

# Modeling of Remedial Action Schemes and Relays in Power Flow Simulations

---

PowerWorld Client Conference  
February 23, 2016



Mark Laufenberg, Ph.D.

President

lauf@powerworld.com

217-384-6330 ext. 10



**PowerWorld**  
Corporation

# Introduction



- Experience with RAS and Relays
  - For 15 years we have been working with engineers on directly modeling the logic and actions that describe RAS and Relays in software
  - We have a lot of experience looking at descriptions of RAS and encoding them in software
  - We have a lot of experience adding new features to software to permit additional wrinkles in how RAS is defined.
  - This feature set has evolved incrementally over the past 15 years doing several dozen very small projects to enhance the software feature set

# A Simple Definition



- Protection System Device (typically a Relay)
  - Device monitors a small set of mostly local signals
  - Usually protects a single piece of equipment from damage
  - Actions are typically limited to opening or closing breakers
- Remedial Action Scheme (RAS)
  - Control scheme monitors a larger set of signals (potentially more than local signals)
  - Protects one or more pieces of equipment or prevents larger system-wide or region-wide collapse
  - Actions are more diverse

# What is the Time-Frame?



- These definitions make no distinction about the time-frame of the Relay or RAS actions
  - If milliseconds to a few seconds
    - a **Transient Stability** model is necessary
  - If tens of seconds to minutes with automated response of Relay or RAS → **Power Flow** solution simulation only
  - Minutes of response as a person (the operator) walks over and takes off the shelf the binder entitled “What to do when stuff happens” → again **Power Flow**
- This presentation relates to the steady state power flow solutions

# In software: when are RAS and Relays used?

---



- Clearly not under normal operating conditions
  - When solving a base case power flow solution
    - RAS should not be doing anything
- Within a software tool, when are RAS and Relays going to matter?
  - **Contingency, Contingency, Contingency**
  - RAS will respond to the changes that occur during an unexpected event → **Contingency**
- For RAS and Relay models to be useful
  - first obtain or define a list of contingencies

# Defining Contingencies



- Each **Contingency** has *Name* which is a unique string used to identify it
  - We need the name because this will also be used to identify it when looking at results
- Each **Contingency** is then made up of many **ContingencyElements**
- A **ContingencyElement** describes
  - *Object* to which an action is applied
  - *Action* that occurs
  - *Criteria, CriteriaStatus, TimeDelay* under which the action occurs (Boolean logic, where to apply, ordering)
    - These aren't needed for plain contingency definitions, but will become vital in the RAS modeling which will be discussed in examples

# ContingencyElement



- *Object*
  - many choices of various contingency actions are available
- *Action*
  - What happens to the object
- *Criteria, CriteriaStatus*
  - Logical criteria under which actions are applied
- *Time Delay*
  - Use for ordering

Contingency Element Dialog

Element Type

- Branch
- Generator
- Load
- Switched Shunt
- Bus
- Interface
- Injection Group
- Multi-Section Line
- Series Capacitor
- Phase Shifter
- 3-Winding Transformer
- Line Shunt
- DC Line
- DC Converter
- Area
- Substation
- Abort
- Contingency Block

Choose the Element

Sort by  Name  Number

9746 ( 9746) #1 [6.90 kV]  
9770 ( 9770) #1 [2.30 kV]  
9783 ( 9783) #1 [13.8 kV]  
9784 ( 9784) #1 [13.8 kV]  
9786 ( 9786) #1 [13.8 kV]  
9787 ( 9787) #1 [13.8 kV]  
9788 ( 9788) #1 [13.8 kV]  
9791 ( 9791) #1 [6.60 kV]  
9793 ( 9793) #1 [6.60 kV]  
981 ( 981) #DC [345 kV]  
983 ( 983) #1 [6.90 kV]  
983 ( 983) #2 [6.90 kV]  
983 ( 983) #3 [6.90 kV]  
9840 ( 9840) #1 [13.8 kV]  
9841 ( 9841) #1 [13.8 kV]  
9842 ( 9842) #1 [13.8 kV]  
9842 ( 9842) #2 [13.8 kV]  
9960 ( 9960) #1 [115 kV]  
9967 ( 9967) #1 [115 kV]  
Treeville GT1 ( 8195) #1 [13.8 kV]

Action Type

- Open
- Close
- Move
- Set To
- Change By

Amount: 0

Constant Find...

Evaluate in Reference State

Make-up Power Sources ...

in

- MW (const pf)
- Percent
- MW
- Mvar
- Setpoint Voltage

Status: POSTCHECK

Model Criteria: Modify Path 1 Unit 1

Inclusion Filter: Add

Time Delay: 15.000000 seconds

Comment: Control Action #2A

OK Delete Cancel Help

# ContingencyElement Actions



- There are many – we continue adding them as users have a need
  - Opening/Closing of transmission lines and transformers
  - Loss or Recovery of a generator, load, or switched shunt
  - Movement of generation, load, injection group, or switched shunt MWs or Mvars.
  - Changing or Setting of generation, load, injection group, or switched shunt MWs or Mvars
  - Changing or Setting of generator or switched shunt voltage setpoint
  - Opening of all lines connected to a bus
  - Opening of all lines connected to a substation
  - Opening/Closing of all lines or transformers in an interface
  - Open/Close, Set/Change injection group values
    - Many special options with this
  - Bypass/Inservice, Set impedance of series capacitors
  - Changing or Setting of phase-shifter setpoint
  - Open/Close 3-winding transformer
  - Open/Close DC lines, Set/Change DC line setpoints or resistance



# Contingency Dialog



Contingency Analysis

Contingencies Options Results

Records Set Columns

	Label	Skip	Category	Processed	Solved	Post-C AUX
1	N-1: Path 1A to 1B	NO		YES	YES	none
2	N-1: Treeville to Refinery	NO		YES	YES	none
3	N-1-1: Treeville to Refinery and North Line	NO		YES	YES	none
4	BSBF: Bus 8176	NO		YES	YES	none
5	N-1: North Line	NO		YES	YES	none
6	N-1-1: Treeville to Refinery and Second North Line	NO		YES	YES	none
7	N-1: Path 2	NO		YES	YES	none

Contingency Analysis

Contingencies Options Results

Modeling

- Basics
  - Generator Post-Contingency AGC
  - Bus Load Throw Over
  - Generator Maximum MW Response
  - Generator Line Drop and RCC
  - Post-Contingency Auxiliary File
  - Transient Models
- Limit Monitoring
- Contingency Definitions
  - All Contingency Elements
  - Contingency Blocks
  - Contingency Block Elements
  - Remedial Actions
  - Remedial Action Elements
  - Contingency Global Actions
  - Model Conditions

All Contingency Elements

Records Set Columns

	Contingency Label	Actions - PW File Format	Model Criteria	Status	Time Delay	Comment
1	N-1: Path 1A to 1B	BRANCH 8222 8194 1 OPEN		ALWAYS	0	
2	N-1: Path 1A to 1B	BRANCH 8222 8226 1 OPEN		ALWAYS	0	
3	N-1: Treeville to Refinery	BRANCH 10440 8194 1 OPEN		ALWAYS	0	
4	N-1-1: Treeville to Refinery and North L	BRANCH 8220 8194 1 OPEN		ALWAYS	0	
5	N-1-1: Treeville to Refinery and North L	BRANCH 8180 8220 1 OPEN		ALWAYS	0	
6	N-1-1: Treeville to Refinery and North L	BRANCH 10440 8194 1 OPEN		ALWAYS	0	
7	BSBF: Bus 8176	BUS 8176 OPEN		ALWAYS	0	
8	N-1: North Line	BRANCH 8220 8194 1 OPEN		ALWAYS	0	
9	N-1: North Line	BRANCH 8180 8220 1 OPEN		ALWAYS	0	
10	N-1-1: Treeville to Refinery and Second	BRANCH 8178 8179 1 OPEN		ALWAYS	0	
11	N-1-1: Treeville to Refinery and Second	BRANCH 10440 8194 1 OPEN		ALWAYS	0	
12	N-1: Path 2	BRANCH 10491 10440 1 OPEN		ALWAYS	0	
13	N-1: Path 2	BRANCH 10491 7453 1 OPEN		ALWAYS	0	

Status Finished with 4 Violations, 0 Unsolvable, and 0 Aborted Contingencies. Initial State Restored.

Refresh Displays After Each Contingency

Load Auto Insert Save Other > Start Run Close Help

# Contingency Definitions



- Final output for sharing with others

```
Contingency (Name, Category, Skip, Memo)
```

```
{  
"L-2_Roughrider-Raven 2&3" "Double" "NO" "My Memo A"  
"L-2_Roughrider-Raven 1&2" "Double" "NO" "My Memo A"  
"L_Falcon-PatriotC1"      "Single" "NO" "My Memo A"  
"T_Falcon-TitanC1"       "Single" "NO" "My Memo A"  
}
```

```
ContingencyElement (Contingency, Object, Action, Criteria, CriteriaStatus,  
                    TimeDelay, Comment)
```

```
{  
"L-2_Roughrider-Raven 2&3" "BRANCH 15 54 2" "OPEN" "" "CHECK" 0 ""  
"L-2_Roughrider-Raven 2&3" "BRANCH 15 54 3" "OPEN" "" "CHECK" 0 ""  
"L-2_Roughrider-Raven 1&2" "BRANCH 15 54 1" "OPEN" "" "CHECK" 0 ""  
"L-2_Roughrider-Raven 1&2" "BRANCH 15 54 2" "OPEN" "" "CHECK" 0 ""  
"L_Falcon-PatriotC1"      "BRANCH 10 13 1" "OPEN" "" "CHECK" 0 ""  
"T_Falcon-TitanC1"       "BRANCH 10 39 1" "OPEN" "" "CHECK" 0 ""  
}
```

# What do you need to model RAS in the Power Flow



- The description of a RAS is really the same as a Contingency.
  - A list of actions that occur
    - Actions become more complex though.
      - Trip MWs from a group of generators equal to 50% of the flow on an interface (or use a 2D lookup table to determine what to trip)
  - The *Criteria* is vital here
    - these actions do not always occur
      - Must describe the Boolean logic of when these actions occur
  - The CriteriaStatus and TimeDelay
    - describe when and at what point in solution process to include
- Question
  - *How do you implement the Boolean checks of when to trigger the RAS and the ordering of actions?*
  - *How do you handle the lookup tables, expressions used in more complex actions?*

# Traditional Modeling of RAS in Software Studies



- Often the more complex features are provided manually by you the power engineer
  - Boolean logic of when to apply
    - May know that taking a double-line outage will cause RAS to be applied
    - Thus if contingency is for double line outage just include RAS actions
  - Figure out the “RAS Arming” level from the base case
  - Solve the contingency →  
If a line is overloaded then open it and resolve



# Other Common Shortcuts



- Run a very detailed study of RAS
  - Particular contingencies cause violations (line overloads, bus voltage violations)
  - Verify that your RAS fixes these violations
- Then, for the next 1, 2, ... (10?) years assume the RAS always works to fix these problems
- Functionally this means
  - Run your list of 100s or 1,000s of contingencies
  - Manually wade through the 1,000s of violations that occur and just ignore what is handled by RAS (experience)



# Another common treatment for RAS



- Power engineer writes custom code to automate all the processes from the previous 2 slides
  - Write custom code to implement RAS
  - Write custom code to remove particular violations from your output reporting



# Problems with these approaches (1/2)

---



- Reproducibility
  - Can you replicate a manual process and get the same answer over and over?
- Validity of assumptions
  - The RAS was designed many years ago. Are you sure your RAS is still always fixing the problems for which it was designed?
- Narrow assumption of when RAS is implemented
  - You are limiting the application of RAS to specific contingencies
  - This prevents you from seeing a cascading outage caused by several RAS interacting with each other

# Problems with these approaches (2/2)



- Documentation and Sharing
  - How do you communicate what your RAS does to another utility or neighbor with the precision necessary to model it in software?
  - Custom code → who manages and takes support calls for that code?
- Input Data Management
  - Much of this leads to manually created contingency lists that are tuned for a particular operating condition
    - RAS arming, Boolean criteria
- Training – Human Resource Problem
  - Rely completely on the power engineer's experience which takes many years to develop
    - Engineers move jobs within a company
    - Engineers switch companies
  - How do you train new engineers or communicate all these assumptions? And do it quickly!

