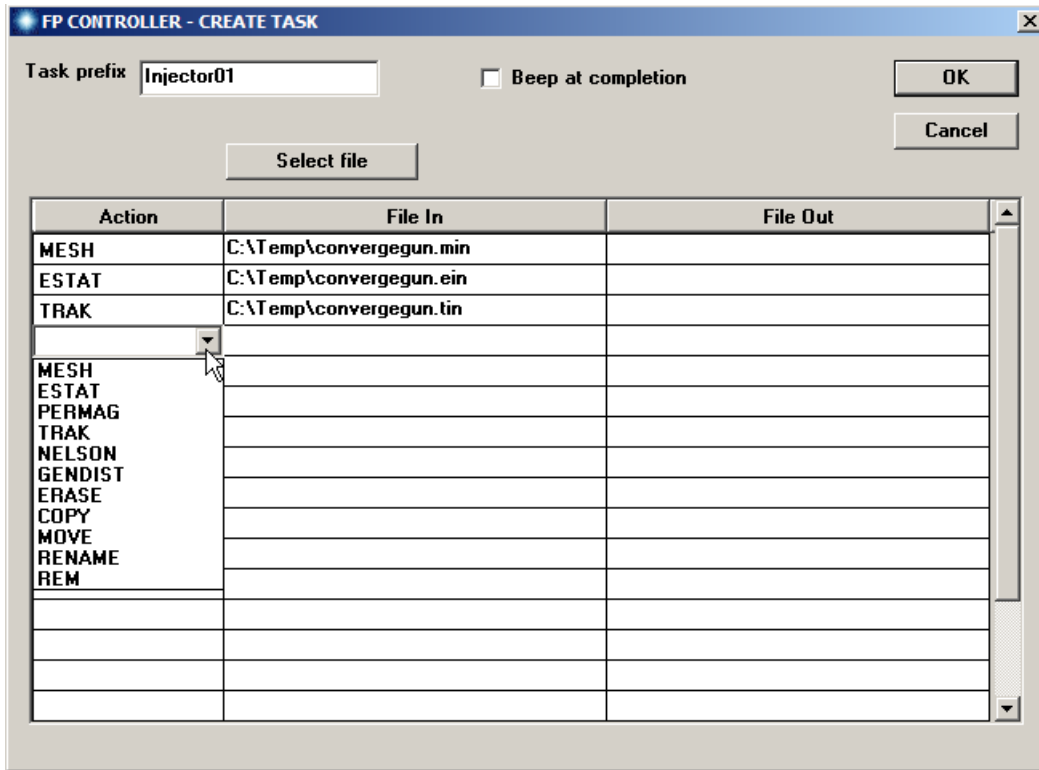


Figure 4: Create task dialog

```
REM FP Controller batch file, Field Precision
START /B /WAIT C:\fieldp\amaze\metamesh.exe C:\Temp\convergegun
START /B /WAIT C:\fieldp\amaze\hiphi.exe C:\Temp\convergegun
START /B /WAIT C:\fieldp\amaze\omnitrak.exe C:\Temp\convergegun
ERASE *.?ls
START /B /WAIT C:\fieldp\notify.exe
IF EXIST Electrode01.ACTIVE ERASE Electrode01.ACTIVE
```

The operations listed perform a complete **OmniTrak** calculation in the background and then erase all listing files. The example has some notable features:

- The operations are performed sequentially because data from one operation may be used in the next. To run calculations in parallel, define and run multiple tasks.
- You can modify the file with an editor if you are familiar with batch commands.
- The batch commands recognize the standard wild card conventions (* for any character grouping, ? for any character).
- The programs adds the command `notify.exe` to the task sequence if the **Beep at completion** box was checked. In this case, the computer makes an audio signal when a task is completed.
- The final command to erase the file `fprefix.active` is added to all batch files. The presence of the file indicates that the task is running.

