



Scripting Topology

Comparison

Daniel Kuraspediani

BPA Technical Operations {TOT}

Work: (360)-418-2148

dkkuraspediani@bpa.gov





Background

- BPA student program since 2008
- *BSEE* at **WSU**
- Pursuing *P.E.*
- Hired in 2011 with *Technical Operations*
- Study areas:

- Raver-Paul
- Paul-Allston
- South of Allston
- West of Cascades South
- West of Cascades North
- Northern Intertie
- South of Custer
- North of Echo Lake



Purpose

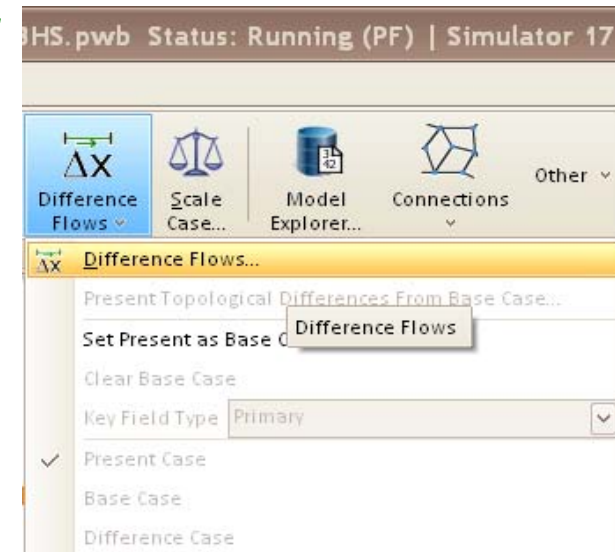
- Compare the topology of an outage case and implement the same topology to other cases.
- Improve study efficiency & reduce human error factor through automation

Overview

- Define filters for comparison
 - Branch differences
 - Added / removed elements
 - Shunt block differences
- Compare outage case to reference case
- Set flags for differences using filters
- Write out AUX files to implement differences to new case(s)

Script {Part 1}

- SCRIPT
- {
- *// Delete old topology difference aux files. "SaveData();" command appends by default so the files need to be deleted if they exist beforehand.*
- `DeleteFile("RemovedElements.aux");`
- `DeleteFile("AddedElements.aux");`
- `DeleteFile("BranchChanges.aux");`
- `Deletefile("ShuntBlockChange.aux");`
- *// Open the original template case with no outages*
- `OpenCase("S13v1SNL_NOEL.pwb");`
- *//Set case as the difference flows reference base case*
- `DiffFlowSetAsBase;`
- *// Open the case with outages. This is the difference flows Present case.*
- `OpenCase("S13v1NSH_SOC.pwb");`
- }



Script {Part 2.1}

- *//Define Filters to check to topology differences, and/or added, and/or removed elements*
- DATA (FILTER, [ObjectType,FilterName,FilterLogic,FilterPre,Enabled],,YES)
- {
- *// Define the filters that will return the elements that have differences*
- "Branch" "Diff Branch Status" "AND" "NO " "YES"
- <SUBDATA Condition>
- LineStatus contains "|"
 </SUBDATA> *//the pipe character is used to distinguish any difference between the present and base case*
- "Shunt" "Diff Shunt Status" "AND" "NO " "YES"
- <SUBDATA Condition>
- SSStatus contains "|"
- </SUBDATA>

Script {Part 2.2}

- *// The following filters will identify elements added to the case*
-
- "Bus" "Buses Added" "AND" "NO " "YES"
- <SUBDATA Condition>
- ContainedInDiffFlowsBC = "NO"
- </SUBDATA>

- "Shunt" "Switched Shunts Added" "AND" "NO " "YES"
- <SUBDATA Condition>
- ContainedInDiffFlowsBC = "NO"
- </SUBDATA>

- "Branch" "Branches Added" "AND" "NO " "YES"
- <SUBDATA Condition>
- ContainedInDiffFlowsBC = "NO"
- </SUBDATA>

Script {Part 2.3}

- // Define comparison filters for shunt blocks
- "Shunt" "Switched Shunt Block Filter" "OR" "NO " "YES"

