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?

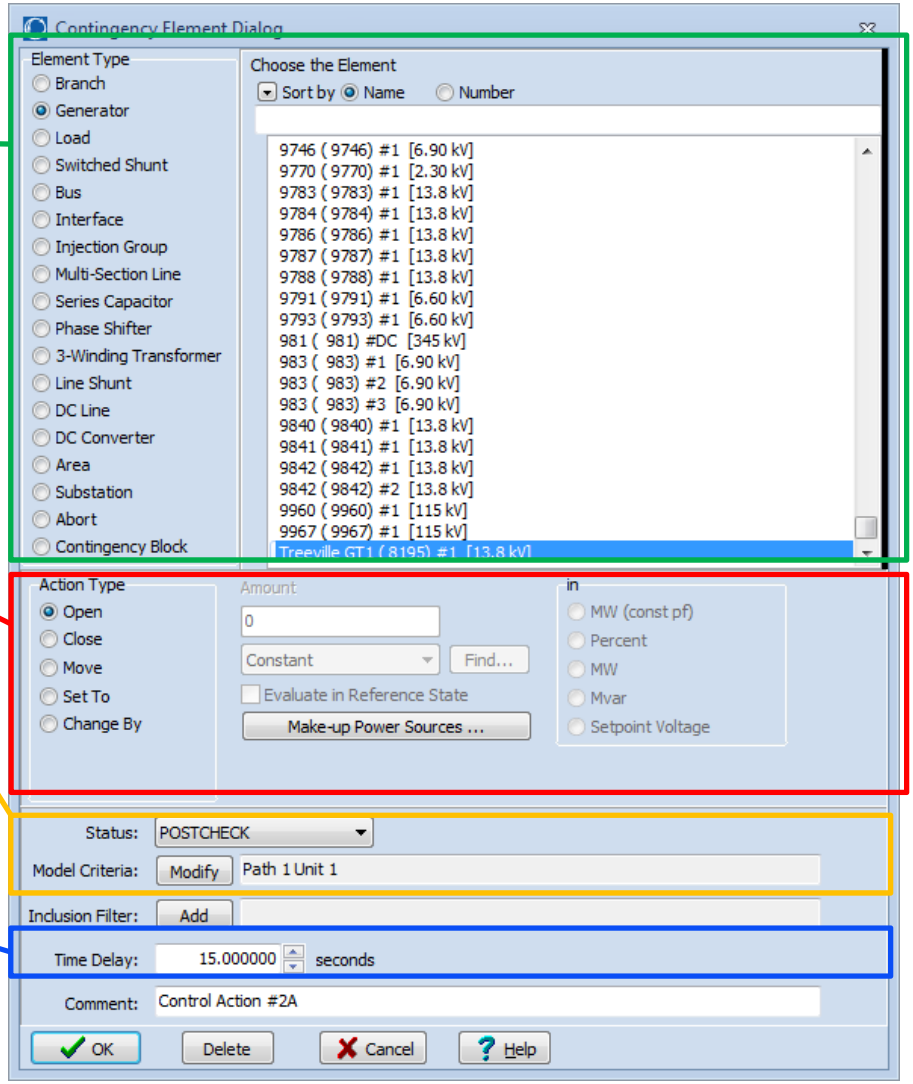
- Clearly *not* under normal operating conditions
  - When solving a base case power flow solution → RAS should not be doing anything
- Within a software tool, when are RAS and Relays going to matter?
  - **Contingency, Contingency, Contingency**
  - RAS will respond to the changes that occur during 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

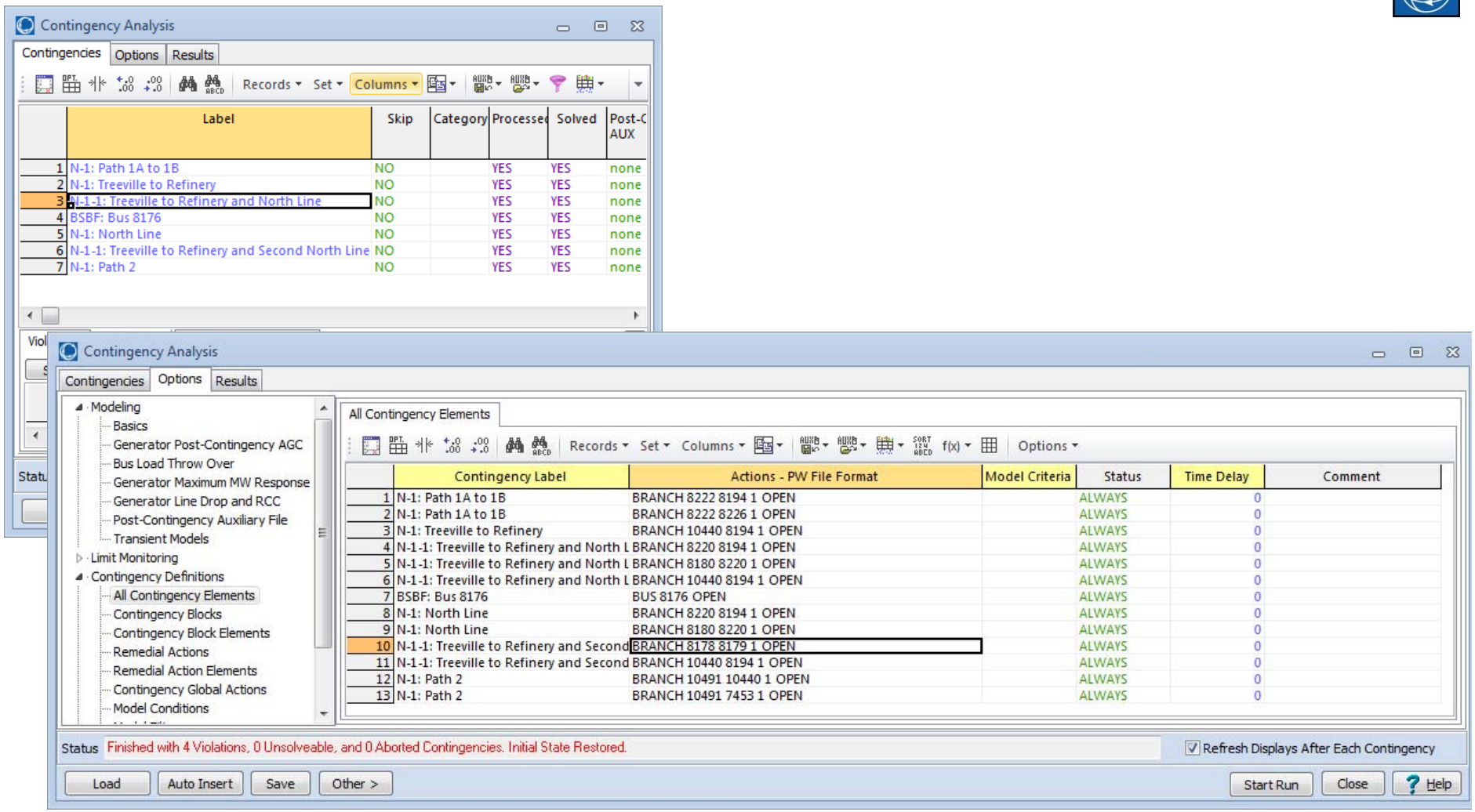
- *Object*
  - many choices of various contingency actions are available
- *Action*
  - What happens to the object
- *Criteria, CriteriaStatus*
  - Logical criteria under which actions are applied
- Time Delay
  - Use for ordering