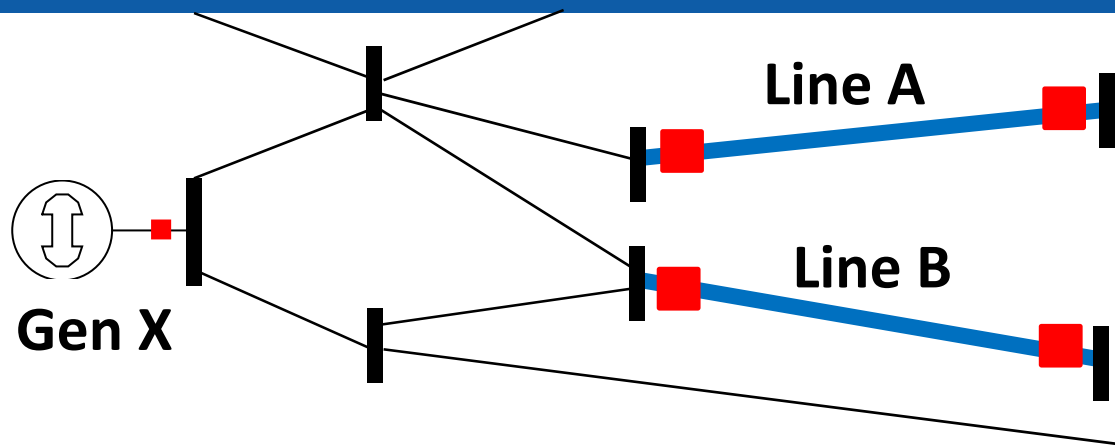


# Another Hidden Problem



- PowerWorld's experience working with utility engineers
  - The engineers running power flow and transient stability studies have a general idea of how RAS functions
  - However, they may not manage and design the RAS itself, so they may miss details
- The implementation of when to “arm” and how much is very specific
  - The details matter!
  - Consider a very simple RAS example next

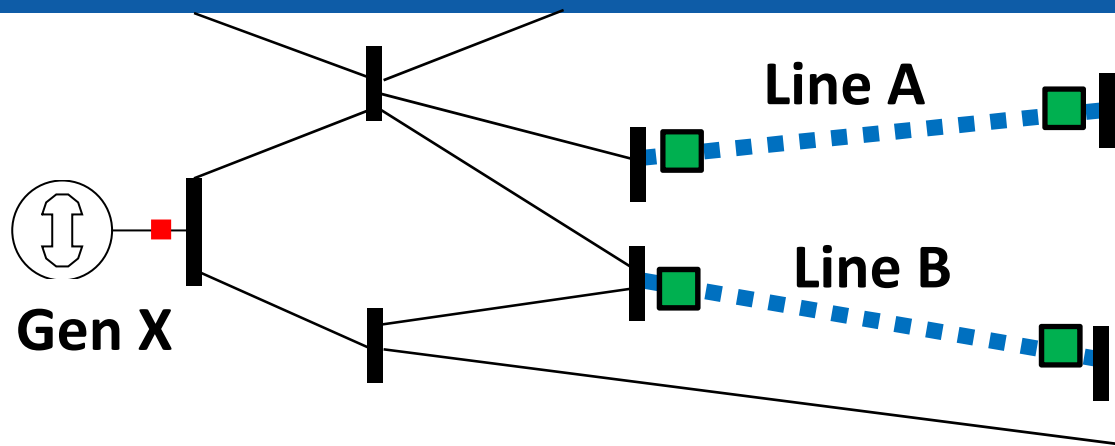
# “Simple RAS”



- General Description of RAS
  - If two transmission lines (**Line A** and **Line B**) are tripped  
→ then trip a generator (**Gen X**)
- RemedialAction definition seems simple
  - *Object* = **Gen X**
  - *Action* = OPEN
  - *Criteria* = (**Line A** is OPEN) AND (**Line B** is OPEN)

# Wait!

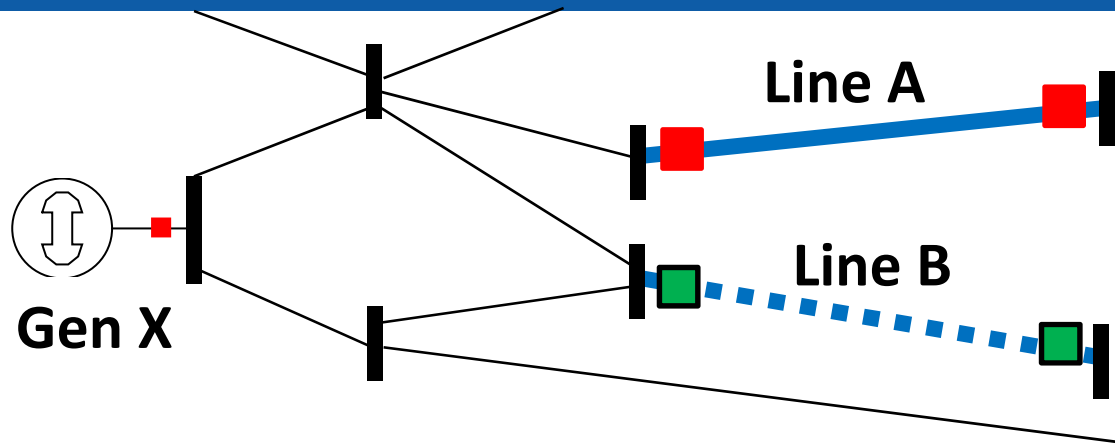
## Differences in Initial Case



- What if **Line A** and **Line B** are out for maintenance this week?
  - Using our “Simple RAS” then this means that the Criteria always evaluates as TRUE!
  - Running a contingency analysis tool with this RAS defined would trip **Gen X** under every contingency
  - In this example, clearly the RAS criteria should evaluate to FALSE

# Wait!

## More differences in Initial Case



- What if **Line B** is out for maintenance this week?
- Using our “Simple RAS” then this means that if **Line A** is opened during the contingency then **Gen X** will be tripped
  - That *might* be correct, ... but
  - It may *not* be correct for some RAS
    - For some if **Line B** is OPEN in the reference case then the RAS will *not be armed!* Tripping **Gen X** is not correct then
- The engineers running power flow/stability need to go talk with the RAS engineers to learn these details

# Communication



- RAS design requires that there be redundant communication systems for RAS to prevent communication failure
- We also need the human communication between different groups of engineers to work as well
- Our experience is there are 3 groups here (though at smaller organizations there is overlap)
  1. “Planning or Operation Engineers” who run power flow and transient stability studies (these could be separate groups too)
  2. “Relay Engineers” who manage and design the system protection
  3. “RAS Engineers” who manage and design the RAS
- Most in audience today are in Group 1, but we all need to engage with folks in Group 2 and 3 to get the details of RAS and Relay modeling correct

# Note on Presentation



- For presentation of a concept, I will do the following
  1. Describe the RAS logic
  2. Show how that would be defined in a Dialog
  3. Show how that is represented in the RAS and Contingency File Format AUX file (text file)
- You'll see that the AUX file represents a bunch of tables
  - Power engineers are accustomed to tables of data about buses, gens, lines, areas, zones, owners, etc...
  - This is just a bunch of tables describing the logic, lookups, etc...

# Treeville RAS Overview



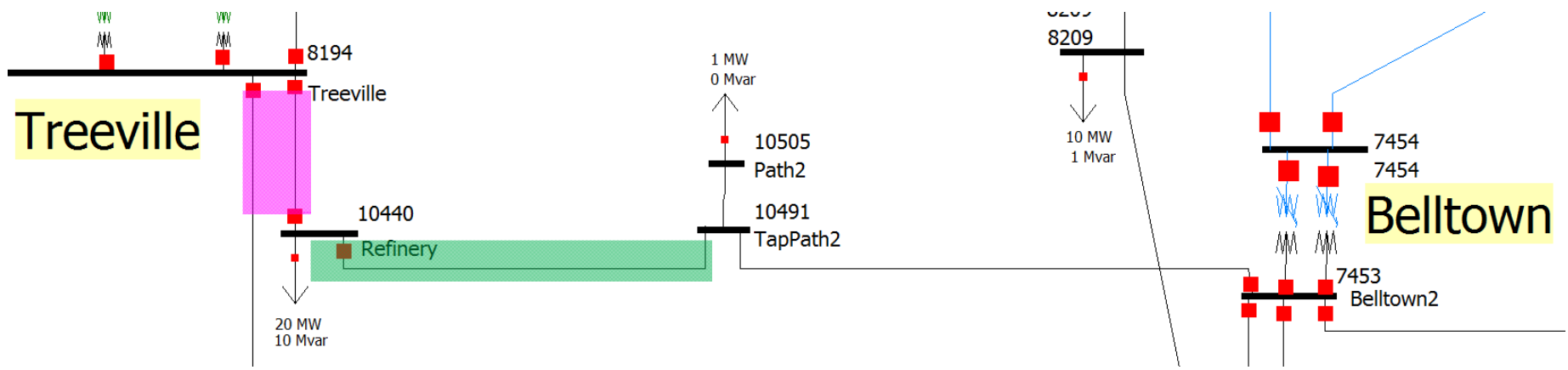
- 4 Separate Control Actions Possible
  1. Ramp Treeville generation down by 40 MW in two minutes
  2. Trip one Treeville gas combustion turbine
  3. Trip the all the Treeville generators
  4. Trip path of the 115 kV line going to the Refinery
- You will see Time Delay used for Actions 1-3
- Action 4 we'll cover first because it's so simple

# Treeville RAS: Control Action #4



Refinery to TapPath2 > 110%  
of seasonal facility rating  $\longrightarrow$  Trip the Treeville to Refinery Line

- If Green Line Overloads then Trip Pink Line





# Treeville RAS Control Action #4: ModelCondition Dialog



Type =  
Branch

Choose  
Branch

Condition  
for Branch

Define Model Conditions

Model Condition Name: Refinery to TapPath2 > 110%

Buttons: Save, Save As, Rename, Delete

Evaluate in Contingency Reference State

Disable if True in Contingency Reference State

Element Type: Branch

49990 entries

Sort by: Name (selected), Number

Filter: Advanced, Branch

Use Area/Zone Filters

Buttons: Quick, Define, Remove

Search Next

Buttons: Search All

You can use wildcard characters \* or ?

