

Updating ICs to implement Ovation change

Init_IC and BatchIC
(or vendor specific utility)

The problem

- While many simulator sites may have had to extensively use the tools for initializing IC's (Init_IC and BatchIC, in GSE Opensim), Farley had always managed to structure major mods to either run with a single initialization (in icdriver.f) or run from a supplied 100% power IC and build all other IC's by operating from the 100% power IC.
- For minor mods, sometimes even new models using model builders, we could get by with one initialization loaded by icdriver.f, take the kick upon going to run, stabilize the IC and resnap it. Service Water and Component Cooling Water, for example, were done this way.

What changed?

- We finally got a big DCS (Ovation Turbine Control). This was introduced during an outage the old fashioned way by operating to all conditions from a supplied 100% IC. However, by the time the installation was done in the plant, several modifications had been made. At this point, the simulator IC's had already been re-snapped. This will be the case for most outages going forward. We won't be able to wait for the Ovation output file from the plant because we can't wait to start our IC updates.
- Even minor Ovation updates, require re-initializing all IC's.

Solution (the below uses Opensim terminology):

- Note: We're not getting into the specifics of the utility program usage, just the overall process.
- Make a list of ic's by IC number.
- The list includes the ic number, the turbine load, various parameters that will need to be matched when the IC is run later to tweak it in (such as Tavg, reference demand, ramp rate, etc.)
- Sort the list by turbine load.

Solution (cont)

- From this list, pick the load steps for the various IC's we want to update. For Farley, this turned out to be the following:
 - Full power
 - 900.2
 - 878.0
 - 767.7
 - 642.4
 - 586.3
 - 513.0
 - 498.2

Solution (cont)

- 325.5
- 256.6
- 240.0
- 204.0
- 194.4
- 159.4
- 153.6
- 122.7
- 111.5
- 40.9
- Shut down

Solution (cont)

- Now, ramp the simulator down from 100% to shutdown at 5 mw/min (chose your own speed), pausing at each one of the above steps to use Init_IC to write out a variable=value (output file or dump file) file for each variable in the ovation global partition. The same could be done for other globals, while you're at it, if the need exists.
- Also create a full power and shutdown output file for the ovation global.

Solution (cont)

- Construct an input file for use with BatchIC (or equivalent utility) using the IC list and a column editor (input file example in later slide).
- Run BatchIC with the input file, updating all your IC's with the appropriate new Ovation global values.

Finishing up

- The full power IC's will have very minor kicks since not all full power IC's are at exactly the same load, valve position, etc. A short stabilization period is necessary for each one.
- Shutdown IC's can be reset and run for a few seconds and resnapped to allow turbine speed to propagate through ovation.