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 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

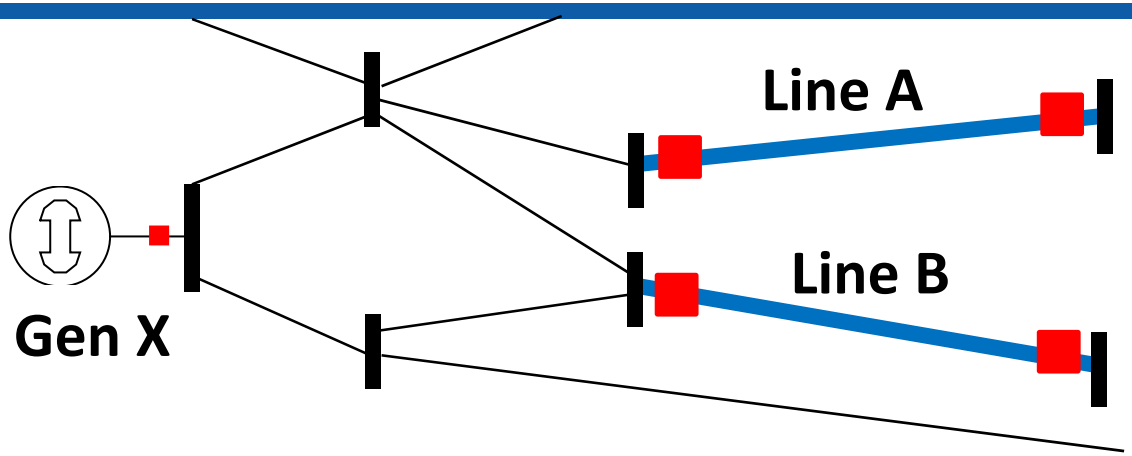
Problems with these approaches (2/2)

- 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

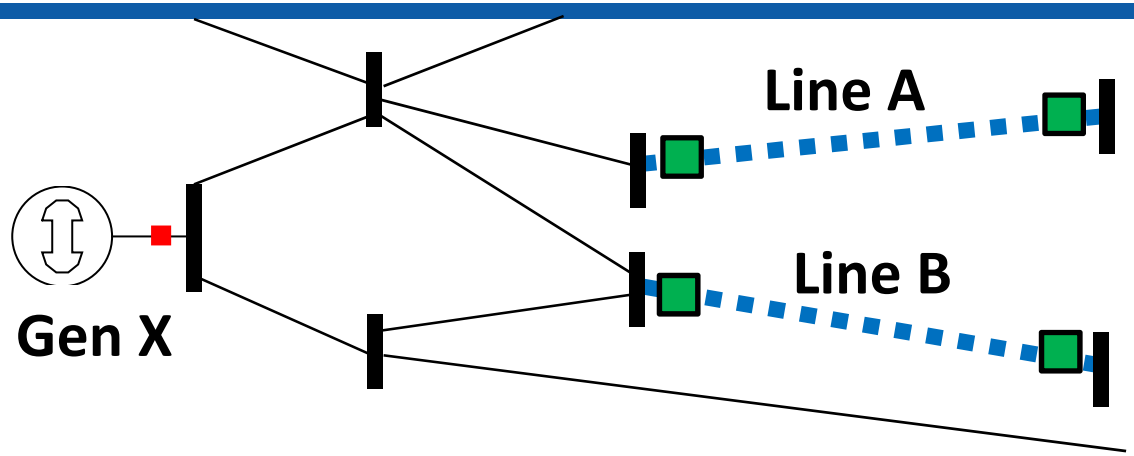
- PowerWorld's experience working with utility engineers
  - The engineers running power flow and transient stability studies have a general idea of how RAS functions
  - However, they may not manage and design the RAS itself, so they may miss details
- The implementation of *when* to "arm" and *how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

