***A\_batchfile.gms***

\* MAIN FILE

\* Platform specific adjustments

\* Setup the file separator to use for relative pathnames

$eolcom //

$setglobal gmsgrid %gams.curdir%gmsgrid.cmd

$ifthen not %system.filesys% == UNIX

$setglobal rm\_dir\_cmd "RMDIR"

$setglobal fs "\"

$setglobal move "move"

$setglobal exe ".exe"

$elseif %system.filesys% == UNIX

$setglobal rm\_dir\_cmd "rm -rf"

$setglobal fs "/"

$setglobal move "mv"

$setglobal exe ""

$endif

\*----------------------------------------------------------------------

\* Process command-line arguments

\*---------------------------------------------------------------------

\* --user=

\* user name to use in accessing database. required to update parameters based

\* on config.

$if not set user $set user 'none\_provided'

\* --focus\_db=rpm\_co or --focus\_db=rpm\_az

$if not set focus\_db $set focus\_db 'rpm\_co'

\* --config=default or --config=test

$if not set config $set config 'default'

\* must be lower case to match .op\* files

$if not set solver $setglobal solver 'cplex'

\* --process\_inputs\_level=0, 1, 2, 3, or 4

\* level 0 - loads what has already been read from the database

\* level 1 - reads non-hourly data from the database; loads everything else from local files

\* level 2 - reads level 1 and subsetted hourly data from database; does not recalculate subsets

\* level 3 - loads all data from local files; recalculates subsets

\* level 4 - reads level 1 and 2 data from database, and recalculates subsets

$if not set process\_inputs\_level $set process\_inputs\_level 4

\* --run\_name='2014-03-24\_test'

$if not set run\_name $set run\_name '%config%'

\* --run\_desc="testing%my%new%feature"

\* GAMS cannot tolerate spaces in the input. The configurtion parser assumes

\* '%' should be replaced with ' '.

$if not set run\_desc $set run\_desc ""

\* SET DEFAULT RPM Version. Correct database will be queried from dav-gis

$if not set rpm\_ver $set rpm\_ver '2016\_06'

\* completely optional argument

\* --gdxsource=runs/rpm\_co/test\_me/inout/gdx/

\* will copy \*.gdx files from specified folder to %gdxfolder%

\* completely optional argument

\* --infeas\_year=2025

\* Will run .op4, which contains an option to diagnose infeasibilities.

\* To see the result, look in the lst file that follows the infeas\_year,

\* search for 'EQUATION COST.Infeas', and then scroll down to see what value is/

\* values are non-zero. This is how much the RHS of that equation/those equations

\* need to be changed to obtain a feasible problem.

\*----------------------------------------------------------------------

\* Define shortcuts to relevant directories

\*----------------------------------------------------------------------

$setglobal projectfolder '%gams.curdir%'

$setglobal runfolder '%projectfolder%runs%fs%%focus\_db%%fs%%run\_name%%fs%'

$setglobal outputfolder '%runfolder%inout%fs%'

$setglobal gdxfolder '%runfolder%inout%fs%gdx%fs%'

$setglobal savedir '%runfolder%t%fs%'

\*----------------------------------------------------------------------

\* Create new scenario folder

\*----------------------------------------------------------------------

\* create new folders for the scenario

$call 'mkdir runs%fs%%focus\_db%'

$call 'mkdir %runfolder%'

$call 'mkdir %runfolder%lstfiles'

$call 'mkdir %runfolder%inout'

$call 'mkdir %runfolder%inout%fs%gdx'

$call 'mkdir %savedir%'

\* copy the code to the runfolder

$call 'cp \*.gms %runfolder%.'

$call 'cp %solver%.op\* %runfolder%.'

\* copy gdx files from specified location:

\* ex. add --gdxsource=runs/rpm\_co/test\_me/inout/gdx/ to command line

$if set gdxsource $call 'cp %gdxsource%%fs%\*.gdx %gdxfolder%'

$if set gdxsource $call 'cp %gdxsource%%fs%..%fs%scaled\_profiles.gdx %outputfolder%'

\* define empty value for configure\_run.py

$if not set gdxsource $set gdxsource ""

$if not set infeas\_year $set infeas\_year 0

\*----------------------------------------------------------------------

\* Set environment parameters

\*----------------------------------------------------------------------

$set env ide=%gams.ide% lo=4 lf=%runfolder%run.log al=1 cdir=%runfolder%

\*----------------------------------------------------------------------

\* Set start year and end year here; starting

\* after 2010 will restart from a save-point

\*----------------------------------------------------------------------

$if not set start $setlocal start '2010'

$if not set end $setlocal end '2035'

$if not set yincrement $setglobal yincrement '5'

$eval current %start%

$eval restart 3+((%current%-2010)/%yincrement%)