List of entries:

- Path1D (8201) TO Path1C (8215) CKT 1 [115 kV]- [115 kV]
- Path1D (8201) TO Path1E (8223) CKT 1 [115 kV]- [115 kV]
- Path1E (8223) FROM Belltown1 (8177) CKT 1 [115 kV]- [115 kV]
- Path1E (8223) FROM Path1D (8201) CKT 1 [115 kV]- [115 kV]
- Path2 (10505) FROM TapPath2 (10491) CKT 1 [115 kV]- [115 kV]
- Refinery (10440) TO Treeville (8194) CKT 1 [115 kV]- [115 kV]
- Refinery (10440) FROM TapPath2 (10491) CKT 1 [115 kV]- [115 kV]**
- TapPath1 (8222) TO Path1B (8226) CKT 1 [115 kV]- [115 kV]

Meets the filter below

Buttons: Set Filter Same As

Select Filter Type: Branch

Pre-filter using Area/Zone/Owner Filters

Enabled (normally checked)

Logical Comparison: AND (selected), OR, Not AND, Not OR

Condition 1:

Find... Limit Monitoring\% at To Bus greater than or equal to 110

Use Another Filter

Buttons: Add >>, Delete ...

Enable Field to Field Comparisons

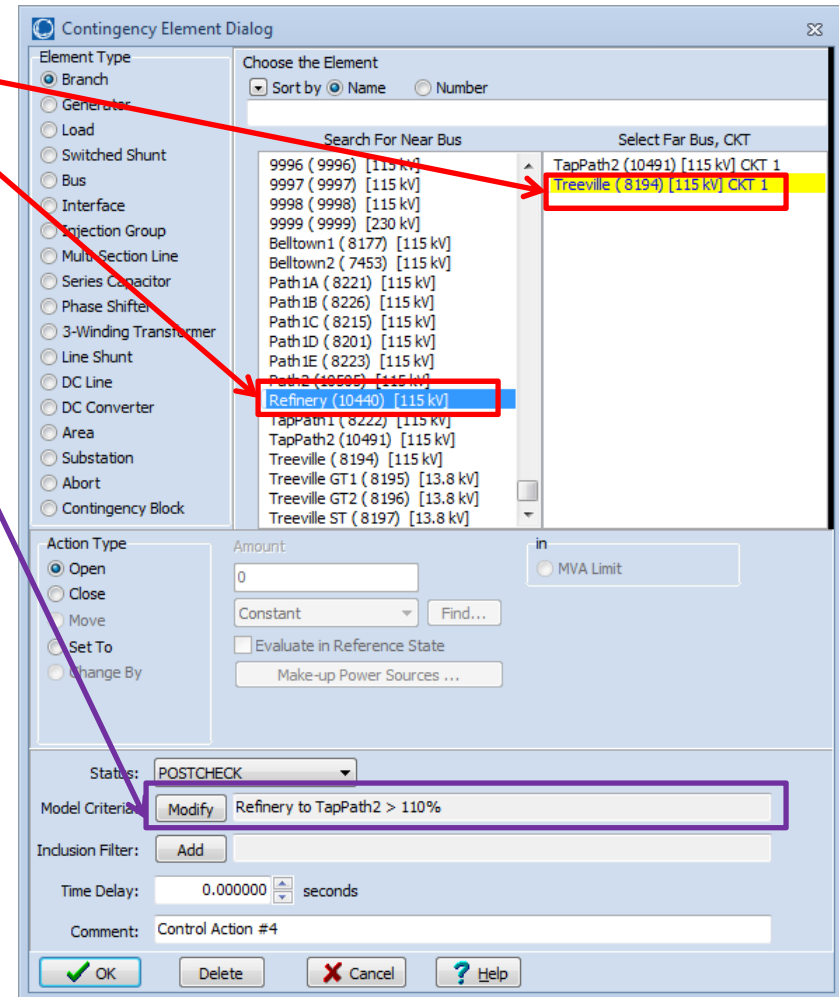
Buttons: OK, Help, Cancel

# Treeville RAS Control Action #4: RemedialAction



Choose Branch to Open

Model Criteria points  
to ModelCondition



# Treeville RAS Control Action #4: File Format

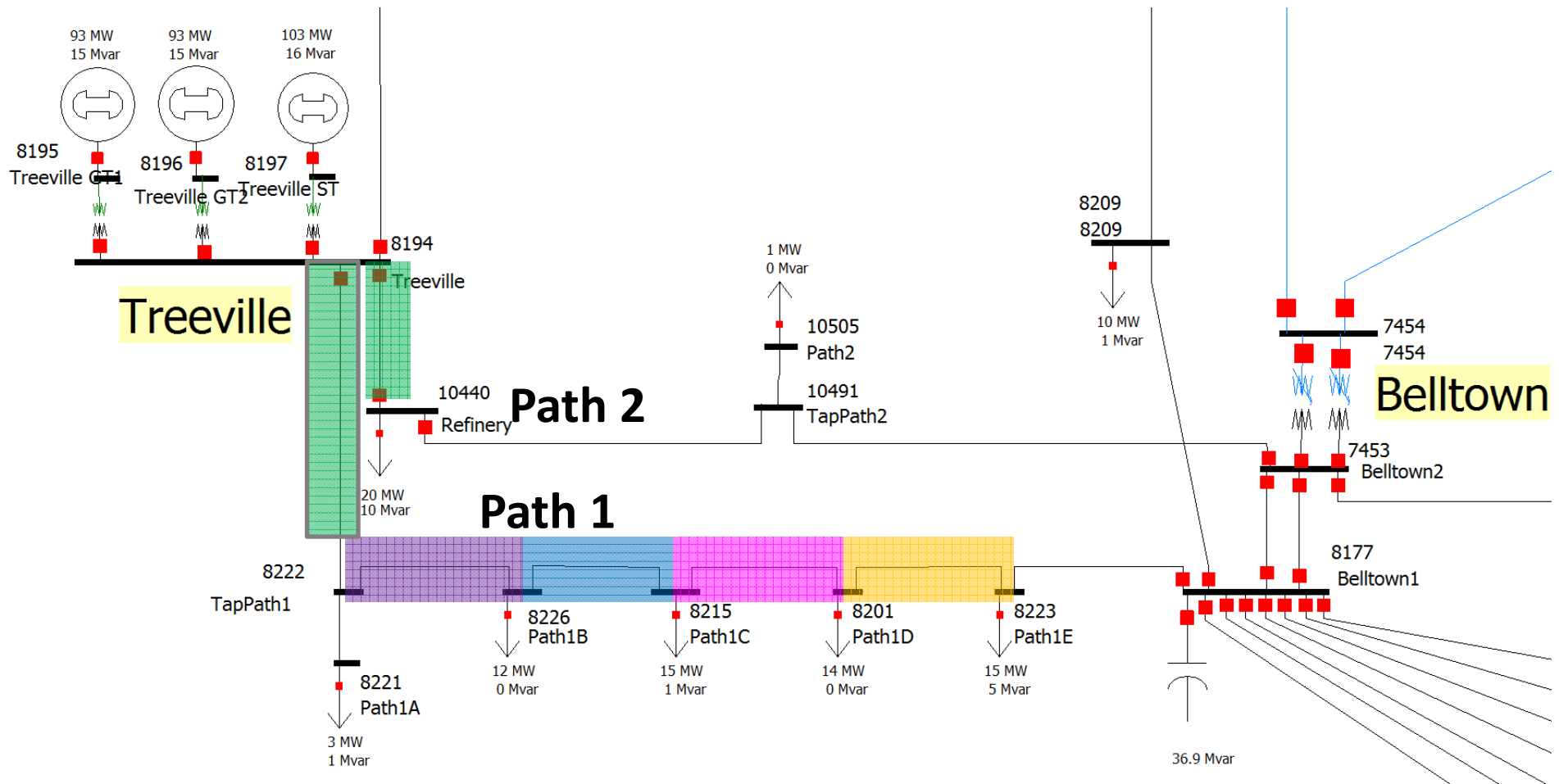


```
MODELCONDITION (Name, Object, FilterObjectType, FilterLogic,
                EvaluateInRef, DisableIfTrueInRef, Memo)
{
"Refinery Tap2 > 110%" "Branch '10491' '10440' '1'" "Branch" "AND" "NO" "NO " ""
}
MODELCONDITIONCONDITION (ModelCondition, CondNum, ObjectField,
                        ConditionType, Value, OtherValue, Absolute)
{
"Refinery Tap2 > 110%" 1 "Percent" ">=" "110" "" "NO "
}

REMEDIALACTION (Name, Skip, Memo)
{
"Treeville Generation Run-Back Scheme" "NO " ""
}
REMEDIALACTIONELEMENT (RemedialAction, Object, Action, Criteria, CriteriaStatus,
                       TimeDelay, InclusionFilter, Comment)
{
"Treeville Generation Run-Back Scheme" "BRANCH 10440 8194 1" "OPEN"
  "Refinery Tap2 > 110%" "POSTCHECK" 0 "" "Control Action #4"
}
```

# Color Codes for Treeville RAS

## Control Actions #1, #2, #3

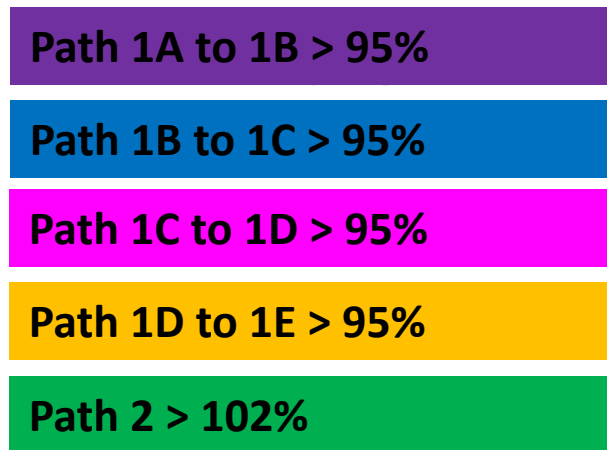


# Treeville RAS: Control Action #1



- Logic provided by utility that manages RAS

## ModelConditions



## Model Filter

## Model Expressions And RemedialAction

Proportionally Reduce Gas Turbine  
Outputs by 40 MW (see not 1)

Set Steam Turbine Outputs for drop of  
GT output (see not 2)

Time Delay:  
120s

**Time Delay = 120 s**

### Notes:

1. CT model expressions are  $-(x1/(x1+x2)*40)$  for unit 1 and  $-(x2/(x1+x2)*40)$  for unit 2 where  $x1$ =actual MW output of unit 1 and  $x2$ =actual MW output of unit 2.