“Simple RAS”



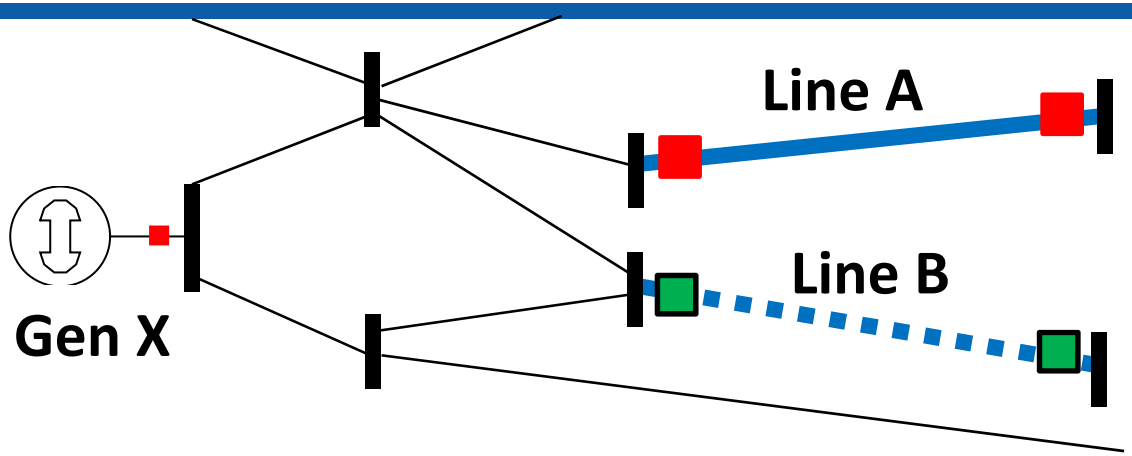
- General Description of RAS
  - If two transmission lines (**Line A** and **Line B**) are tripped → then trip a generator (**Gen X**)
- RemedialAction definition seems simple
  - *Object* = **Gen X**
  - *Action* = OPEN
  - *Criteria* = (**Line A** is OPEN) AND (**Line B** is OPEN)

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 their 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

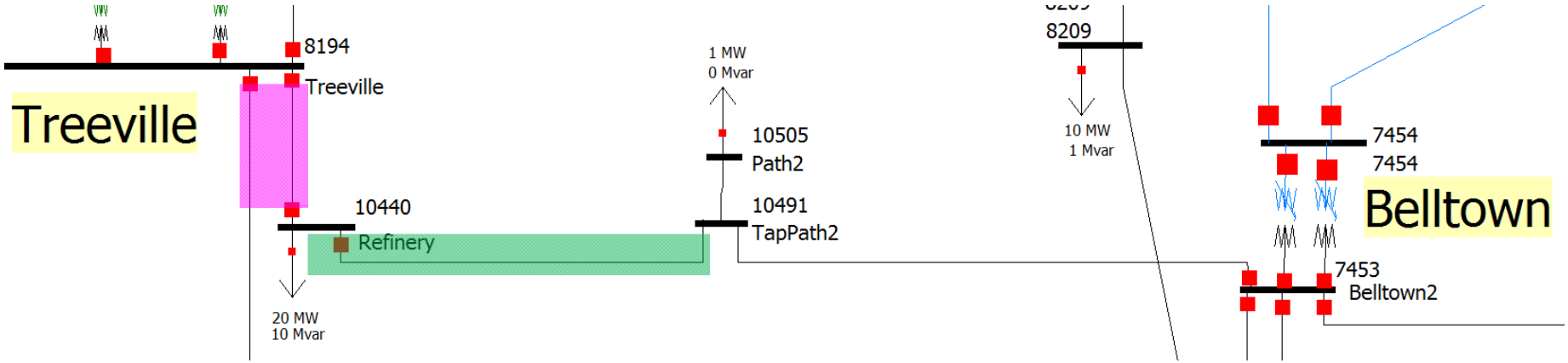
- 4 Separate Control Actions Possible
  1. Ramp Treeville generation down by 40 MW in two minutes
  2. Trip one Treeville gas combustion turbine
  3. Trip the all the Treeville generators
  4. Trip path of the 115 kV line going to the Refinery

- 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 ⟶ Trip the Treeville to Refinery Line