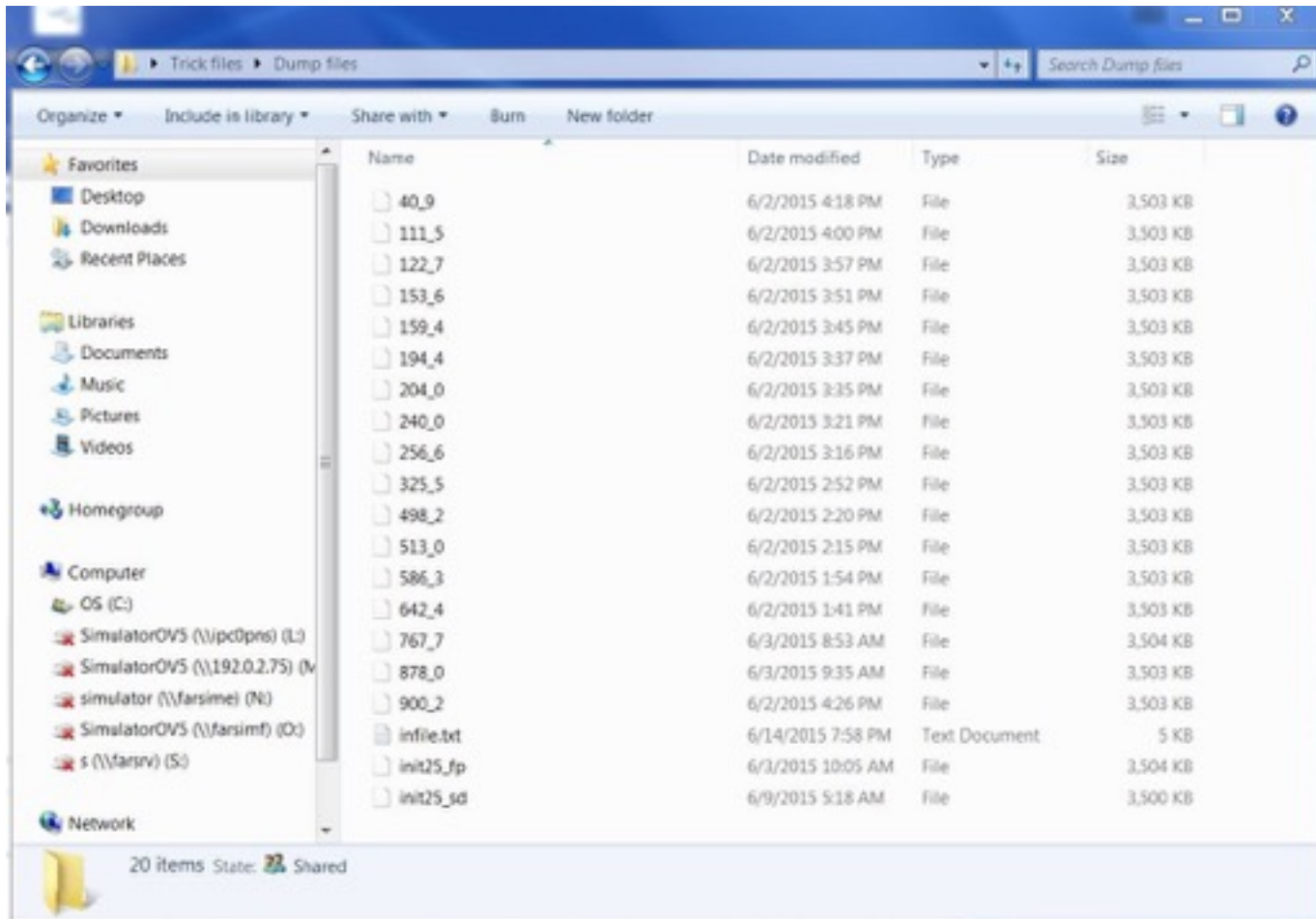
Finishing Up (intermediate power IC's)

- Reset into each IC.
- Freeze Xe/Sm
- Go to run and immediately place the turbine in hold.
- Adjust reference demand and ramp to restore turbine load, T_{avg} , etc., to