2. ST model expression is  $x5*((x1+x2-40)/(x3+x4))$  where  $x1$ =actual MW output of unit 1,  $x2$ =actual MW output of unit 2,  $x3$ =Pmax of unit 1,  $x4$ =Pmax of unit 2 and  $x5$ =Pmax of steam unit.

# Define Model Conditions: Path 1A to 1B > 95%



Type =  
Branch

Choose  
Branch

Condition  
for Branch

Define Model Conditions

Model Condition Name: Path 1A to 1B > 95%

Buttons: Save, Save As, Rename, Delete

Evaluate in Contingency Reference State

Disable if True in Contingency Reference State

Element Type: Branch

49990 entries

Sort by: Name (selected), Number

Filter: Advanced, Branch

Use Area/Zone Filters

Buttons: Quick, Define, Remove

Search Next

Buttons: Search All

You can use wildcard characters \* or ?

Branches list:

- Belltown2 ( 7453) TO Belltown1 ( 8177) CKT 3 [115 kV]- [115 kV]
- Belltown2 ( 7453) FROM 7454 ( 7454) CKT 1 [115 kV]- [230 kV]
- Belltown2 ( 7453) FROM 7454 ( 7454) CKT 2 [115 kV]- [230 kV]
- Belltown2 ( 7453) FROM TapPath2 (10491) CKT 1 [115 kV]- [115 kV]
- Path1A ( 8221) TO TapPath1 ( 8222) CKT 1 [115 kV]- [115 kV]
- Path1B ( 8226) FROM TapPath1 ( 8222) CKT 1 [115 kV]- [115 kV]
- Path1C ( 8215) TO Path1B ( 8226) CKT 1 [115 kV]- [115 kV]

Meets the filter below: Set Filter Same As

Select Filter Type: Branch

Pre-filter using Area/Zone/Owner Filters

Enabled (normally checked)

Logical Comparison: AND (selected), OR, Not AND, Not OR

Condition 1:

Find... Limit Monitoring\% at From Bus greater than or equal to 95

Use Another Filter

Buttons: Add >>, Delete ...

Enable Field to Field Comparisons

Buttons: OK, Help, Cancel

# Define Model Conditions: Path 2 > 102%



Type =  
Branch

Choose  
Branch

Condition  
for Branch

# Treeville RAS Control Action #1: ModelCondition File Format



```
MODELCONDITION (Name, Object, FilterObjectType, FilterLogic, EvaluateInRef,
                DisableIfTrueInRef, Memo)
{
"Path 1A to 1B > 95%" "Branch '8222' '8226' '1'" "Branch" "AND" "NO" "NO" " " " "
"Path 1B to 1C > 95%" "Branch '8215' '8226' '1'" "Branch" "AND" "NO" "NO" " " " "
"Path 1C to 1D > 95%" "Branch '8201' '8215' '1'" "Branch" "AND" "NO" "NO" " " " "
"Path 1D to 1E > 95%" "Branch '8201' '8223' '1'" "Branch" "AND" "NO" "NO" " " " "
"Path 2 > 102%" "Branch '10491' '10440' '1'" "Branch" "AND" "NO" "NO" " " " "
}
MODELCONDITIONCONDITION (ModelCondition, CondNum, ObjectField, ConditionType, Value,
                          OtherValue, Absolute)
{
"Path 1A to 1B > 95%" 1 "Percent" ">=" "95" "" "NO" "
"Path 1B to 1C > 95%" 1 "Percent" ">=" "95" "" "NO" "
"Path 1C to 1D > 95%" 1 "Percent" ">=" "95" "" "NO" "
"Path 1D to 1E > 95%" 1 "Percent" ">=" "95" "" "NO" "
"Path 2 > 102%" 1 "Percent" ">" "102" "" "NO" "
}
```



# Treeville RAS Control Action #1: ModelFilter Dialog



OR Logic

List of Model  
Conditions

The screenshot shows the 'Filter Conditions' dialog box. At the top, the 'Filter Name' is 'Path 1 > 95% OR Path 2 > 102%'. Below this are buttons for 'Save', 'Save As', 'Rename', 'Delete', and 'View Filter Logic'. The 'Logical Comparison' section has four radio buttons: 'AND', 'OR', 'Not AND', and 'Not OR'. The 'OR' button is selected and highlighted with a red box and a red arrow. Below this is a list of four model conditions, each with a 'Find...' button, a text field, and radio buttons for 'Condition', 'Filter', and 'Not'. The conditions are: 'Model Condition 2' (Path 1C to 1D > 95%), 'Model Condition 3' (Path 1A to 1B > 95%), 'Model Condition 4' (Path 1B to 1C > 95%), and 'Model Condition 5' (Path 2 > 102%). At the bottom are buttons for 'Add >>', 'Delete ...', 'Modify Model Conditions', 'OK', 'Help', and 'Cancel'.

# Treeville RAS Control Action #1: ModelFilter File Format



```
MODELFILTER (Name,Logic,Memo)
{
"Path 1 > 95% OR Path 2 > 102%" "OR" ""
}
MODELFILTERCONDITION (ModelFilter,CondNum,Criteria,Logic)
{
"Path 1 > 95% OR Path 2 > 102%" 1 "Path 1A to 1B > 95%" ""
"Path 1 > 95% OR Path 2 > 102%" 2 "Path 1B to 1C > 95%" ""
"Path 1 > 95% OR Path 2 > 102%" 3 "Path 1C to 1D > 95%" ""
"Path 1 > 95% OR Path 2 > 102%" 4 "Path 1D to 1E > 95%" ""
"Path 1 > 95% OR Path 2 > 102%" 5 "Path 2 > 102%" ""
}
```

# Treeville RAS Control Action #1: Model Expression



Move Steam Plant by the same net MW in proportion to Max MW

Reduce the net MW output of Gas Unit 1 and 2 by a 40 MW (move proportional to present output)

Define Model Expressions

Name: ST Output After Runback

Buttons: Save, Save As, Rename, Delete

Choose the Type of Expression: Expression, Lookup Table

x1= Define...	Gen Treeville GT1 (8195) #1 : Generator's present MW output
x2= Define...	Gen Treeville GT2 (8196) #2 : Generator's present MW output
x3= Define...	Gen Treeville GT1 (8195) #1 : Generator's maximum MW limit
x4= Define...	Gen Treeville GT2 (8196) #2 : Generator's maximum MW limit
x5= Define...	Gen Treeville ST (8197) #L : Generator's maximum MW limit
x6= Define...	Choose a Model Field
x7= Define...	Choose a Model Field
x8= Define...	Choose a Model Field

function ( x1, x2, x3, x4, x5, x6, x7, x8) =

```
x5*((x1+x2-40)/(x3+x4))
```

Buttons: OK, Help, Cancel

Define Model Expressions

Name: GT2 Run Back Value

Buttons: Save, Save As, Rename, Delete

function ( x1, x2, x3, x4, x5, x6, x7, x8) =

```
-(x2/(x1+x2)*40)
```

Define Model Expressions

Name: GT1 Run Back Value

Buttons: Save, Save As, Rename, Delete

function ( x1, x2, x3, x4, x5, x6, x7, x8) =

```
-(x1/(x1+x2)*40)
```

# Treeville RAS Control Action #1: ModelExpression File Format



```
MODELEXPRESSION (Name,Type,Expression,Memo,  
    Object1,x1,BlankZero1,Object2,x2,BlankZero2,  
    Object3,x3,BlankZero3,Object4,x4,BlankZero4,  
    Object5,x5,BlankZero5,Object6,x6,BlankZero6,  
    Object7,x7,BlankZero7,Object8,x8,BlankZero8)  
  
{  
"GT1 Run Back Value"      "Expression" "-(x1/(x1+x2)*40)" ""  
    "Gen '8195' '1'" "MW" "YES" "Gen '8196' '2'" "MW" "YES"  
    "" "" "NO" "" "" "NO" "" "" "NO" "" "" "NO" "" "" "NO" "" "" "NO" ""  
  
"GT2 Run Back Value"      "Expression" "-(x2/(x1+x2)*40)" ""  
    "Gen '8195' '1'" "MW" "YES" "Gen '8196' '2'" "MW" "YES"  
    "" "" "NO" "" "" "NO" "" "" "NO" "" "" "NO" "" "" "NO" "" "" "NO" ""  
  
"ST Output After Runback" "Expression" "x5*((x1+x2-40)/(x3+x4))" ""  
    "Gen '8195' '1'" "MW" "YES" "Gen '8196' '2'" "MW" "YES"  
    "Gen '8195' '1'" "MWMax" "YES" "Gen '8196' '2'" "MWMax" "YES"  
    "Gen '8197' 'L'" "MWMax" "YES" "" "" "NO" "" "" "NO" "" "" "NO" ""  
}
```