- If Green Line Overloads then Trip Pink Line

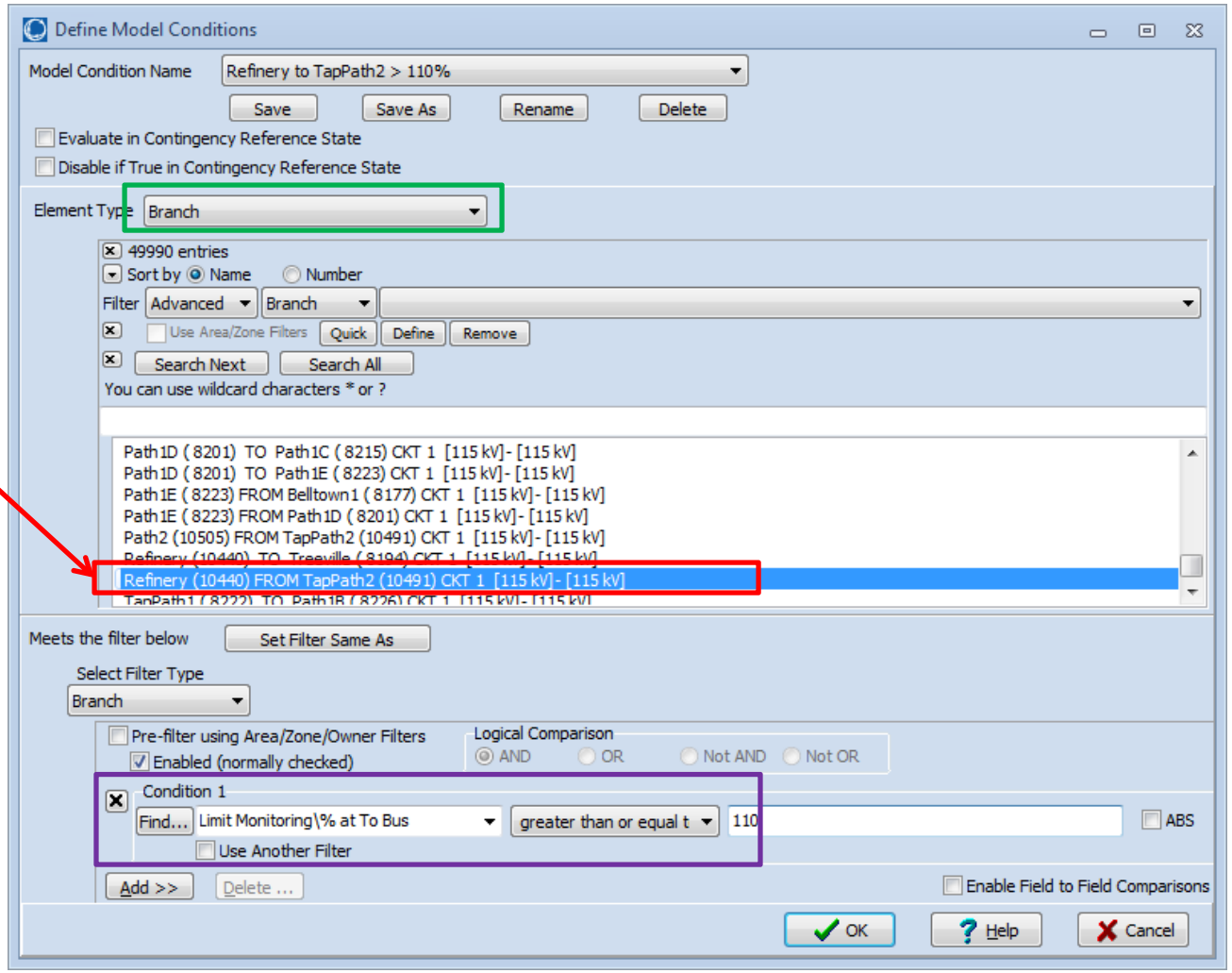


Treeville RAS Control Action #4: ModelCondition Dialog

Type = Branch

Choose Branch

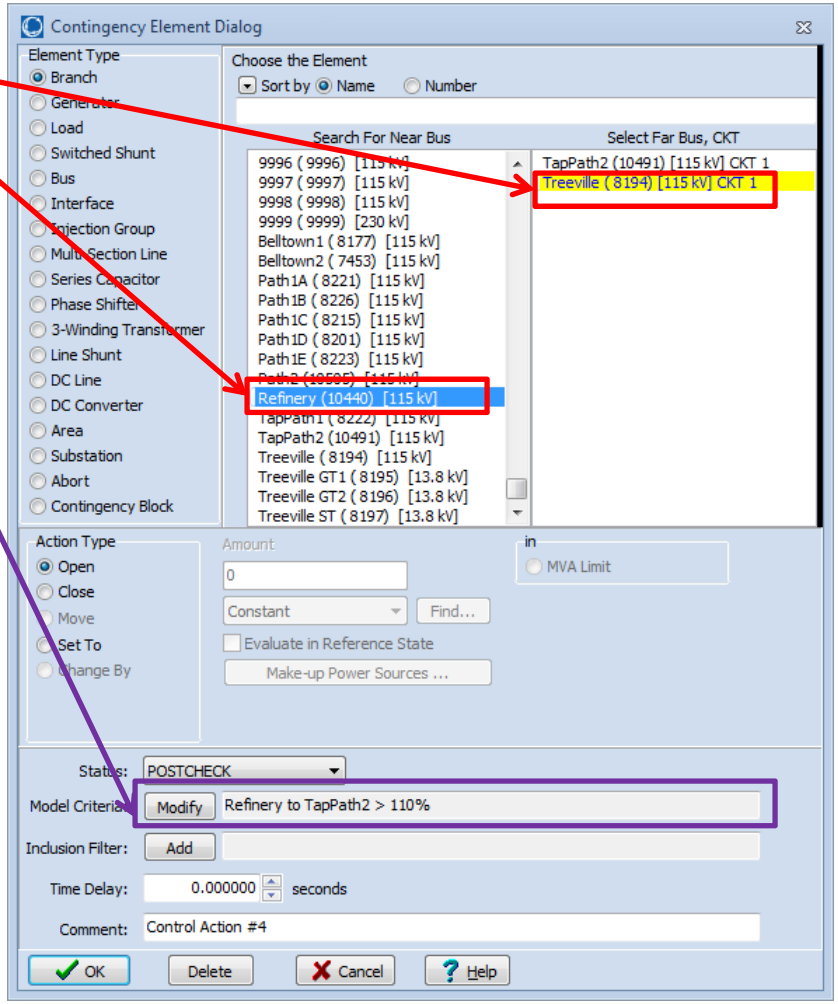
Condition for Branch



Treeville RAS Control Action #4: RemedialAction

Choose Branch to Open

