Change to OpenDSS for the DMS Controls Sandbox End of Time Step Cleanup

Adam Birchfield, 5/22/2015, Electric Power Research Institute (EPRI)

Adam's Request for Change to Roger, 5/22/2015

For the Python-COM time series simulations on OpenDSS with custom controls, I think there is a need for some more options in controlling the EndOfTimeStepCleanup behavior. Here is what I would propose:

1. Expose the EndOfTimeStepCleanup to a text command and COM function so that it can be called independently.
2. Add a setting to turn the EndOfTimeStepCleanup actions on or off for time series solution modes. So that if you solve with endcleanup=off, the EndOfTimeStepCleanup actions don't run. This check could be implemented inside the EndOfTimeStepCleanup function itself.
3. In time series solution modes, allow the option not to sample or save energy meters and monitors automatically. This could be done by either adding these functions to the EndOfTimeStepCleanup or providing a check inside each solution mode to see if endcleanup=off.

These changes would allow me from the COM interface to run any mode the user chooses with number=1 and do all the included setup functions for the particular mode without having the storage and inverter classes update their states or the monitors taking a sample until I am ready to do it manually. I suggest that this solution would be better than adding a new solution mode because if the user has set up his model for Daily, Yearly, Duty, or Dynamics the COM module can just run the file in the specified mode, except with endcleanup=off and number=1, and then perform the EndOfTimeStepCleanup and sample monitors/meters after the custom controls have operated.

Changes by Davis and Roger by 6/26/2015:

New method of COM interface Solution class:

1) `Cleanup()` – this method updates storage, inverter controls, etc. for the end of a time step.
2) `FinishTimeStep()` – this method runs `Cleanup()` and then increments time

Simple Python Example

The Python code on the next page demonstrates one way to use the end-of-time-step clean up in the implementation of a control model.

In this example, the individual steps of the solve process are carried out individually so that the control logic can be inserted into the middle of the time step loop. The steps are:

- InitSnap
- **`ControlActionsDone`**
- **`SolveNoControl`**
- **`SampleControlDevices`**
- **`DoControlActions`**
- **`FinishTimeStep`**

```
# Example Python Custom Control Loop Template

import win32com.client

dssObj = win32com.client.Dispatch("OpenDSSEngine.DSS")
dssText = dssObj.Text
dssCircuit = dssObj.ActiveCircuit
dssSolution = dssCircuit.Solution

# Load the given circuit master file into OpenDSS
dssText.Command = r"compile 'master.dss'"

if dssSolution.Mode == 0:
    dssSolution.Mode = 0
originalSteps = dssSolution.Number
dssSolution.Number = 1

# For each time step ----
for stepNumber in range(originalSteps):
    # get current time
    simulationTime = dssSolution.dblHour
    h = dssSolution.Hour
    s = dssSolution.Seconds
    # let user make changes before custom controls are run
    customControlsOptimizationFunction()
    # solve with controls
    iteration = 0
    dssSolution.InitSnap()
    while not dssSolution.ControlActionsDone:
        dssSolution.SolveNoControl()
        customControlLoopFunction()
        dssSolution.SampleControlDevices()  # OpenDSS controls
        dssSolution.DoControlActions()
        # handle user-inserted control queue functions
        while self.dssCircuit.CtrlQueue.PopAction != 0:
            deviceHandle = self.dssCircuit.CtrlQueue.DeviceHandle
            actionCode = self.dssCircuit.CtrlQueue.ActionCode
            customQueueFunction = actionMap[deviceHandle]
            customQueueFunction(actionCode)
        self.iteration += 1
        if self.iteration >= self.dssSolution.MaxControlIterations:
            print "max control iterations exceeded!"
            self.iteration = "MAX EXCEEDED"
            break
    return outputList
    self.dssSolution.FinishTimeStep()

self.dssCircuit.Monitors.SaveAll()
```