# Treeville RAS Control Action #1: RemedialAction Gas Units



**Choose Generator Unit 2**

**Model Expression for Unit 2**

**Model Criteria points to ModelFilter**

**Time Delay**

**Same but for Steam Unit**

**Same but for Unit 1**

# Treeville RAS Control Action #1: ModelExpression File Format



```
REMEDIALACTION (Name,Skip,Memo)
{
"Treeville Generation Run-Back Scheme" "NO " ""
}

REMEDIALACTIONELEMENT (RemedialAction,Object,Action,Criteria,
                        CriteriaStatus,TimeDelay,InclusionFilter,Comment)
{
"Treeville Generation Run-Back Scheme" "GEN 8195" "CHANGEBY 'GT1 Run Back Value' MW"
    "Path 1 > 95% OR Path 2 > 102%" "POSTCHECK" 120.000000 "" "Control Action #1"

"Treeville Generation Run-Back Scheme" "GEN 8196" "CHANGEBY 'GT2 Run Back Value' MW"
    "Path 1 > 95% OR Path 2 > 102%" "POSTCHECK" 120.000000 "" "Control Action #1"

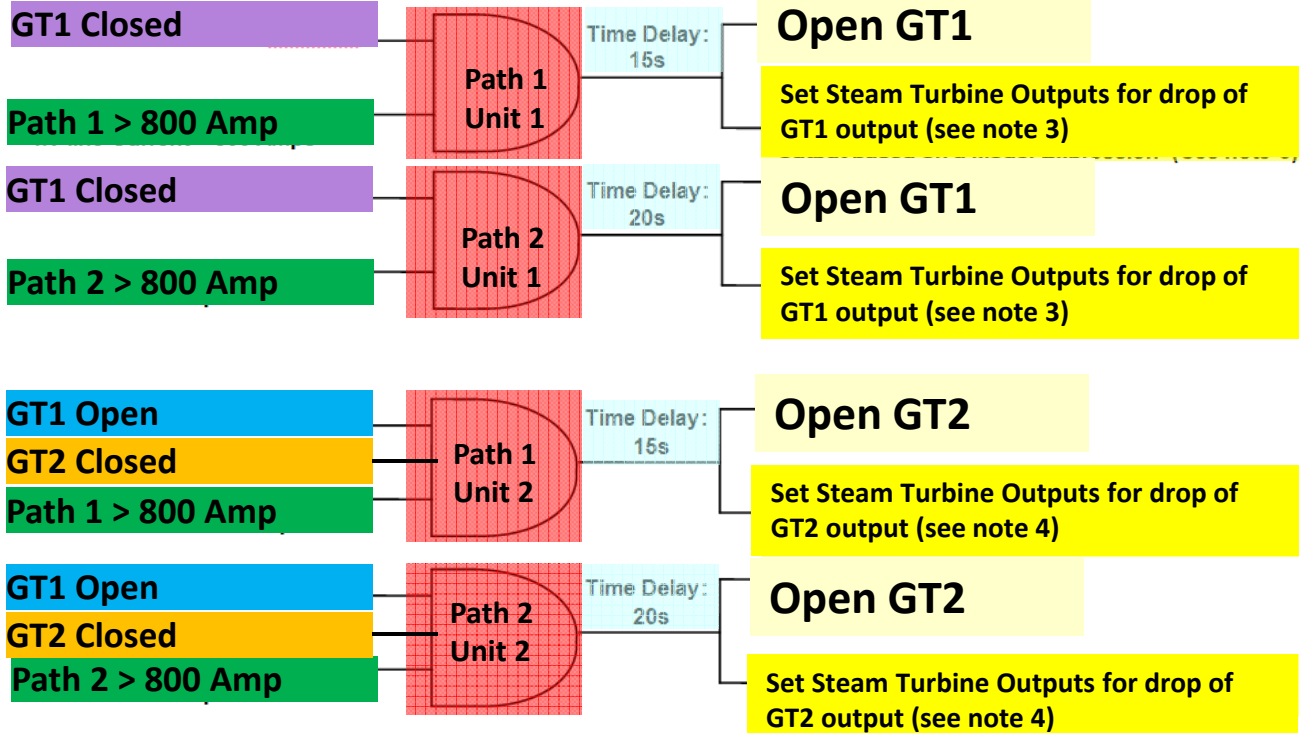
"Treeville Generation Run-Back Scheme" "GEN 8197" "SETTO 'ST Output After Runback' MW"
    "Path 1 > 95% OR Path 2 > 102%" "POSTCHECK" 120.000000 "" "Control Action #1"
}
```

# Treeville RAS: Control Action #2



## ModelConditions

Time Delay  
= 15 or 20 s



## ModelExpressions RemedialActions

3. ST model expression is  $x5 * ((x2) / (x3 + x4))$  where  $x1$ =actual MW output of unit 1,  $x2$ =actual MW output of unit 2,  $x3$ =Pmax of unit 1,  $x4$ =Pmax of unit 2 and  $x5$ =Pmax of steam unit .

4. ST model expression is  $x5 * ((x1) / (x3 + x4))$  where  $x1$ =actual MW output of unit 1,  $x2$ =actual MW output of unit 2,  $x3$ =Pmax of unit 1,  $x4$ =Pmax of unit 2 and  $x5$ =Pmax of steam unit .

Set Steam Turbine Outputs for drop of GT1 output (see note 3)

## ModelFilters

# Treeville RAS Control Action #2: ModelCondition ModelFilter



```

MODELCONDITION (Name, Object, FilterObjectType, FilterLogic, EvaluateInRef, DisableIfTrueInRef, Memo)
{
"GT1 Closed"      "Gen '8195' '1'"      "Gen" "AND" "NO" "NO" " " ""
"GT1 Open"       "Gen '8195' '1'"      "Gen" "AND" "NO" "NO" " " ""
"GT2 Closed"     "Gen '8196' '2'"      "Gen" "AND" "NO" "NO" " " ""
"Path 1 > 800 A" "Branch '8222' '8194' '1'" "Branch" "AND" "NO" "NO" " " ""
"Path 2 > 800 A" "Branch '10440' '8194' '1'" "Branch" "AND" "NO" "NO" " " ""
}
MODELCONDITIONCONDITION (ModelCondition, CondNum,
    ObjectField, ConditionType, Value, OtherValue, Absolute)
{
"GT1 Closed"      1 "Status" "startswith" "C" "" "NO" "
"GT1 Open"       1 "Status" "startswith" "O" "" "NO" "
"GT2 Closed"     1 "Status" "startswith" "C" "" "NO" "
"Path 1 > 800 A" 1 "AmpsMax" ">" "800" "" "NO" "
"Path 2 > 800 A" 1 "AmpsMax" ">" "800" "" "NO" "
}

```

```

MODELFILTER (Name, Logic, Memo)
{
"Path 1 Unit 1" "AND" ""
"Path 2 Unit 1" "AND" ""
"Path 1 Unit 2" "AND" ""
"Path 2 Unit 2" "AND" ""
}

```

```

MODELFILTERCONDITION (ModelFilter, CondNum, Criteria, Logic)
{
"Path 1 Unit 1" 1 "GT1 Closed"      ""
"Path 1 Unit 1" 2 "Path 1 > 800 A" ""
"Path 2 Unit 1" 1 "GT1 Closed"      ""
"Path 2 Unit 1" 2 "Path 2 > 800 A" ""
"Path 1 Unit 2" 1 "GT1 Open"        ""
"Path 1 Unit 2" 2 "GT2 Closed"      ""
"Path 1 Unit 2" 3 "Path 1 > 800 A" ""
"Path 2 Unit 2" 1 "GT1 Open"        ""
"Path 2 Unit 2" 2 "GT2 Closed"      ""
"Path 2 Unit 2" 3 "Path 2 > 800 A" ""
}

```



# Treenville RAS Control Action #2: ModelExpression, RemedialAction



```

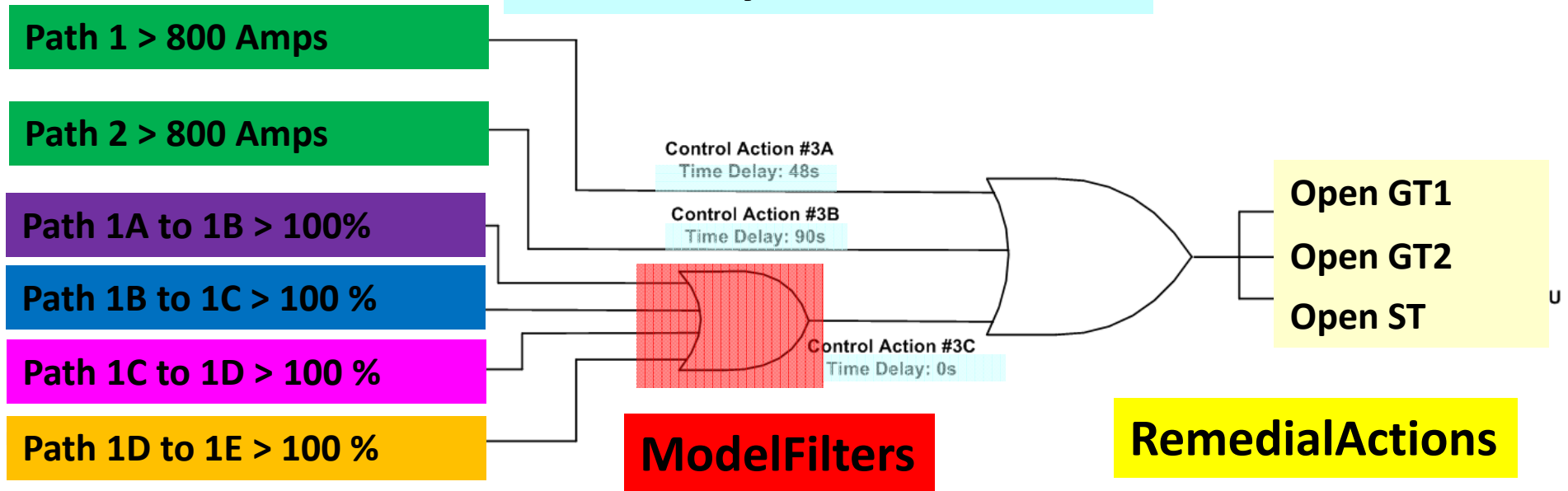
MODELEXPRESSION (Name,Type,Expression,Memo,Object1,x1,BlankZero1,Object2,x2,BlankZero2,Object3,x3,
BlankZero3,Object4,x4,BlankZero4,Object5,x5,BlankZero5,Object6,x6,BlankZero6,
Object7,x7,BlankZero7,Object8,x8,BlankZero8)
{
"ST Output for GT1 Outage" "Expression" "x5*((x2)/(x3+x4))" ""
    "Gen '8195' '1'" "MW" "YES" "Gen '8196' '2'" "MW" "YES"
    "Gen '8195' '1'" "MWMax" "YES" "Gen '8196' '2'" "MWMax" "YES"
    "Gen '8197' 'L'" "MWMax" "YES" "" "" "NO" "" "" "NO" "" "" "NO" ""
"ST Output for GT2 Outage" "Expression" "x5*((x1)/(x3+x4))" ""
    "Gen '8195' '1'" "MW" "YES" "Gen '8196' '2'" "MW" "YES"
    "Gen '8195' '1'" "MWMax" "YES" "Gen '8196' '2'" "MWMax" "YES"
    "Gen '8197' 'L'" "MWMax" "YES" "" "" "NO" "" "" "NO" "" "" "NO" ""
}
REMEDIALACTION (Name,Skip,Memo)
{
"Treenville Generation Run-Back Scheme" "NO" ""
}
REMEDIALACTIONELEMENT (RemedialAction,Object,Action,Criteria,CriteriaStatus,TimeDelay,InclusionFilter,
Comment)
{
"Treenville Generation Run-Back Scheme" "GEN 8195 1" "OPEN"
    "Path 1 Unit 1" "POSTCHECK" 15.000000 "" "Control Action #2A"
"Treenville Generation Run-Back Scheme" "GEN 8196 2" "OPEN"
    "Path 1 Unit 1" "POSTCHECK" 15.000000 "" "Control Action #2A"
"Treenville Generation Run-Back Scheme" "GEN 8195 1" "OPEN"
    "Path 2 Unit 1" "POSTCHECK" 20.000000 "" "Control Action #2B"
"Treenville Generation Run-Back Scheme" "GEN 8196 2" "OPEN"
    "Path 2 Unit 1" "POSTCHECK" 20.000000 "" "Control Action #2B"
"Treenville Generation Run-Back Scheme" "GEN 8197" "SETTO 'ST Output for GT1 Outage' MW"
    "Path 1 Unit 2" "POSTCHECK" 15.000000 "" "Control Action #2A"
"Treenville Generation Run-Back Scheme" "GEN 8197" "SETTO 'ST Output for GT2 Outage' MW"
    "Path 1 Unit 2" "POSTCHECK" 15.000000 "" "Control Action #2A"
"Treenville Generation Run-Back Scheme" "GEN 8197" "SETTO 'ST Output for GT1 Outage' MW"
    "Path 2 Unit 2" "POSTCHECK" 20.000000 "" "Control Action #2B"
"Treenville Generation Run-Back Scheme" "GEN 8197" "SETTO 'ST Output for GT2 Outage' MW"
    "Path 2 Unit 2" "POSTCHECK" 20.000000 "" "Control Action #2B"
}