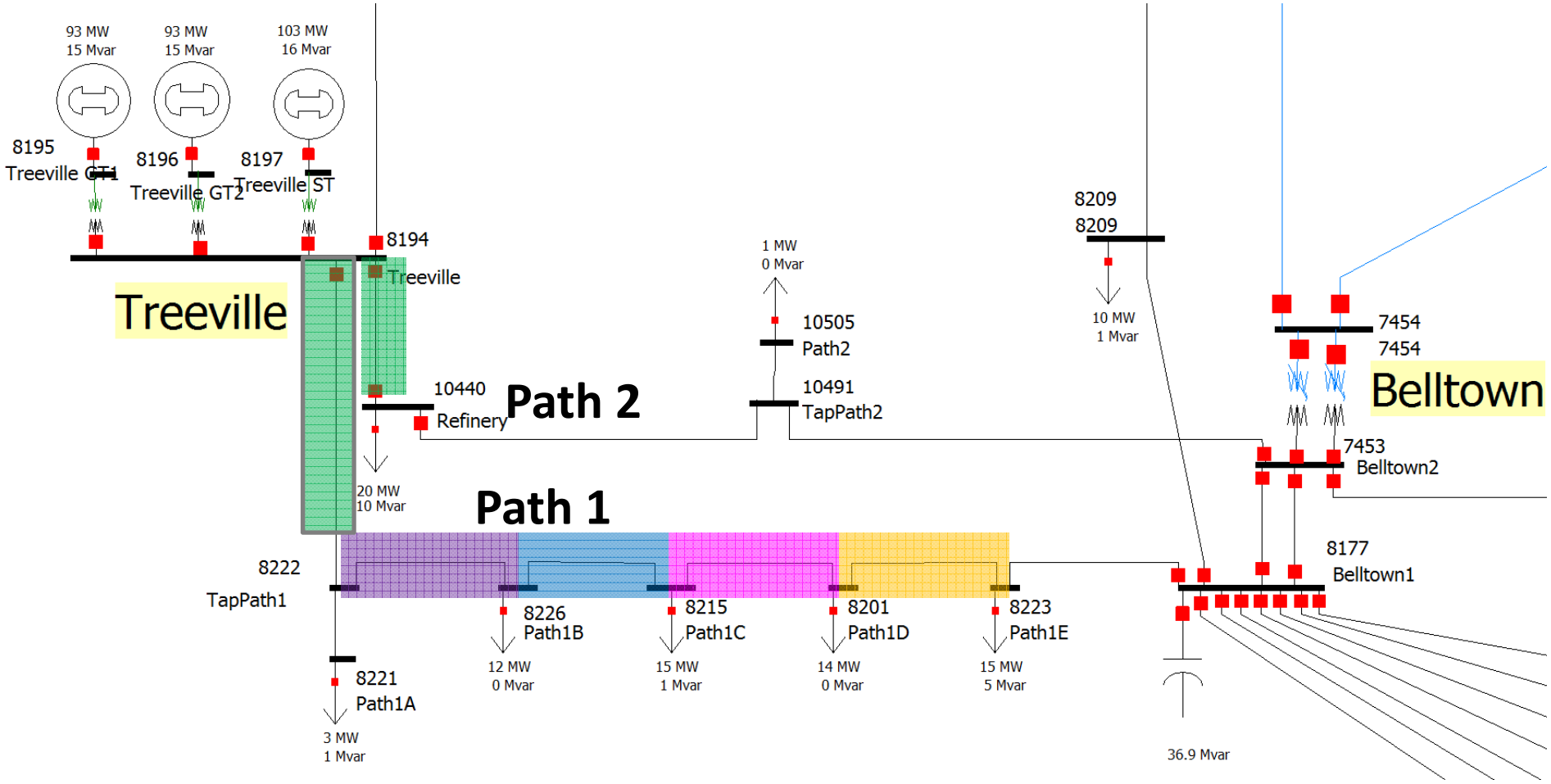
Model Criteria points to ModelCondition



Treeville RAS Control Action #4: File Format

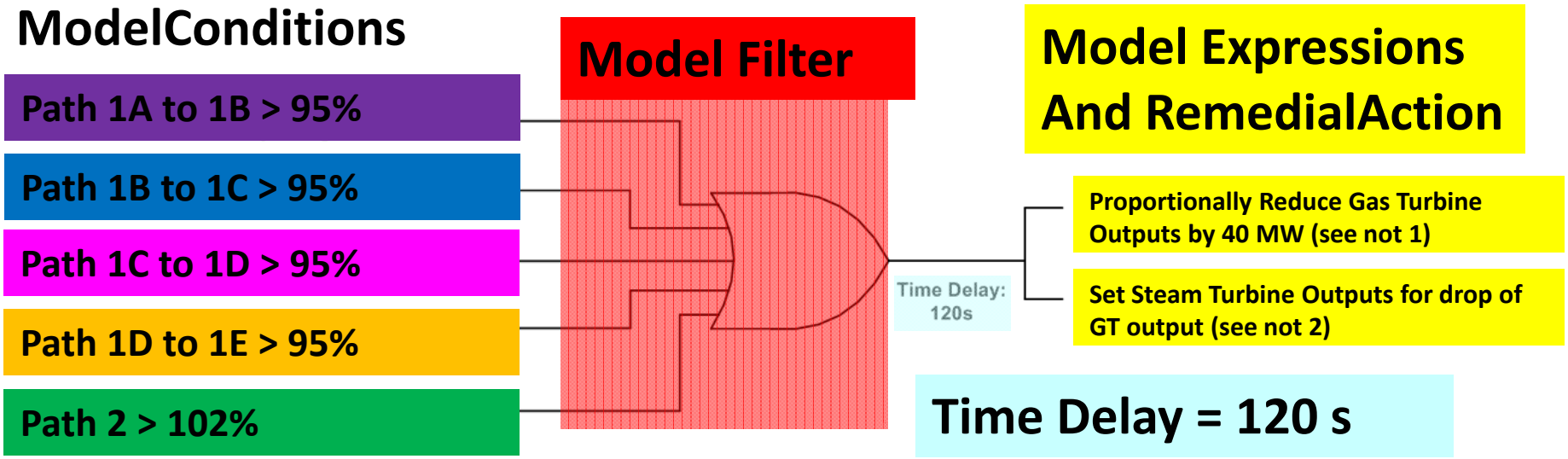
```
MODELCONDITION (Name,Object,FilterObjectType,FilterLogic,
                EvaluateInRef,DisableIfTrueInRef,Memo)
{