the original IC's values (recorded in the list at the beginning), finishing with ramp rate at the original IC's value and run to let the plant stabilize.

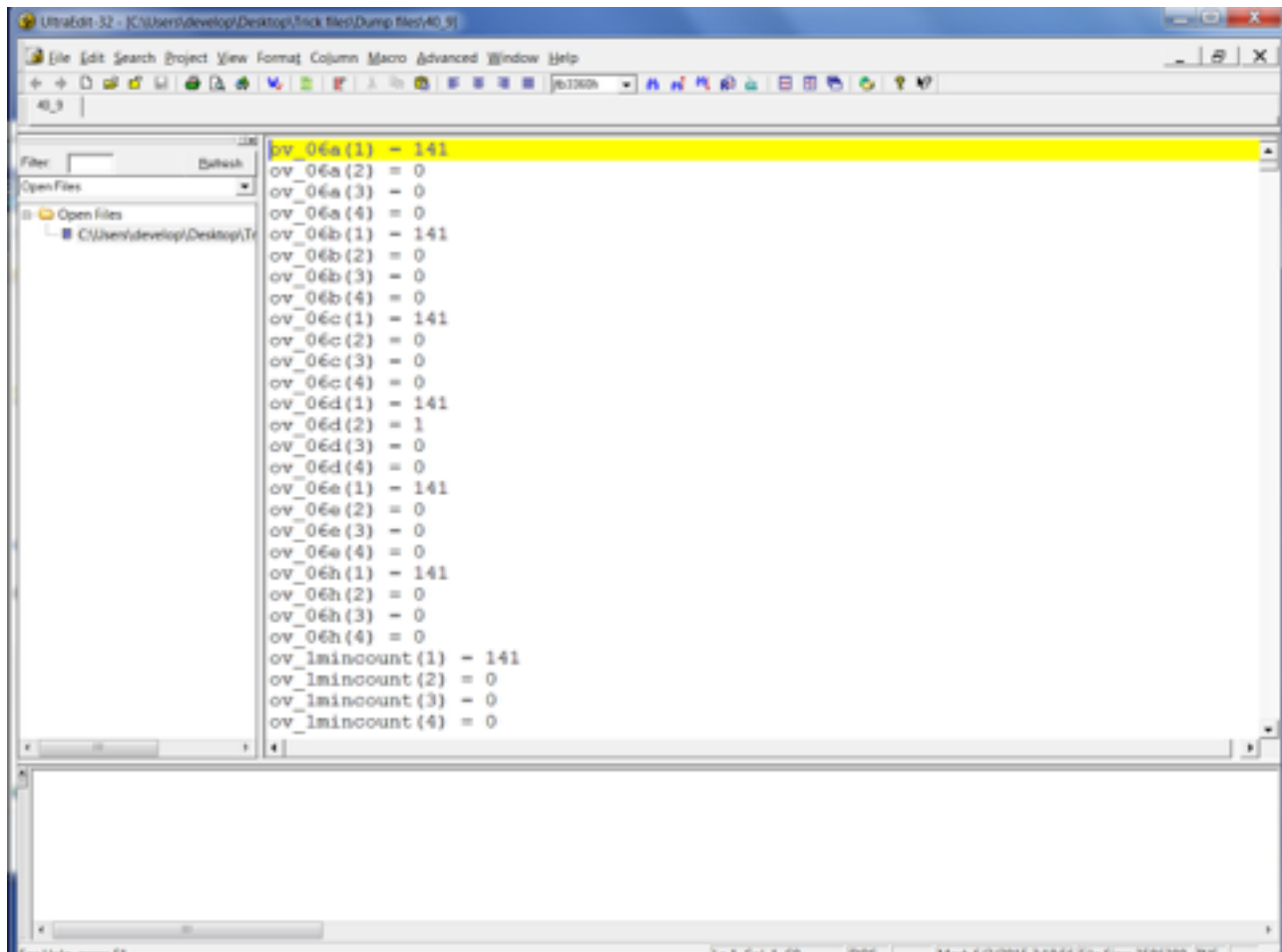
Finishing Up (intermediate power IC's)