$eval save %restart%+1

$show

\*----------------------------------------------------------------------

\* Save run parameters to text file so they can be passed on to B\_\*.gms

\* and turned into global variables, etc., as needed

\*----------------------------------------------------------------------

$onechos > %runfolder%setupcommands.txt

Eolonly 1

--solver=%solver%

o=%runfolder%lstfiles%fs%readinputs.lst

s=%savedir%a1

ide=%gams.ide%

lo=4

sysout=1

lf=%runfolder%run.log

--config=%config%

--user=%user%

--focus\_db=%focus\_db%

--rpm\_ver=%rpm\_ver%

--run\_name=%run\_name%

--run\_desc=%run\_desc%

--projectfolder=%projectfolder%

--runfolder=%runfolder%

--outputfolder=%outputfolder%

--gdxfolder=%gdxfolder%

--gdxsource=%gdxsource%

--process\_inputs\_level=%process\_inputs\_level%

--start=%start%

--end=%end%

--infeas\_year=%infeas\_year%

--yincrement=%yincrement%

Eolonly 0

$offecho

\*----------------------------------------------------------------------

\* If starting from the model start year (i.e., 2010)

\* Read in model inputs (B\_readinputs)

\* Do hourly data calculations (C\_hourlydata)

\* Read in model (D\_model)

\*----------------------------------------------------------------------

$ifthene %current%==2010

\* call readinputs.gms to load data

$call 'gams %runfolder%B\_readinputs\_update.gms pf=%runfolder%setupcommands.txt cdir=%runfolder%';

$if errorlevel 1 $abort 'error in readinput'

\* call hourlydata.gms

$call 'gams %runfolder%C\_hourlydata\_update.gms o=%runfolder%lstfiles%fs%hourlydata.lst %env% r=%savedir%a1 s=%savedir%a2';

$if errorlevel 1 $abort 'error in hourlydata'

$call 'cp -f %outputfolder%scaled\_profiles.gdx %gdxfolder%'

\* call model.gms to load model

$call 'gams %runfolder%D\_model\_update.gms o=%runfolder%lstfiles%fs%model.lst %env% r=%savedir%a2 s=%savedir%a3';

$if errorlevel 1 $abort 'error in model'

$endif

\*$exit

\*----------------------------------------------------------------------

\* Run optimization routine from 'start' to 'end' (E\_doayear)

\*----------------------------------------------------------------------

$label top

$onechos > %runfolder%doayearcommands.txt

Eolonly 1

mip=%solver%

optdir=%runfolder%

o=%runfolder%lstfiles%fs%doayear%current%.lst

r=%savedir%a%restart%

s=%savedir%a%save%

u1=%current%

u2=%current%

u3=%yincrement%

Eolonly 0

$offecho

$call 'gams %runfolder%E\_doayear.gms %env% pf=%runfolder%doayearcommands.txt';

$call 'cp -f Regional\_Model\_%current%\_%run\_name%\_p.gdx %runfolder%'

$call 'rm -f Regional\_Model\_%current%\_%run\_name%\_p.gdx'

$if errorlevel 1 $abort 'error in %current%'

$eval current %current%+ %yincrement%

$eval restart %restart%+1

$eval save %save%+1

$show

$ife %current%<=%end% $goto top

$call 'gams %runfolder%F\_outputs.gms o=%runfolder%lstfiles%fs%output.lst %env% r=%savedir%a%restart% s=%savedir%a%save%';

\*$exit

\*----------------------------------------------------------------------

\* Run a SQL query defined in "cmd2.txt" which

\* copies .csv file data to Postgres

\*----------------------------------------------------------------------

\*$call 'gams G\_datatransfer.gms o=./lstfiles/datatransfer.lst %env% r=%savedir%a1 ';

\*----------------------------------------------------------------------

\* structure of files:

\*----------------------------------------------------------------------

\* using external calls for files that

\* A\_batchfile.gms

\*

\* B\_readinputs\_update.gms - read in raw inputs from dav-gis

\* B1\_switches\_update.gms - defines input processing switches and loads model parameters

\* B2\_sets\_update.gms - defines model sets

\*

\* C\_hourlydata.gms - defines temporal extent of model and normalizes 8760 data

\*

\* D\_model.gms - defines the capacity expansion model variables and equations

\* D1\_parameters.gms - defines all model parameters

\* D2\_definemodel.gms - selects equations to be used in the optimization subject to model parameter switches

\*

\* E\_doayear.gms - drives the annual solution process

\* E1\_process\_prev\_solve.gms - do post-processing calculations for the previous solve year

\* E2\_precalc\_update.gms - get parameters ready for the next model year optimization

\* E3\_variability\_update.gms - do variability calculations, that is capacity value, curtailment, etc. based on 8760 data

\*

\* F\_outputs.gms - save results to gdx