"Refinery Tap2 > 110%" "Branch '10491' '10440' '1'" "Branch" "AND" "NO" "NO " ""
}
MODELCONDITIONCONDITION (ModelCondition,CondNum,ObjectField,
                         ConditionType,Value,OtherValue,Absolute)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



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.

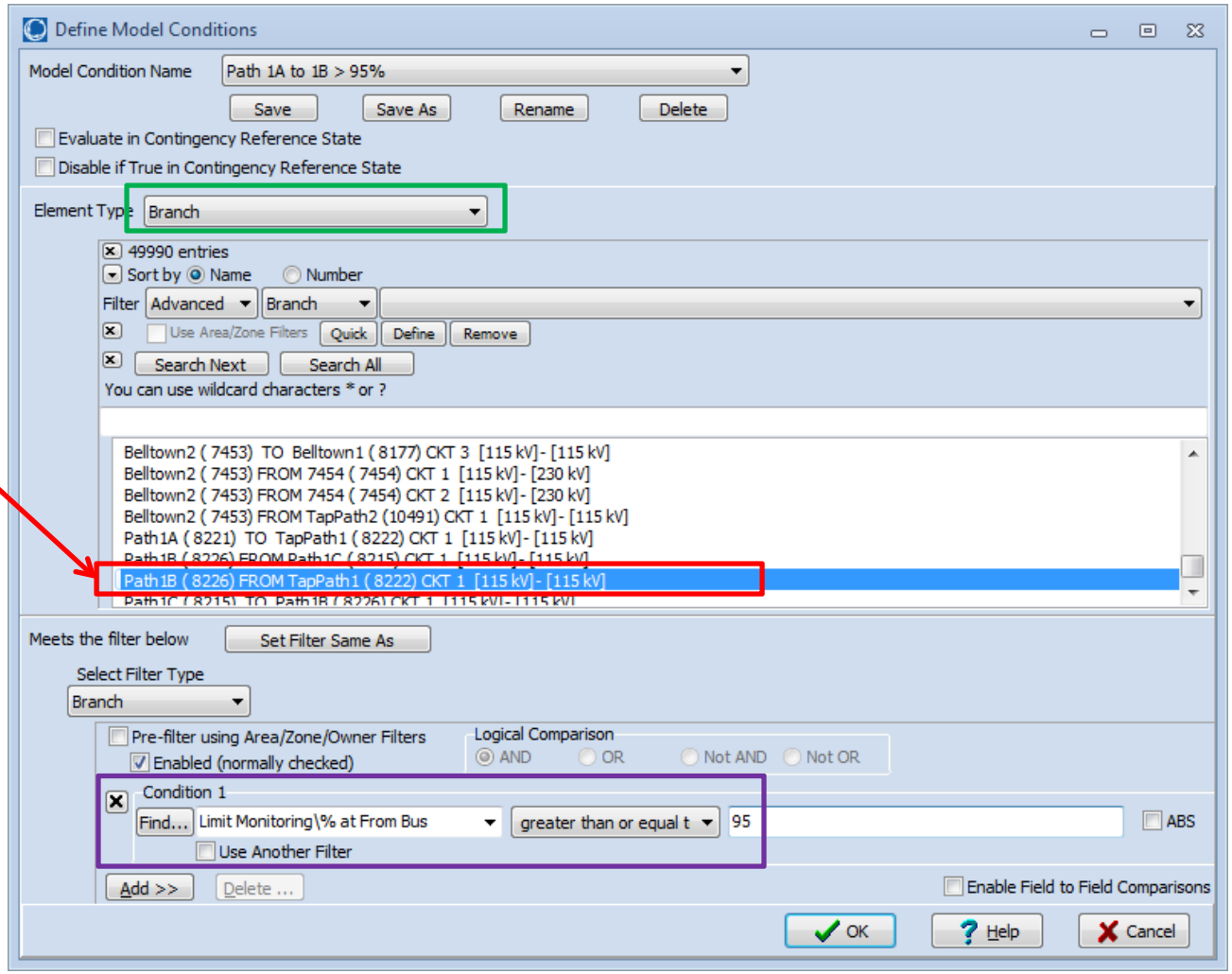
