## Environment Variables to Assist Running

Some environment variables can be used with single or several idf file(s) to keep IDF files pristine while allowing developers or others trying to determine a problem or run a group of files. Many of these have counterparts in Output:Diagnostics objects (some may not be documented). Likewise there may be some “sections” that will produce certain outputs not generally documented.

#### DDOnly: Design Days Only

Setting to “yes” will cause EnergyPlus to set Run Control option (regardless of whether there is a Run Control object in the input file) for Do the Design Day Simulation to “yes” and Do the Weather File Simulation to “no”. (Uses logical variable **DDOnly** in module DataSystemVariables). There is no Output:Diagnostics equivalent.

Set DDOnly=yes

#### FullAnnualRun: Full Annual simulation

Setting to “yes” will cause EnergyPlus to set Run Control option (regardless of whether there is a Run Control object in the input file) for Do the Weather File Simulation to “yes”. Scripts should use a weather file when this environment variable is set. (Uses logical variable **FullAnnualRun** in module DataSystemVariables). There is no Output:Diagnostics equivalent.

Set FullAnnualRun=yes

And appropriate changes to script files

#### NoWeatherFile: Do not use weatherfile even if indicated

Setting to “yes” doesn’t cause EnergyPlus to do anything but can be used in the scripts to not copy a weather file even when indicated. There is no Output:Diagnostics equivalent.

Set NoWeatherFile=yes

And appropriate changes to script files

#### ReverseDD: Reverse Design Days during run

Setting to “yes” causes the first two design days requested in the input file to be reversed during EnergyPlus execution. (Uses logical variable **ReverseDD** in module DataSystemVariables). For proper comparisons to original order, a program such as **ReverseDDInCSV** or **ReverseDDInESO** must be run or hand edited. There is no Output:Diagnostics equivalent.

Set ReverseDD=yes

And appropriate changes to script files

#### MinReportFrequency: Set minimum reporting frequency for outputs

Some developers persist in reporting at the timestep or detailed level even if their runs are full annual runs. This is really burdensome on developers that try to run the full test suite for checking changes. The MinReportFrequency environment variable allows EnergyPlus to report at a higher/less frequent level that still allows for changes to be checked (though differences may require more frequent reporting to track down). EnergyPlus reads this environment variable and sets reporting frequency appropriately. There is no Output:Diagnostics equivalent.

Set MinReportFrequency=daily

The standard frequencies accepted by EnergyPlus must be used: detail, timestep, hourly, daily, monthly, runperiod, environment, annual. In addition, if this environment variable is used, the following will show in the .eio file:

! <Minimum Reporting Frequency (overriding input value)>, Value, Input Value

Minimum Reporting Frequency, !Daily [Value,Min,Hour,Minute,Max,Hour,Minute],DAILY

#### ReportDuringWarmup: Cause reporting during warmup

Setting to “yes” causes reporting (Output:Variable, Output:Meter) to be reporting during the warmup days at the start of each environment. (Uses logical variable **ReportDuringWarmup** in module DataSystemVariables). The Output:Diagnostics, ReportDuringWarmup; is equivalent.

Set ReportDuringWarmup=yes

#### DisplayAllWarnings: turn on all extra warnings

Setting to “yes” causes turning on the warnings: DisplayAllWarnings, DisplayExtraWarnings, DisplayUnusedSchedules, DisplayUnusedObjects. DisplayUnusedObjects also displays unused fluids (that is, fluids that are not directly retrieved by the simulation). The Output:Diagnostics, DisplayAllWarnings; is equivalent.

Set DisplayAllWarnings=yes

#### DisplayExtraWarnings: turn on extra warnings

Setting to “yes” causes turning on DispalyExtraWarnings. The Output:Diagnostics, DisplayExtraWarnings; is equivalent.

Set DisplayExtraWarnings=yes

#### DisplayUnusedObjects: turn on display of unused objects and fluids

Setting to “yes” causes the orphan objects and fluids that are in the input file but not used to be displayed at the end of the simulation. The Output:Diagnostics, DisplayUnusedObjects; is equivalent.

Set DisplayUnusedObjects=yes

#### DisplayUnusedSchedules: turn on display of unused schedules

Setting to “yes” causes the schedules that are in the input file but not used to be displayed at the end of the simulation. The Output:Diagnostics, DisplayUnusedSchedules; is equivalent.

Set DisplayUnusedSchedules=yes

#### DisplayZoneAirHeatBalanceOffBalance: turn on this development output

Setting to “yes” causes the program to calculate and display the Zone Air Heat Balance “out of balance” warnings. The Output:Diagnostics, DisplayZoneAirHeatBalanceOffBalance; is equivalent.