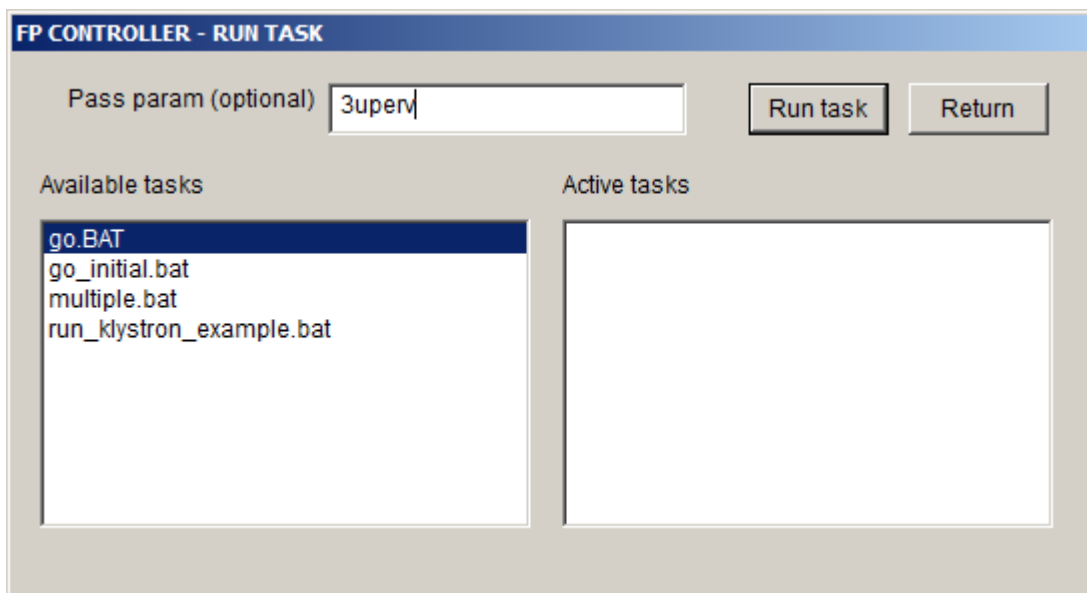


Figure 5: Run task dialog

- The switch `/B` specifies that the process should run in the background without opening a command window. The switch `/WAIT` designates that the process should run to completion before starting the next action. This ensures, for example, that the output file from **MetaMesh** will be available for input to the **HiPhi** calculation.

Click the **Run task** button when you have created tasks or moved predefined task files to the data directory. The dialog (Figure 5) organizes tasks into two groups: ones that are available to run and ones that are currently running (*i.e.*, `fprefix.active` has been detected). To launch a task, choose one from the left-hand list and click **OK**. The program creates a file `fprefix.active` and runs the batch file. The program sequence runs silently in the background. In the meantime, you can prepare other inputs or run other tasks. If there is an error and a batch file does not run to completion, you may need to delete `fprefix.active` manually to restore the list.

Advanced batch file techniques

The example of the previous section performed an **OmniTrak** run for a specific set of files. Rather than create a batch file for each calculation, we can set up a sequence that acts like a macro, applicable to all solutions with the same set of operations. Suppose we use the **Create task** dialog to generate the file `Go.bat`. We enter the same set of operations, but type `%1` in the **File** column. The symbol indicates that the batch file should substitute the first pass parameter after the command. The modified file has this content:

```
REM Go macro: Trak runs with electric field solutions
START /B /WAIT C:\fieldp\amaze\metamesh.exe %1
START /B /WAIT C:\fieldp\amaze\hiphi.exe %1
START /B /WAIT C:\fieldp\amaze\omnitruk.exe %1
ERASE *.?ls
START /B /WAIT C:\fieldp\amaze\notify.exe
IF EXIST %1.ACTIVE ERASE %1.ACTIVE
```

The task file should be in the working directory that contains sets of input files. The file names in each set have the same prefix with the suffixes min, hin and oin. To complete the solution with file prefix 3uperv, we enter the prefix in the **Pass param** box of the **Run task** dialog (Fig. 5), highlight **go.bat** and click **Run task**. An alternate method is to click the **Command line** button in the main menu (Fig. 1) to open a terminal window. Then type

```
go 3uperv
```

and click **Enter**. Batch file macros may also be called from other batch files. Suppose the file **CheckExamples.BAT** has the following content

```
call go 3muperv
call go beambfielddemo
call go beamstatistics
call go BiPolar
...
```

Highlight the file in the **Run task** dialog, then click the **Run task** button to carry out all four solutions. When calling a batch file from a batch file, note that the **call** command must precede the subtask name.

You can create batch files to process large sets of input files automatically. Suppose the goal is to check the effect of small changes in focusing electrode geometry on the output beam distribution of an electron gun. There are twenty sets of input files with names of the form **FocusCheck01.min**, **FocusCheck02.min**, ..., **FocusCheck20.min**. The following batch file performs the complete set of operations

```
FOR /L %%A IN (1,1,20) DO (
  IF %%A LSS 10 CALL GO FocusCheck0%%A
  IF %%A GEQ 10 CALL GO FocusCheck%%A
)
```

The double percent symbols designate a batch file variable. The **/L** command specifies that the loop contains a range of values with the syntax

```
FOR /L %%var_name IN (Lowerlimit, Increment, Upperlimit) DO (some_code)
```

The **if** commands ensure that the file prefix includes a zero for values of the index less than 10. In conclusion, the utility of Windows batch files in automated calculations makes up for their quirky syntax. The Internet contains innumerable references on batch files. Here is a useful tutorial:

<https://www.trytoprogram.com/batch-file/>