- <SUBDATA Condition>
- SSBlockMVarPerStep > 0.00010 1
- SSBlockMVarPerStep:1 > 0.00010 1
- SSBlockMVarPerStep:2 > 0.00010 1
- SSBlockMVarPerStep:3 > 0.00010 1
- SSBlockMVarPerStep:4 > 0.00010 1
- SSBlockMVarPerStep:5 > 0.00010 1
- SSBlockMVarPerStep:6 > 0.00010 1
- SSBlockMVarPerStep:7 > 0.00010 1
- SSBlockMVarPerStep:8 > 0.00010 1
- SSBlockMVarPerStep:9 > 0.00010 1
- SSBlockNumSteps > 0 1
- SSBlockNumSteps:1 > 0 1
- SSBlockNumSteps:2 > 0 1
- SSBlockNumSteps:3 > 0 1
- SSBlockNumSteps:4 > 0 1
- SSBlockNumSteps:5 > 0 1
- SSBlockNumSteps:6 > 0 1
- SSBlockNumSteps:7 > 0 1
- SSBlockNumSteps:8 > 0 1
- SSBlockNumSteps:9 > 0 1
- </SUBDATA>
- }

-Switched Shunts Mvar Blocks

Number of Steps	1	1	2	1			
Mvars per Step	-148.8	-297.5	311.5	329.2			

Check the shunt block MVAR value

Check the shunt block number

-Switched Shunts Mvar Blocks

Number of Steps	1	1	2	1			
Mvars per Step	-148.8	-297.5	311.5	329.2			

Script {Part 3}

- SCRIPT BranchDiff
- {
- DiffFlowMode(DIFFERENCE); *// Toggle to DIFFERENCE mode to identify case differences*
- UnSelectAll(Branch); *// Initialize all branches custom field to "Selected" = NO. "Selected" custom field can be set to "NO" or "YES"*
- *// "SetData" command will apply the "Diff Branch Status" filter created to identify topology changes and set custom // field "Selected" to "YES"*
- SetData(BRANCH,[SELECTED],[YES],"Diff Branch Status");
- SetData(SHUNT, [SELECTED], [YES],"Diff Shunt Status");
- DiffFlowMode(PRESENT); *//Switch to PRESENT mode to save differences*
- SaveData("BranchChanges.aux",AUX,BRANCH,[BusNum,BusNum:1,LineCircuit,LineStatus],[, SELECTED,[]]);
- SaveData("BranchChanges.aux",AUX,SHUNT,[BusNum,ShuntID,SSCMode,SSNMVR,SSRegNum,SSVHigh, SSVLow,SSStatus],[,SELECTED,[]]);
- }

Script {Part 4}

- SCRIPT AddedElements
- {
- *// Toggle in and out of DIFFERENCE Mode for Simulator to properly select differences.*
- DiffFlowMode(Base);
- DiffFlowMode(Present);
- *// Write commands to the file being saved to enter edit mode before loading in data*
- WriteTextToFile("AddedElements.aux","SCRIPT");
- WriteTextToFile("AddedElements.aux","{");
- WriteTextToFile("AddedElements.aux","EnterMode(EDIT);");
- WriteTextToFile("AddedElements.aux","}");
- *// Save all added element differences*
- SaveData("AddedElements.aux",AUX,BUS,[BusNum,BusName,BusNomVolt,AreaNum,ZoneNum],[],
"BusesAdded",[]);
- SaveData("AddedElements.aux",AUX,SHUNT,[BusNum,ShuntID,SSCMode,SSNMVR,SSRegNum,
SSVHigh,SSVLow,SSStatus],[],"Switched Shunts Added",[]);
- SaveData("AddedElements.aux",AUX,BRANCH,[BusNum,BusNum:1,LineCircuit,LineStatus,
LineC,LineR,LineX,LineAMVA,LineAMVA:1, LineAMVA:2],[],"Branches Added",[]);
- }