- Unfreeze Xe/Sm
- Re-snap the IC.
- We got the ILT and LOCT programs to supply us a list of IC's for the upcoming weeks and concentrated on getting those stabilized and did the remainder of the IC's as time permits.

What the output (dump) files look like



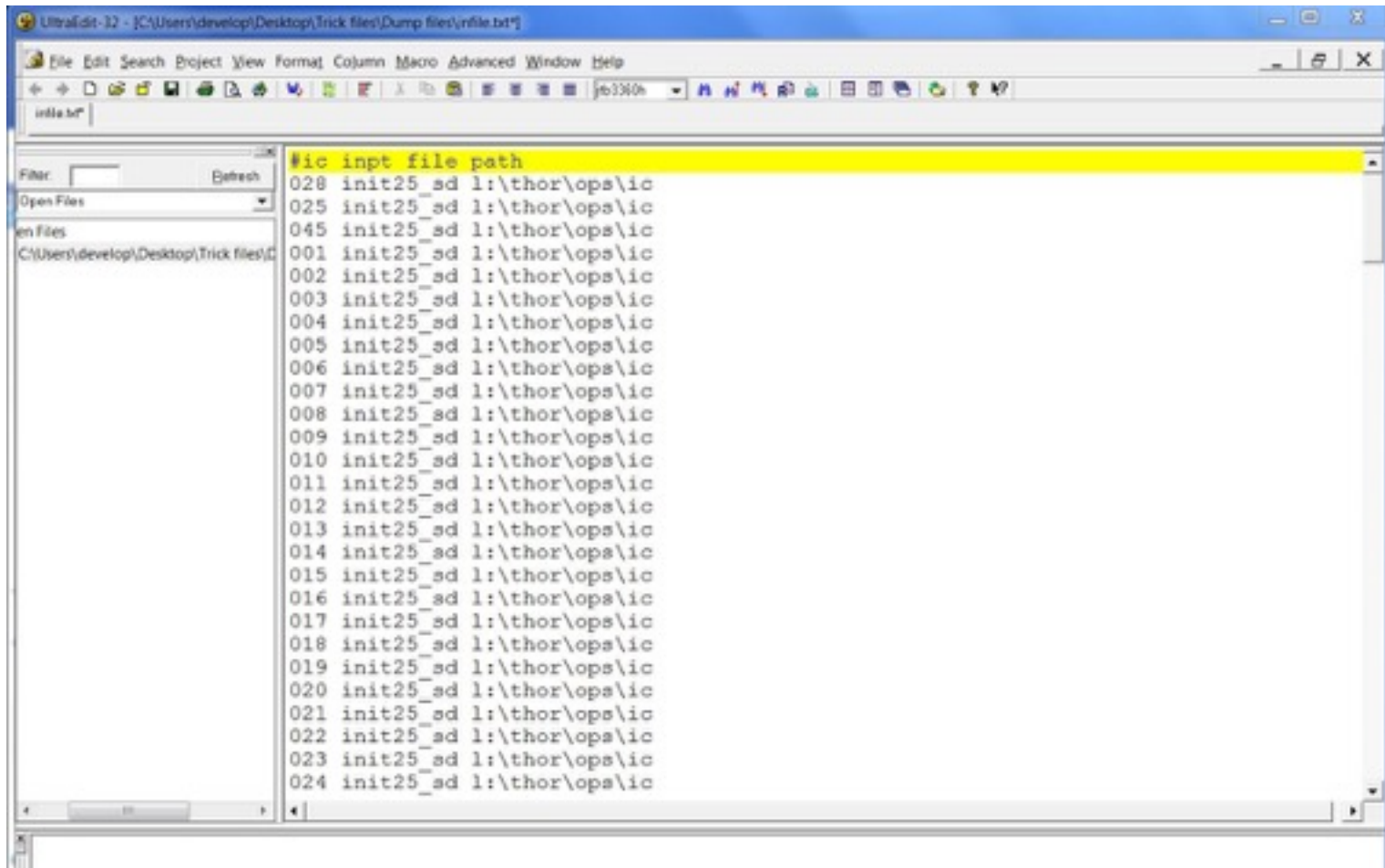
A variable=value file



The screenshot shows the UltraEdit 32 interface with a list of variables and their values. The first line, 'ov_06a(1) = 141', is highlighted in yellow. The list includes variables for memory addresses 06a, 06b, 06c, 06d, and 06e, each with four instances, and a variable named 'ov_lmincount' with four instances. The values are either 141 or 0, with some instances of 06d having a value of 1.

```
ov_06a(1) = 141
ov_06a(2) = 0
ov_06a(3) = 0
ov_06a(4) = 0
ov_06b(1) = 141
ov_06b(2) = 0
ov_06b(3) = 0
ov_06b(4) = 0
ov_06c(1) = 141
ov_06c(2) = 0
ov_06c(3) = 0
ov_06c(4) = 0
ov_06d(1) = 141
ov_06d(2) = 1
ov_06d(3) = 0
ov_06d(4) = 0
ov_06e(1) = 141
ov_06e(2) = 0
ov_06e(3) = 0
ov_06e(4) = 0
ov_06h(1) = 141
ov_06h(2) = 0
ov_06h(3) = 0
ov_06h(4) = 0
ov_lmincount(1) = 141
ov_lmincount(2) = 0
ov_lmincount(3) = 0
ov_lmincount(4) = 0
```

The input file for use with BatchIC



The screenshot shows the UltraEdit-32 application window. The title bar reads "UltraEdit-32 - [C:\Users\develop\Desktop\Trick files\Dump files\infile.txt]". The menu bar includes "File", "Edit", "Search", "Project", "View", "Format", "Column", "Macro", "Advanced", "Window", and "Help". The toolbar contains various icons for file operations. The main text area displays a list of file paths, with the first line highlighted in yellow. The list starts with a header line "#ic inpt file path" and contains 21 data lines, each with a line number and a file path.

```
#ic inpt file path
028 init25_sd 1:\thor\ops\ic
025 init25_sd 1:\thor\ops\ic
045 init25_sd 1:\thor\ops\ic
001 init25_sd 1:\thor\ops\ic
002 init25_sd 1:\thor\ops\ic
003 init25_sd 1:\thor\ops\ic
004 init25_sd 1:\thor\ops\ic
005 init25_sd 1:\thor\ops\ic
006 init25_sd 1:\thor\ops\ic
007 init25_sd 1:\thor\ops\ic
008 init25_sd 1:\thor\ops\ic
009 init25_sd 1:\thor\ops\ic
010 init25_sd 1:\thor\ops\ic
011 init25_sd 1:\thor\ops\ic
012 init25_sd 1:\thor\ops\ic
013 init25_sd 1:\thor\ops\ic
014 init25_sd 1:\thor\ops\ic
015 init25_sd 1:\thor\ops\ic
016 init25_sd 1:\thor\ops\ic
017 init25_sd 1:\thor\ops\ic
018 init25_sd 1:\thor\ops\ic
019 init25_sd 1:\thor\ops\ic
020 init25_sd 1:\thor\ops\ic
021 init25_sd 1:\thor\ops\ic
022 init25_sd 1:\thor\ops\ic
023 init25_sd 1:\thor\ops\ic
024 init25_sd 1:\thor\ops\ic
```