2. ST model expression is x5*((x1+x2-40)/(x3+x4)) where x1=actual MW output of unit 1, x2=actual MW output of unit 2, x3=Pmax of unit 1, x4=Pmax of unit 2 and x5=Pmax of steam unit.

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

Type = Branch

Choose Branch

Condition for Branch

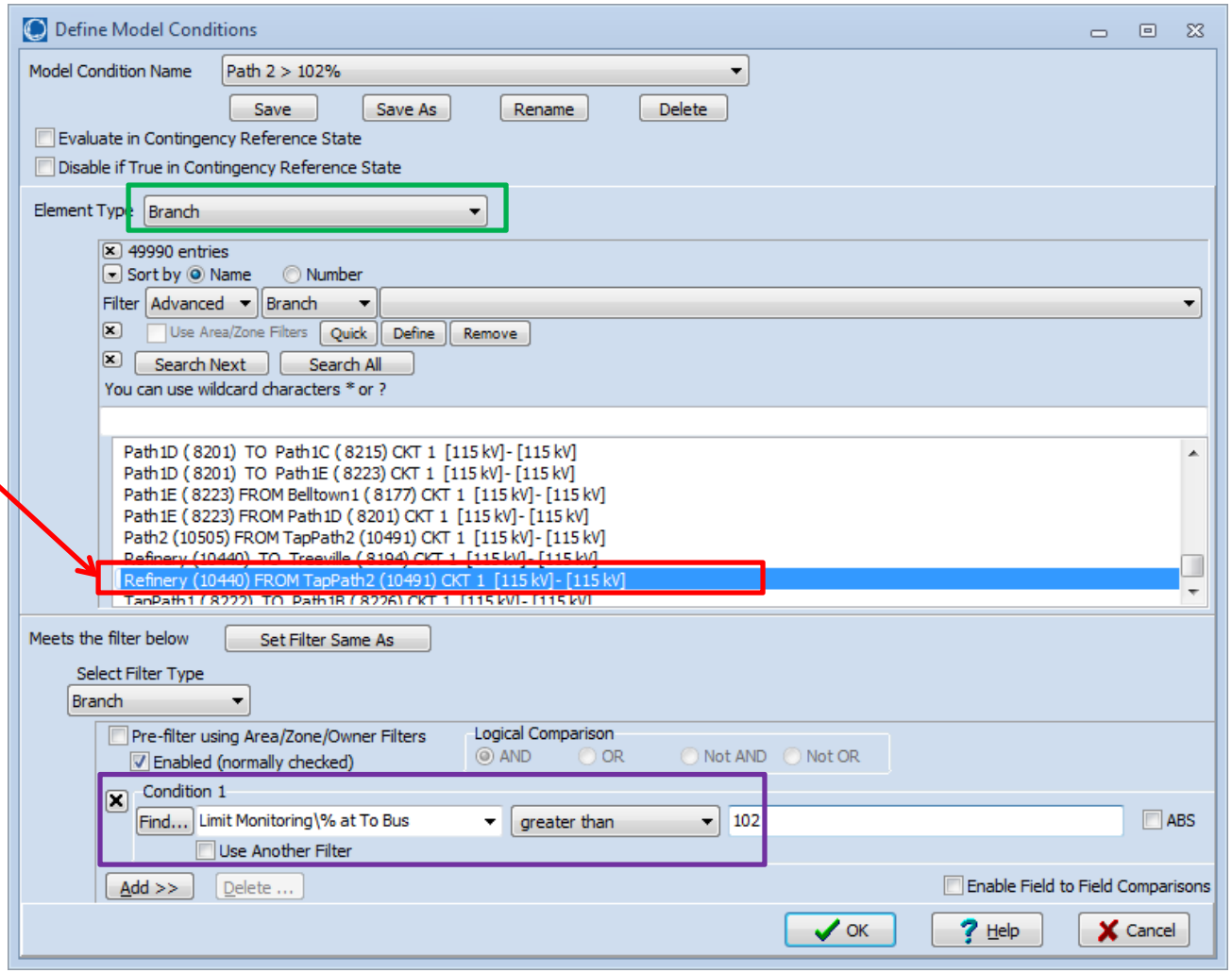


