

OpenFIRE Middleware Wrapper

This manual includes the information for a user to utilise the middleware to couple FDS with the OpenSEES. It includes a number of modules to perform various tasks such as making devices for the FDS, transferring data from FDS to OpenSEES and so on. All these models are discussed later in this manual.

FSDM middleware can be run both in **Windows**, **MacOS**, and **Linux** machines.

For Windows run **FSDM.exe** file, it will run in the command window. And, for MacOS and Linux run **FSDM** (Unix Executable) by clicking (double click) the FSDM file, the middleware will execute in the Terminal window.

Available Modules in FSDM

FSDM middleware package includes various modules which are listed below and explained later one by one.

FSDDevices : Generates devices for the FDS script.

BNDFBatchFile : When BNDF method is used to measure some quantity in the FDS, this module in the middleware generates a batch file that can be run in Terminal to produce output of the FDS in ascii format.

FDS2OpenSEES : When Device approach is used to get the output from the FDS, this module converts the data in the OpenSEES format.

BNDF2OpenSEES : When BNDF approach is used, this module will utilise the output from the fds2ascii program, and finally it will generate the output in the OpenSEES format to perform heat transfer analysis. Refer FDS User's manual [Ref] to get the data from fds2ascii.

Running FSDM

Once user run the executable file, a prompt will occur and ask the user to enter the module name. User needs to provide the module name for the job he/she wants to perform, for example if user wants to take data from the FDS generated .csv file, the input should be **FDS2OpenSEES**.

Enter module name: Here user needs to provide the module name.

The module name can be typed in both uppercase or lowercase (or in combination). Once user provide the appropriate module name then, for all modules, user would be asked the directory where user's input files are saved or where user wants to save the output of the particular module.

Write the directory location: Provide the directory location (Please check the format for directory location for Windows and Mac or Linux users).

After entering the directory location, the FSDM will execute further. Now, the inputs are relative to the job user wants to perform. In the next chapters all modules are discussed one by one, as and inputs.

FDSDevices

FDS provides the output data in the same order as devices (the quantity to be measured) are written in the FDS script. This module helps to produce the 'devices' in the pre-defined order as of OpenSEES entities [REF]. The DEVICES, including the coordinates and orientations, can be copied or added to the FDS script file.

The output file will be saved at the location user provided in the previous step. Once user provided the location, user would be asked to give the FDS file name.

Name of the FDS Job: Provide the name of the FDS file. User can define any extension such as *.fds* or *.txt*.

Are devices installed to measure quantity? : Now, program will ask, if any device is installed to measure the quantity. User can answer 'y' or 'n' (uppercase or lowercase).

What do you want to measure? : User needs to tell what quantity he/she is willing to measure as an output from the FDS e.g. Adiabatic Surface Temperatures, Heat Fluxes, Heat Transfer Coefficient, etc. Refer FDS User's Manual for the name of the **Quantity** [Ref].

What is the ID of the device? : User needs to provide the ID of the device such as AST for Adiabatic Surface Temperature or HF for Heat Flux (or whatever name user wants to provide).

Now, program will ask about the locations of the devices. This is very critical to follow the sequence, as FDS provides the data in the same sequence as devices are installed.

As this code is written to make a devices for the FDS script file, the devices needs to be in the same **sequence as OpenSEES entities** so that the outputs from the FDS comes in the same sequence to transfer the final inputs for OpenSEES for HT analysis. **Therefore, make sure that sequence to be followed in OpenSEES HT entities.**

In OpenFIRE framework, the sequence of data inputs at the structural elements is maintained and must be followed to get the right inputs for OpenSEES to perform heat transfer analysis. Firstly columns are defined then beams which is followed by slabs.

Therefore, the program will ask if user wants to install any device on the columns to get the data from the FDS. So, first define the entities of the **columns**.

Do you have columns? Y or N

Answer in "y" or "n" if the OpenSEES models include columns.

If "y", first provide the value of X and Y coordinates.

X COORDINATE: Please provide the value of X: Provide the coordinate in X direction, it is recommended to user to start assigning the location of the devices from **the bottom of the first column** (easy to follow the sequence, which is critical for very large structure).

Y COORDINATE: Please provide the value of Y: provide the coordinate in Y direction.

Now, code would ask the initial coordinate in the Z direction and the range of Z until the entities are defined for the particular columns (if all entities are regular size, then height would be the range and initial value would be zero).

Initial Value of Z: provide the initial value of the Z coordinates (generally, starts from the bottom).

Increment of Z: Now, user has to provide the **increment** over Z direction, if the devices are located at equidistance, if not, then repeat as another column.