Script {Part 5}

- SCRIPT RemovedElements
- {
- DiffFlowMode(Base);
- DiffFlowMode(Present);
- *// Write commands to the file being saved*
- WriteTextToFile("RemovedElements.aux","SCRIPT");
- WriteTextToFile("RemovedElements.aux","{");
- WriteTextToFile("RemovedElements.aux","EnterMode(EDIT);");
- WriteTextToFile("RemovedElements.aux","}");
- *// Save all removed element differences*
- SaveData("RemovedElements.aux",AUX,REMOVEDBUS,[BusNum],[,],[,]);
- SaveData("RemovedElements.aux",AUX,REMOVEDBRANCH,[BusNum,BusNum:1,LineCircuit],[,],[,]);
- SaveData("RemovedElements.aux",AUX,REMOVEDSHUNT,[BusNum,ShuntID,SSCMode,SSStatus],[,],[,]);
- }

Script {Part 6}

- Script
- {
- *// Toggle in and out of Difference Mode for Simulator to properly select differences.*
- DiffFlowMode(DIFFERENCE);
- UnSelectAll(SHUNT);
- SetData(SHUNT, [SELECTED], [YES], "Switched Shunt Block Filter");
- DiffFlowMode(PRESENT); *//Switch to "Present" mode to save differences*
- SaveData("ShuntBlockChange.aux",AUX,SHUNT,[BusNum,ShuntID,SSCMode,SSNMVR,SSRegNum,SSVHigh,SSVLow,SSStatusSSBlockMVarPerStep,SSBlockMVarPerStep:1,SSBlockMVarPerStep:2,SSBlockMVarPerStep:3,SSBlockMVarPerStep:4,SSBlockMVarPerStep:5,SSBlockMVarPerStep:6,SSBlockMVarPerStep:7,SSBlockMVarPerStep:8,SSBlockMVarPerStep:9,SSBlockNumSteps,SSBlockNumSteps:1,SSBlockNumSteps:2,SSBlockNumSteps:3,SSBlockNumSteps:4,SSBlockNumSteps:5,SSBlockNumSteps:6,SSBlockNumSteps:7,SSBlockNumSteps:8,SSBlockNumSteps:9],[,],SELECTED,[]);
- }

Implementation Process

1. Open `S13v1SNL_NOEL.pwb`
2. Take outages / change topology
3. Save case
4. Load `SCRIPT`
5. OUTPUT:
 - `BranchChanges.aux`
 - `AddedElements.aux`
 - `RemovedElements.aux`
 - `ShuntBlockChange.aux`
6. Load output AUX files in *EDIT MODE*

EXAMPLE: SOC/NOEL study

PW Scripting Commands

- **SetData(objecttype,[fieldlist],[valuelist],filter);**
 - **Objecttype:** The object type being set
 - **[fieldlist]:** A list of fields to save
 - **[valuelist]:** A list of values to set the respective fields to
 - **Filter:** There are four options for selecting a filter to use
 - **SetData(...);** no filter is specified.
 - **SetData(...,"filtername");** use the defined filter "filtername" to set data if conditions are met
 - **SetData(...,SELECTED);** set data for all objects whose *selected* field is set to YES
 - **SetData(...,ALL);** set data for all objects of the specified object type

PW Scripting Commands

- **SaveData("filename", filetype, objecttype, [fieldlist], [subdatalist], filter, [SortFieldList]);**
 - **"Filename"**: The file to save the data to.
 - **Filetype**: *AUX (or AUXCSV)* – save as a space-delimited (or comma-delimited) aux file.
CSV – save as a normal CSV file with the first few lines containing the objectname and field names.
CSVColHeader – save as a CSV file without the aux file syntax; instead the first row shows the column headers you would see in a case information display.
 - **Objecttype**: the object type being saved.
 - **Filter**: There are four options for selecting a filter to use
 - **SetData(...)**; no filter is specified.
 - **SetData(..., "filtername")**; use the defined filter "filtername" to set data if conditions are met
 - **SetData(..., SELECTED)**; set data for all objects whose *selected* field is set to YES
 - **SetData(..., ALL)**; set data for all objects of the specified object type
 - **[SortFieldList]**: Allows the specification of a sort order in which the data will be saved.
 The format is: [variablename1:+:0,variablename2:-:1]
 - *Variablename*: the name of the field to sort by.
 - + or - : the second parameter indicates sort ascending for + and sort descending for -.
 - 0 or 1: the third parameter indicates: if 0, case insensitive and do not use absolute value; if 1, case sensitive or use absolute value.