```

# Treeville RAS: Control Action #3



Time Delay= 0, 48 or 90 s



ModelConditions

RemedialActions

# Treeville RAS Control Action #2: ModelCondition ModelFilter



```
MODELCONDITION (Name, Object, FilterObjectType, FilterLogic, EvaluateInRef, DisableIfTrueInRef, Memo)
{
  "Path 1 > 800 A"          "Branch '8222' '8194' '1'" "Branch" "AND" "NO" "NO" " "
  "Path 2 > 800 A"          "Branch '10440' '8194' '1'" "Branch" "AND" "NO" "NO" " "
  "Path 1A to 1B > 100%"    "Branch '8222' '8226' '1'" "Branch" "AND" "NO" "NO" " "
  "Path 1B to 1C > 100%"    "Branch '8215' '8226' '1'" "Branch" "AND" "NO" "NO" " "
  "Path 1C to 1D > 100%"    "Branch '8201' '8215' '1'" "Branch" "AND" "NO" "NO" " "
  "Path 1D to 1E > 100%"    "Branch '8201' '8223' '1'" "Branch" "AND" "NO" "NO" " "
}
MODELCONDITIONCONDITION (ModelCondition, CondNum,
  ObjectField, ConditionType, Value, OtherValue, Absolute)
{
  "Path 1 > 800 A" 1 "AmpsMax" ">" "800" "" "NO "
  "Path 2 > 800 A" 1 "AmpsMax" ">" "800" "" "NO "
  "Path 1A to 1B > 100%" 1 "Percent" ">" "100" "" "NO "
  "Path 1B to 1C > 100%" 1 "Percent" ">=" "100" "" "NO "
  "Path 1C to 1D > 100%" 1 "Percent" ">=" "100" "" "NO "
  "Path 1D to 1E > 100%" 1 "Percent" ">=" "100" "" "NO "
}
```

```
MODELFILTER (Name, Logic, Memo)
{
  "Path 1 > 100%" "OR" ""
}
MODELFILTERCONDITION
(ModelFilter, CondNum, Criteria, Logic)
{
  "Path 1 > 100%" 1 "Path 1A to 1B > 100%" ""
  "Path 1 > 100%" 2 "Path 1B to 1C > 100%" ""
  "Path 1 > 100%" 3 "Path 1C to 1D > 100%" ""
  "Path 1 > 100%" 4 "Path 1D to 1E > 100%" ""
}
```

# Treeville RAS Control Action #2: ModelExpression, RemedialAction



```
REMEDIALACTION (Name,Skip,Memo)
{
"Treeville Generation Run-Back Scheme" "NO " ""
}
REMEDIALACTIONELEMENT (RemedialAction,Object,Action,
Criteria,CriteriaStatus,TimeDelay,InclusionFilter,Comment)
{
// Trip GT1
"Treeville Generation Run-Back Scheme" "BRANCH 8195 8194 1" "OPEN" "Path 1 > 800 A"
"POSTCHECK" 90.000000 "" "Control Action #3B"
"Treeville Generation Run-Back Scheme" "BRANCH 8195 8194 1" "OPEN" "Path 2 > 800 A"
"POSTCHECK" 48.000000 "" "Control Action #3A"
"Treeville Generation Run-Back Scheme" "BRANCH 8195 8194 1" "OPEN" "Path 1 > 100%"
"POSTCHECK" 0 "" "Control Action #3C"
// Trip GT2
"Treeville Generation Run-Back Scheme" "BRANCH 8196 8194 1" "OPEN" "Path 1 > 800 A"
"POSTCHECK" 90.000000 "" "Control Action #3B"
"Treeville Generation Run-Back Scheme" "BRANCH 8196 8194 1" "OPEN" "Path 2 > 800 A"
"POSTCHECK" 48.000000 "" "Control Action #3A"
"Treeville Generation Run-Back Scheme" "BRANCH 8196 8194 1" "OPEN" "Path 1 > 100%"
"POSTCHECK" 0 "" "Control Action #3C"
// Trip ST
"Treeville Generation Run-Back Scheme" "BRANCH 8197 8194 1" "OPEN" "Path 1 > 800 A"
"POSTCHECK" 90.000000 "" "Control Action #3B"
"Treeville Generation Run-Back Scheme" "BRANCH 8197 8194 1" "OPEN" "Path 2 > 800 A"
"POSTCHECK" 48.000000 "" "Control Action #3A"
"Treeville Generation Run-Back Scheme" "BRANCH 8197 8194 1" "OPEN" "Path 1 > 100%"
"POSTCHECK" 0 "" "Control Action #3C"
}
```

# Treeville RAS:

## Note On Time Delay Parameter



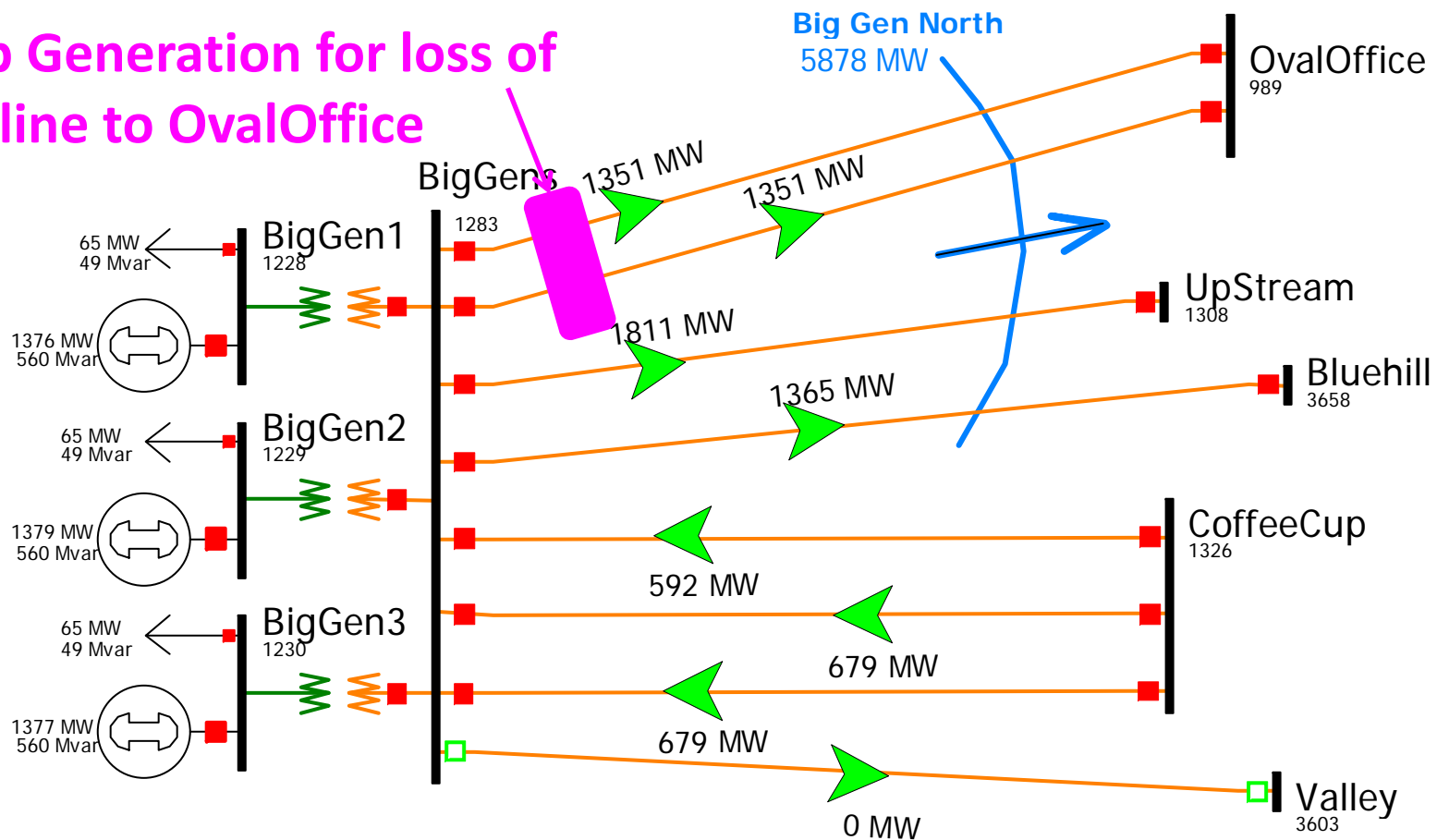
- For the Treeville RAS actions, there are time-delays assigned which are important
- Time Delays are associated with monitoring current on two branches leaving Treeville
  - Action #1: 120 seconds for 102% overload
  - Action #2: 15 or 20 seconds for over 800 Amps then trip only one gas plant (and modify steam)
  - Action #3: 48 or 90 seconds for over 800 Amps then trip both gas plants and steam plants
- Need to do Action #2 first which trips only one generator
  - That may make unnecessary Action #3 which trips all the generators at Treeville
- Details Matter!