Maximum Value of Z: Provide the height of the column, if the devices are located at equidistance, if not then the maximum height would be the device location, and process would be repeated as another column.

Orientation: It is the orientation of the devices on the structural elements over which the quantity (AST, HTC, etc) is to be measured, its values would be **plus** or **minus** 1,2,and 3 (-1 for negative X, 1 for positive X, -2 for negative Y, 2 for positive Y , -3 for negative Z , 3 for positive Z). Provide this value according to the location of the surface of the entity such as +3 if values of the quantities are taken at slabs. For more details please refer to FDS User's Manual [\[Ref\]](#)

After finalizing these values, code will ask if there are more columns. If "y", procedure would be repeated, however if "n". Program will ask you if there are beams (Longitudinal or Transverse) and slabs. **Remember sequence of OpenSEES entities here again.**

Sequence is set: First Columns, then Beams (First longitudinal then transverse) and finally Slabs

Once the locations for first quantity is set, code will ask if more devices are installed to measure another quantity such as if user inputs Adiabatic Surface Temperature earlier, now he/she wants to install devices to measure heat fluxes.

If another type of devices are installed to measure quantity?: if user wants to measure another quantity, he/she can answer in 'Y' and 'N'. If 'y', program will continue and repeat the whole process, if 'n', it will terminate the program

This program will create devices locations for the FDS script file according the OpenSEES entities location. The other information for the FDS model such as geometry, HRR and so on, can be added to this file.

Note: Make sure the sequence of the data is same as of the of the OpenSEES entities.

BNDFBatchFile

While using the BNDF approach, FDS provides the output data in binary (.smv file) which can be converted to ASCII format by using a program 'fds2ascii' that comes with FDS package. However, to get the data for whole simulation time, running the fds2ascii is cumbersome. This module can make a **batch file** which can be run in fds2ascii to get the data from the FDS. Currently, to get the data from the '.smv' running the batch file is the most appropriate method.

Write the SMV file name: Write the name of the FDS job name, generally the '.smv' file has the same name as FDS job name. No need to write the format of the file e.g. if the file is 'example.smv' just type 'example'. Because the program fds2ascii can read the file just by name (refer FDS User's Manual [Ref] for more information).

Now, program will ask the inputs for the batch file. To be precise it will ask where user wants to get the data and simulation parameters used in the FDS calculations.

Time of simulation: How long the simulation in FDS has been performed or until what time user wants to import data from the '.smv' file.

Mesh Size: Provide the mesh size in the FDS (in the region where solid entities are defined), it maintains the limit between the coordinates (to avoid the errors related to fractions as size of the meshes in CFD is much larger than the meshes for the structural components in OpenSEES).

Initial Value of time (T Initial): Provide the initial value of time (one can use 0). It generally ask the user from where user wants to exploit data from the CFD simulation.

Time interval (T Interval): Provide the time over which average of the quantities to be taken for the output from the FDS such as 5 sec or 10 sec. For a thermally thick material it can be larger than 5 seconds.

Note: This code is written to make a batch file in the **sequence OpenSEES entities** to be defined to make the final inputs for OpenSEES for HT analysis. **Make sure that sequence to be followed in OpenSEES HT entities.**

First define the entities of the **columns**.

Do you have columns? Y or N

Answer in "y" or "n" if the OpenSEES models include columns.

If "y", first provide the value of X and Y coordinates.

What is the X ?: Provide the coordinate in X direction.

Value of Y: Provide the Coordinate in Y direction

Then code would ask the initial coordinate in the Z direction and the range of Z until the entities are defined for the particular columns (if all entities are regular size, then height would be the range and initial value would be zero).

Initial Value of Z: provide the coordinate value of coordinates (generally from the bottom, if the first entity is at bottom it would be zero).

Range of Z: Provide the height of the column, if all entities are equal in size otherwise run each time for columns.

Increment of Z: Now, user has to provide the **increment** over Z direction, basically it is the size of each entity if all entities are equal in size. If the entities are irregular in size, one can keep running the program as another column.

What is the value of IOR: It is the orientation over which the quantity (AST, HTC, etc) is required, its values would be plus or minus 1,2,and 3 (-1 for negative X, 1 for positive Z, -2 for negative Y, 2 for positive Y , -3 for negative Z , 3 for positive Z). Provide this value according to the location of the surface of the entity such as +3 if values of the quantities are taken at slabs.

Index of the variable: Now provide the index of the variable (it asks how many quantities to be calculated for example if only one quantity is needed just give 1 , this is tricky as for the same quantity such as AST, index value will change if multiples meshes are used). Therefore, **it will ask the index for each entity.**

Note: It is recommend to a user first run the smv file in fds2ascii to make sure the index number for each quantity in particular mesh region.