Set DisplayZoneAirHeatBalanceOffBalance=yes

#### IgnoreSolarRadiation: turn off using Solar in simulation

Setting to “yes” causes the program to ignore all solar values during simulation. One might use this when a new set of weather files are produced but you are unsure of their impacts. It also speeds the program as no shading calculations are done. The Output:Diagnostics, IgnoreSolarRadiation; is equivalent.

Set IgnoreSolarRadiation=yes

#### DisplayAdvancedReportVariables: turn on access to advance report variables

Setting to “yes” causes the program to be able to access the “advanced report variable” set in the program. The Output:Diagnostics, DisplayAdvancedReportVariables; is equivalent.

Set DisplayAdvancedReportVariables=yes

#### SortIDD: turn on (or off) the sorting of IDD objects

Setting to “yes” (internal default) causes the program to use sorted lists (a speed up mechanism) in searching for existant objects during IDF processing. There is really no reason to turn it off. There is no Output:Diagnostics equivalent.

Set SortIDD=yes

#### DeveloperFlag: turn on (or off) some different outputs for the developer

Setting to “yes” (internal default is “no”) causes the program to display some different information that could be useful to developers. In particular, this will cause the Warmup Convergence output to show the last day for each zone, each timestep. There is no Output:Diagnostics equivalent.

Set DeveloperFlag=yes

#### IgnoreBeamRadiation: turn on (or off) the beam radiation

Setting to “yes” causes the program to ignore beam radiation in solar calculations. Of limited use, but may be useful for some simulation developments or isolating problems. There is no Output:Diagnostics equivalent.

Set IgnoreBeamRadiation=yes

#### IgnoreDiffuseRadiation: turn on (or off) the diffuse radiation

Setting to “yes” causes the program to ignore diffuse radiation in solar calculations. Of limited use, but may be useful for some simulation developments or isolating problems. There is no Output:Diagnostics equivalent.

Set IgnoreDiffuseRadiation=yes

#### SutherlandHodgman: turn on (or off) the polygon clipping algorithm

Setting to “yes” (internal default) causes the program to use the SutherlandHodgman polygon clipping algorithm. You might use this if you wanted to run a bunch of files with the legacy clipping algorithm (convex Weiler Atherton). There is no Output:Diagnostics equivalent.

Set SutherlandHodgman=yes

#### MinimalShadowing: turn on (or off) the use of Minimal Shadowing

Setting to “yes” causes the program to use Minimal Shadowing (a speed up mechanism) in calculations. This could be useful when large files are being run to search for non-shadowing related problems. There is no Output:Diagnostics equivalent.

Set MinimalShadowing=yes

#### DisplayInputInAudit: turn on (or off) to show the input file in the audit file

Setting to “yes” causes the audit file to include a line-by-line echoing of the input file which may be useful for debugging purposes. For versions 8.2 and earlier, the default behavior was to include the line-by-line echoing of the input file into the audit file but it was changed to not do this by default as a way to speed up the input processing portion of executing EnergyPlus. When not set to “yes”, the audit file contains a reference on using the DisplayInputInAudit environment variable to see the line-by-line echoing of the input file.

#### Caution: Environment Variables

Some combinations will cause fatal errors from EnergyPlus – **DDOnly** and **FullAnnualRun**, for example. **FullAnnualRun** and **NoWeatherFile** won’t cause fatal errors from EnergyPlus but probably should from the script files. We welcome any suggestions for future environment variables.

## Air Loop Simulation Environment Variables

To facilitate comparing the runtime performance of various solution techniques for the air loop simulation a mechanism to track runtime statistics has been implemented in EnergyPlus. To facilitate debugging the air loop simulation a tracing mechanism operating either at the air loop-level or at the controller-level has been implemented in EnergyPlus. To use, one uses the “Set” command and the value as indicated.

#### TRACK\_AIRLOOP: Runtime performance tracker for air loop simulation

Setting to “yes” will cause EnergyPlus to track the performance of the air loop simulation at runtime and dump the results in a file named “statistics.HVACControllers.csv” upon ending the simulation. (Uses logical variable **TrackAirLoopEnvFlag** in module DataSystemVariables).

Set TRACK\_AIRLOOP=yes

#### TRACE\_AIRLOOP: Air loop simulation tracer

Setting to “yes” will cause EnergyPlus to write to a trace file named “controller.<Air Loop Name>.csv” the converged solutions of all controllers defined on each air loop, at each HVAC iteration. (Uses logical variable **TraceAirLoopEnvFlag** in module DataSystemVariables).

Set TRACE\_AIRLOOP=yes

#### TRACE\_HVACCONTROLLER: Individual HVAC controller tracer

Setting to “yes” will cause EnergyPlus to write to a trace file named “controller.<Controller Name>.csv” a detailed description of each controller iteration at each HVAC iteration. (Uses logical variable **TraceControllerEnvFlag** in module DataSystemVariables).

Set TRACE\_HVACCONTROLLER=yes