# Example #2: Double Line Outage with Generation Dropping



- Consider system below

Drop Generation for loss of one line to OvalOffice



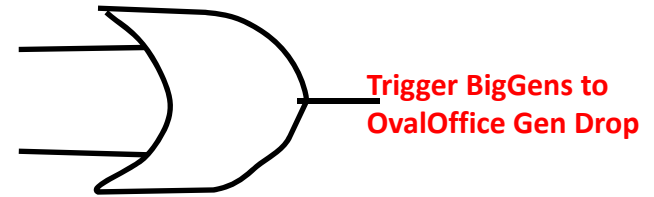
# BigGens to OvalOffice Gen Drop



- RAS is triggered when either of lines from BigGens to OvalOffice are opened by the contingency
  - If one line is out in the initial system (for example for maintenance), the RAS will also trigger if the second line is opened during the contingency
- When the RAS is triggered, it will trip generation from the 3 BigGens units based on a lookup table which is a function of MW flow on the interface “Big Gen North”

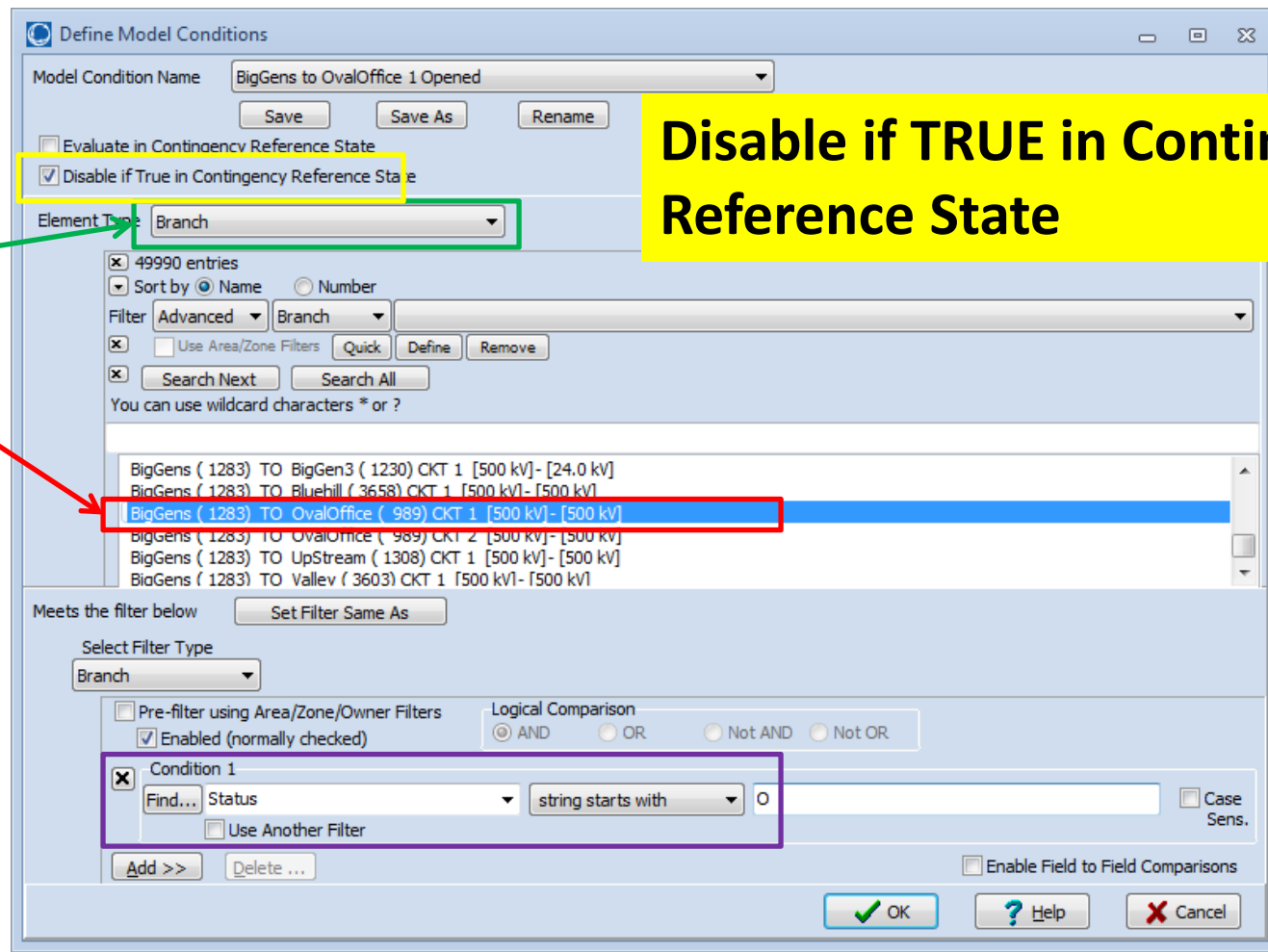
BigGens – OvalOffice  
Circuit 1 Open

BigGens – OvalOffice  
Circuit 2 Open



Big Gen North MW Flow	Generation Change
0	0
3000	-500
4000	-1200
5000	-2000

# BigGens to OvalOffice Gen Drop: ModelCondition Dialog



**Disable if TRUE in Contingency Reference State**

Type = Branch

Choose Branch

Condition for Branch



# BigGens to OvalOffice Gen Drop: ModelFilter Dialog



OR Logic

List of Model  
Conditions

Filter Conditions

Filter Name: One Of BigGens to OvalOffice Opened

Buttons: Save, Save As, Rename, Delete, View Filter Logic

Logical Comparison:  AND  OR  Not AND  Not OR

Model Condition 1: Find... BigGens to OvalOffice 1 Opened  Condition  Filter  Not

Model Condition 2: Find... BigGens to OvalOffice 2 Opened  Condition  Filter  Not

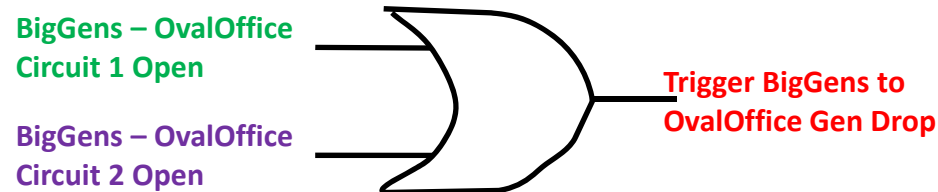
Buttons: Add >>, Delete ..., Modify Model Conditions

Buttons: OK, Help, Cancel

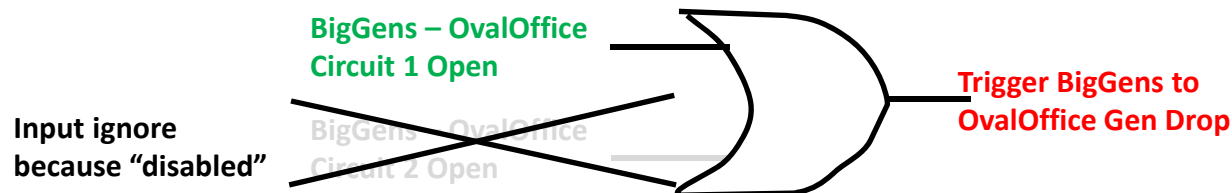
# Disable if TRUE in Contingency Reference State Meaning



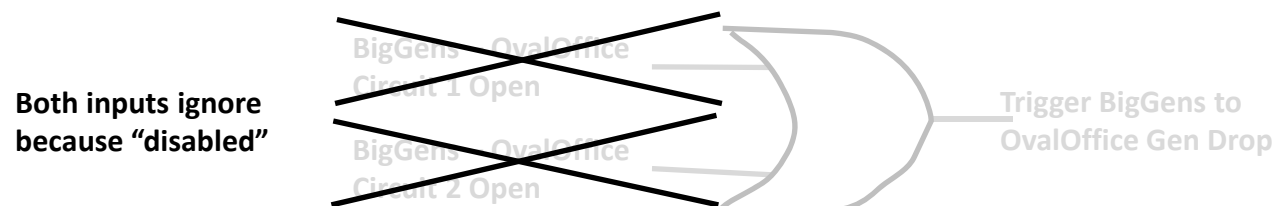
- If both lines are inservice then logic looks like



- If only circuit 1 is inservice then logic changes to



- If both lines are out of service then all inputs to the model filter are all "disabled" and thus it's output is considered "Disabled"
  - If the model filter is fed into another filter that input is disabled
  - If model filter is used directly, it returns FALSE



# BigGens to OvalOffice Gen Drop: ModelCondition, ModelFilter File



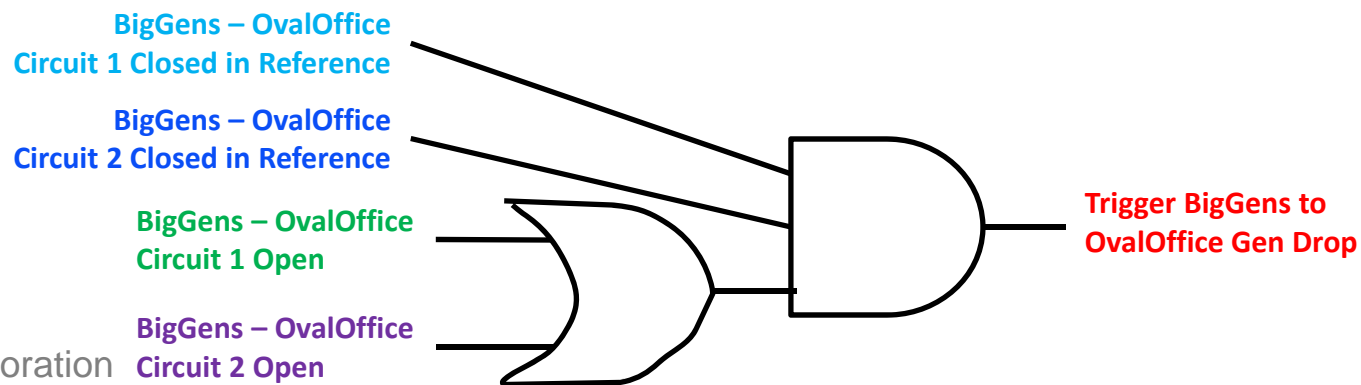
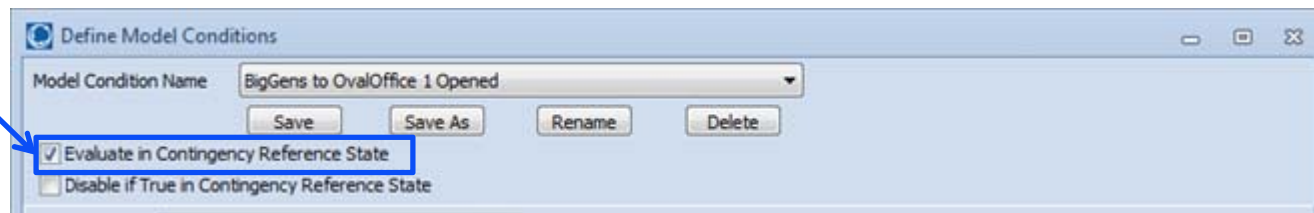
```
MODELCONDITION (Name, Object, FilterObjectType, FilterLogic, EvaluateInRef, DisableIfTrueInRef, Memo)
{
  "BigGens to OvalOffice 1 Opened" "Branch '1283' '989' '1'" "Branch" "AND" "NO" "YES" ""
  "BigGens to OvalOffice 2 Opened" "Branch '1283' '989' '2'" "Branch" "AND" "NO" "YES" ""
}
MODELCONDITIONCONDITION (ModelCondition, CondNum, ObjectField, ConditionType, Value,
                          OtherValue, Absolute)
{
  "BigGens to OvalOffice 1 Opened" 1 "Status" "startswith" "O" "" "NO "
  "BigGens to OvalOffice 2 Opened" 1 "Status" "startswith" "O" "" "NO "
}
MODELFILTER (Name, Logic, Memo)
{
  "One Of BigGens to OvalOffice Opened" "OR" ""
}
MODELFILTERCONDITION (ModelFilter, CondNum, Criteria, Logic)
{
  "One Of BigGens to OvalOffice Opened" 1 "BigGens to OvalOffice 1 Opened" ""
  "One Of BigGens to OvalOffice Opened" 2 "BigGens to OvalOffice 2 Opened" ""
}
```

# Different Logic for Reference State Outage



- What if RAS should only be triggered if both lines are in service initially?
  - Thus if one was out in the reference state you would not trigger the RAS