After finalizing these values, code will ask if there are more columns. If “y”, procedure would be repeated, however if “n”. Program will ask you if there are beams (Longitudinal or Transverse) and slabs. **Remember sequence of OpenSEES entities here again.**

Sequence is set: First Columns, then Beams (First longitudinal then transverse) and finally Slabs.

After finalizing the all process, user can exit the program by **pressing any key.**

This program will create a batch file to exploit data from the FDS2ASCII. The final command file (*BatchFile.command*) is saved in a folder (created by the program) named as **Batchfile.**

#####

Running the script file

While running in Terminal (MacOX or Linux), use:

`chmod +x BatchFile.command`

and in the next line run the Batch file with

`.\BatchFile.command`

Remember to be in the same directory while running in Terminal and (*BatchFile* is the name of the command file).

!!! Don't forget to save the SMV file in the same folder, while running the BatchFile in the fds2ascii program.

#####

FDS2OpenSEES

This code is to be run when the time-temperature .csv file is available (basically when Device approach is used) and data is required in the OpenSEES format. *&DEV*C approach is used in FDS where all data are stored in one .csv file (the sequence of the devices would be based on the devices installed while writing the FDS script). From this program a number of .DAT files for time-varying quantity (AST, HTC, heat fluxes) would be created for each entity.

Before running the middleware: Save the output file of the FDS (*jobname_dev*.csv) in the directory where user wants to work and required the output files from this module of the middleware (directory given in the last step of the code).

After defining the location of the directory, user needs to type the name of the output file from the FDS.

FDS output file name: Define the input file name of the FDS output. The output .csv file would be named as "*jobname_dev*.csv". Add this file in the directory and write its name in the code. Do not forget to add the extension for the .csv file.

Note: Make sure the sequence of the devices in the *jobname_dev*.csv has the same sequence of the OpenSEES entities.

Now, program will ask if there are devices there, if yes, then type "y" else "n". If no, then in the next prompt give zero.

Do you have Devices? y or n: Once the user replied as "y", program will ask about the number of devices of one type, so provide the first type of devices in the input file (*jobname_dev*.csv) for example if a file has 10 AST devices, 10 wall heat flux, and 10 heat transfer coefficient. Then user will enter 10 and he/she will get the AST devices data.

How many Devices? Provide the number of one type of devices (e.g. 10 for above example)

Write the folder name: here, user can provide the name of the folder of each device separately to distinguish each device in separate folder for example AST as folder 1, then HF, and HTC. Folders will be created in the same directory. All files will be stored in the relevant folder of each type of devices with time-varying data for each entity in OpenSEES format.

The files in each folder would be named as the folder name e.g. in AST folder files would be AST1, AST2 and so on, and similarly in HF it would be HF1, HF2 and so on, **these numbering from 1 to last are in the same sequence of OpenSEES entity.**

After giving the folder name, the procedure will be repeated for another quantities, if any. IF there is no more devices give 0 as input for the number of devices.

Make sure that sequence for each device to be followed in OpenSEES HT entities.

BNDF2OpenSEES

This module of the middleware is to utilised to generate files in the format of OpenSEES from the data obtained from *fds2ascii* (a program distributed with FDS) after running the command file (produced by running the **BNDFBatchFile**). Therefore, this code is run after running the script file and obtaining the data from *fds2ascii*. This code removes the unnecessary information such as coordinates location, units from the files obtained from the *fds2ascii*, and finally join all files as time-varying quantity.

Before running the middleware : Once user defined the location in previous step. User is recommended to save the output files first which comes after running the *BNDFBatchfile.command* in *fds2ascii* in the directory user provided in the previous step.

Note: the file named as *test1*, *test2*, and so on are stored in the folder.

Then, user has to define the total simulation in FDS calculation.

Total Simulation Time: Type up to how long the simulation in FDS had been performed.

Time interval (T Interval): Provide the time over which average of the quantities to be taken for the output from the FDS such as 5 sec or 10 sec.

How many locations data needs to be extracted? : Here, user has to define the number of locations data had been extracted from the BNDF file using *fds2ASCII*.

Once the code is finished it will make a number of folders containing a large number of files. To separate the files and managing large files a number of folders (Header/Values/DAT/OpenSEES) are generated. The final files with time-varying data in the appropriate format for OpenSEES to perform HT analysis can be found in a folder named: *OpenSEES*.

Note: Make sure the sequence of the data (locations where data is being extracted) is same as of the of the OpenSEES entities.

A large number of files are obtained after running the script file in *fds2ascii*, the name of the files must be same as mentioned in this program i.e. *test1.csv*, *test2.csv*, and so on.