## Evaluate in Contingency Reference State



# Different Logic for Reference State Outage: File Format



```

MODELCONDITION (Name, Object, FilterObjectType, FilterLogic, EvaluateInRef, DisableIfTrueInRef, Memo)
{
"BigGens to OvalOffice 1 ClosedRef" "Branch '1283' '989' '1'" "Branch" "AND" "YES" "NO" ""
"BigGens to OvalOffice 2 ClosedRef" "Branch '1283' '989' '2'" "Branch" "AND" "YES" "NO" ""
"BigGens to OvalOffice 1 Opened" "Branch '1283' '989' '1'" "Branch" "AND" "NO" "YES" ""
"BigGens to OvalOffice 2 Opened" "Branch '1283' '989' '2'" "Branch" "AND" "NO" "YES" ""
}
MODELCONDITIONCONDITION (ModelCondition, CondNum, ObjectField, ConditionType, Value,
                          OtherValue, Absolute)
{
"BigGens to OvalOffice 1 ClosedRef" 1 "Status" "startswith" "C" "" "NO" "
"BigGens to OvalOffice 2 ClosedRef" 1 "Status" "startswith" "C" "" "NO" "
"BigGens to OvalOffice 1 Opened" 1 "Status" "startswith" "O" "" "NO" "
"BigGens to OvalOffice 2 Opened" 1 "Status" "startswith" "O" "" "NO" "
}
MODELFILTER (Name, Logic, Memo)
{
"One Of BigGens to OvalOffice Opened" "OR" ""
"NewFilterName" "OR" ""
}
MODELFILTERCONDITION (ModelFilter, CondNum, Criteria, Logic)
{
"One Of BigGens to OvalOffice Opened" 1 "BigGens to OvalOffice 1 Opened" ""
"One Of BigGens to OvalOffice Opened" 1 "BigGens to OvalOffice 2 Opened" ""
"NewFilterName" 1 "BigGens to OvalOffice 1 ClosedRef" ""
"NewFilterName" 2 "BigGens to OvalOffice 2 ClosedRef" ""
"NewFilterName" 3 "One Of BigGens to OvalOffice Opened" ""
}

```

# Implementing Gen Drop: Lookup Table, Model Expression



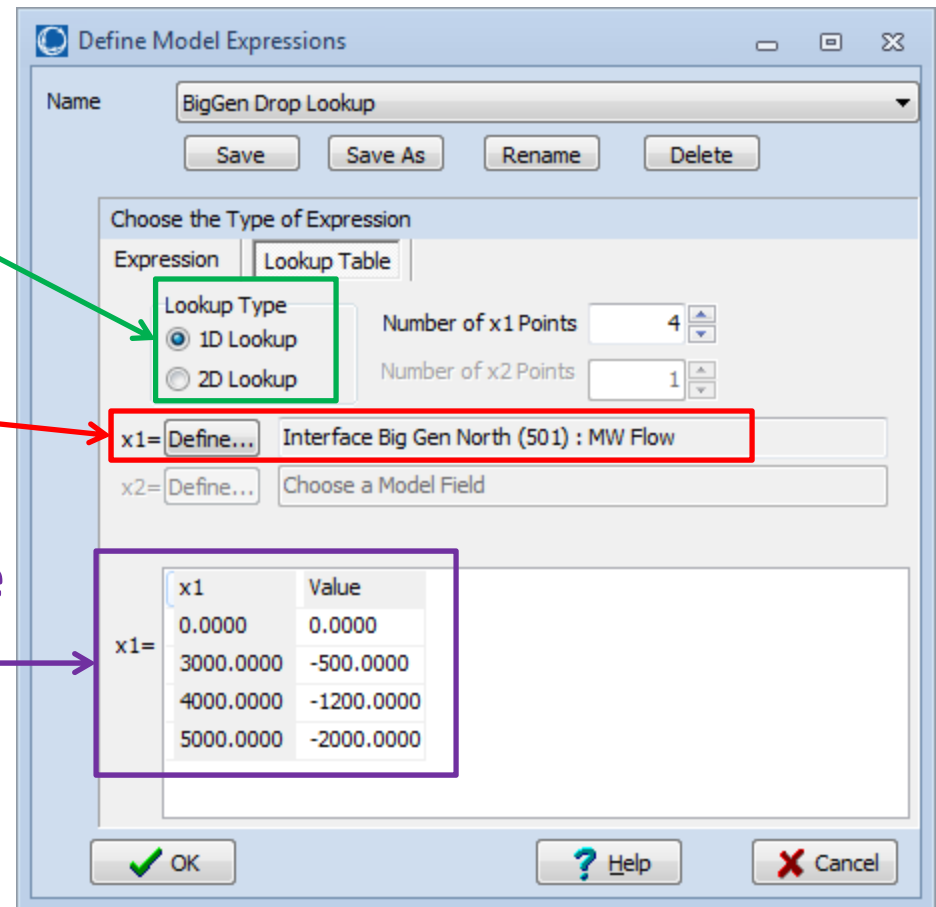
- 1D Lookup Table using a ModelExpression

Choose 1D or 2D lookup

X1 = MW Flow on  
Interface Big Gen North

Big Gen North MW Flow	Generation Change
0	0
3000	-500
4000	-1200
5000	-2000

Lookup Table



# Injection Group Load Dropping



- First you must define an Injection Group

Name

Injection Groups

Name: BigGens

# Gens: 3    % MW Gen Part.: 100.00  
# Loads: 0    % MW Load Part.: 0.00  
# Shunts: 0    % MVR Load Part.: 0.00  
% MVR Shunt Part.: 0.00

Participation Points: Custom

Insert Points (or right-click to insert or delete points)

	Point Type	Number	Name	ID	AutoCalc?	Initial Value	ParFac
1	GEN	1228	BigGen1	1	NO	SPECIFIED	1.00
2	GEN	1229	BigGen2	1	NO	SPECIFIED	2.00
3	GEN	1230	BigGen3	1	NO	SPECIFIED	3.00

? Help    X Cancel

List of Participation Points

# BigGens to OvalOffice Gen Drop: ModelExpression, InjectionGroup



```
MODELEXPRESSION (Name,Type,Expression,Memo,  
Object1,x1,BlankZero1,Object2,x2,BlankZero2,Object3,x3,BlankZero3,  
Object4,x4,BlankZero4,Object5,x5,BlankZero5,Object6,x6,BlankZero6,  
Object7,x7,BlankZero7,Object8,x8,BlankZero8)  
{  
"BigGen Drop Lookup" "Lookup" "" ""  
"Interface 'Big Gen North'" "MW" "YES"  
"" "" "NO " "" "" "NO " "" "" "NO " "" "" "NO "  
"" "" "NO " "" "" "NO " "" "" "NO "  
<SUBDATA LookupTable>  
  x1      value  
    0.0    0.0  
 3000.0  -500.0  
 4000.0  -1200.0  
 5000.0  -2000.0  
</SUBDATA>  
}
```

## Model Expression

```
INJECTIONGROUP (Name)  
{  
"BigGens"  
}  
PARTPOINT (GroupName,Object,AutoCalcMethod,PartFact,AutoCalc)  
{  
"BigGens" "Gen '1228' '1'" "SPECIFIED" 1.00000 "NO "  
"BigGens" "Gen '1229' '1'" "SPECIFIED" 2.00000 "NO "  
"BigGens" "Gen '1230' '1'" "SPECIFIED" 3.00000 "NO "  
}
```

## Injection Group



# Details, Details, Details: Gen Tripping using Lookup Table



- RAS will refer to an Injection Group to do generation (or load) tripping
  - Tripping is done in the order of highest participation factor as assigned by PartPoints in injection group
- Trip based on **Lookup table**... Details matter
  - When should lookup calculation be done?
    - Often “gen drop arming levels” are based on system conditions before any event happens
    - Must base calculation on the **Reference State** as the interface flows may change during the contingency solution process
  - Tripping is done one unit at a time
    - Ordering is done using an injection group using “**Merit Order**”
  - Choice
    - **Do not EXCEED amount** of lookup table
    - Allow to EXCEED amount
- In this example it may be that the double line outage results in unsolvable power flow solution → **TOPOLOGYCHECK**

# BigGens to OvalOffice Gen Drop: RemedialAction



The screenshot shows the 'Contingency Element Dialog' with several annotations:

- Choose Injection Group** (red text): Points to the 'Injection Group' radio button in the 'Element Type' list and the 'BigGens' entry in the 'Choose the Element' list.
- Change by ModelExpression** (blue text): Points to the 'Change By' radio button in the 'Action Type' section and the 'Model Expression' dropdown in the 'Amount' section.
- Reference** (magenta text): Points to the 'Evaluate in Reference State' checkbox in the 'Amount' section.
- Open in Merit Order** (yellow text): Points to the 'Open in Merit Order' checkbox in the 'in' section.
- Open in Merit Order** (yellow text): Points to the 'Use Merit Order' checkbox in the 'in' section.
- Do Not Exceed Amount** (brown text): Points to the 'Do Not Exceed Amount' radio button in the 'in' section.
- CriteriaStatus = TOPOLOGYCHECK** (green text): Points to the 'Status' dropdown menu.
- Model Criteria points to ModelFilter** (purple text): Points to the 'Model Criteria' text field.

# BigGens to OvalOffice Gen Drop: ModelExpression, InjectionGroup



```
REMEDIALACTION (Name,Skip,Memo)
{
"BigGen OvalOffice Gen Drop" "NO " ""
}
```

```
REMEDIALACTIONELEMENT (RemedialAction,Object,Action,
Criteria,CriteriaStatus,TimeDelay,InclusionFilter,Comment)
{
"BigGen OvalOffice Gen Drop"
"INJECTIONGROUP 'BigGens'"
"CHANGEBY 'BigGen Drop Lookup' MWMERITORDEROPEN REF"
"Both BigGens to OvalOffice Opened" "TOPOLOGYCHECK" 0 "" ""
}
```

# TOPOLOGYCHECK

## Criteria based on *Status Only*



- Persistent problem in contingency runs when using RAS
  - RAS is configured to open 2,000 MW of generation when Line X trips
  - Before 2012, this was achieved by configuring a *POSTCHECK* action that looks at the branch status and trips generation if the branch is out of service
  - Works most of the time, but...
  - What if the outage of Line X results in an unsolvable steady state power flow solution?
    - Basically means that the RAS scheme is actually preventing a voltage collapse from occurring
    - This means that the *POSTCHECK* action is never evaluated because the power flow solution failed.
- Solution: *TOPOLOGYCHECK* actions

# Summary



- Communication
  - Develop relationships with folks in your company who maintain RAS and Relays
  - Communicate across utilities and WECC members
- Fundamental pieces of RAS are simple
  - But... the details of how they are put together are very particular
  - And... these details matter
- RAS file format structure is being maintained *incrementally* by PowerWorld
  - [http://www.powerworld.com/files/PowerWorld\\_RASFileFormat.pdf](http://www.powerworld.com/files/PowerWorld_RASFileFormat.pdf)