Define Model Conditions: Path 2 > 102%

Type = Branch

Choose Branch

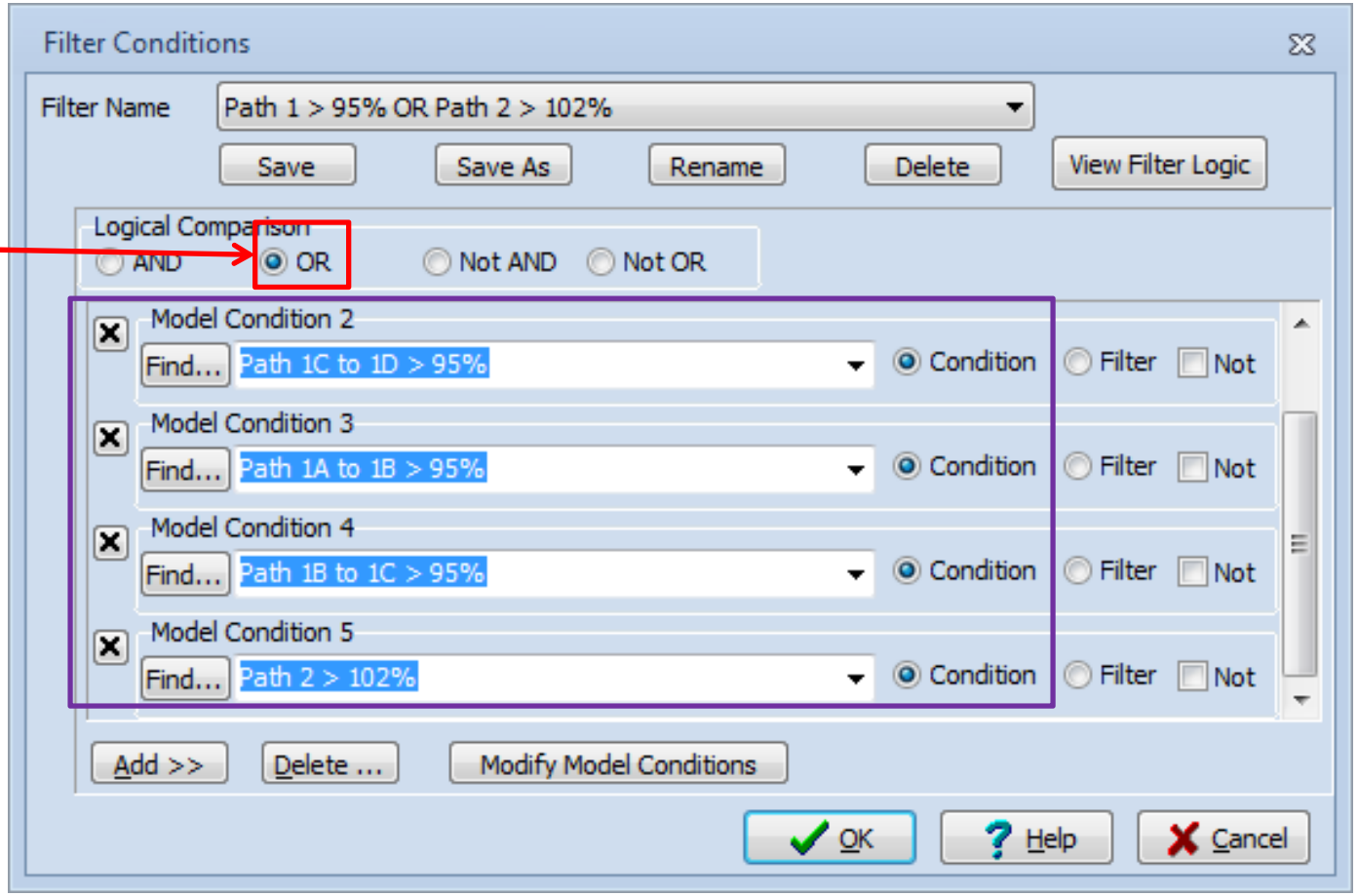
Condition for Branch



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

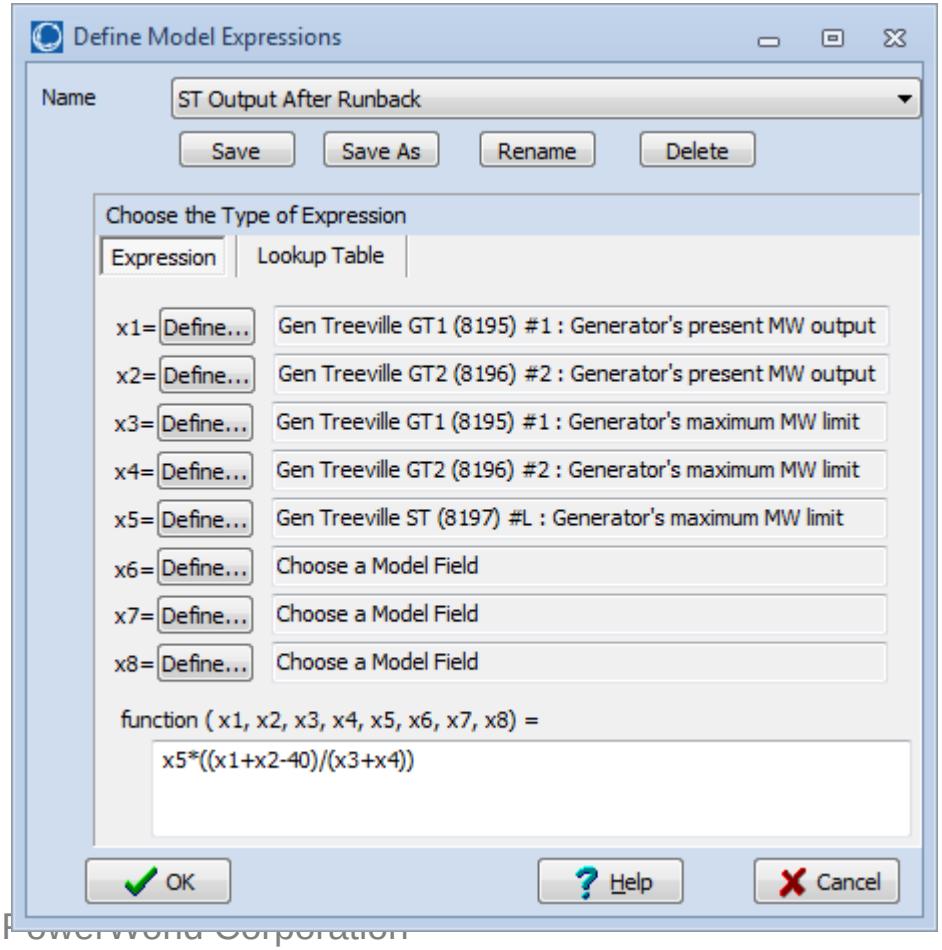
List of Model Conditions

Treeville RAS Control Action #1: ModelFilter File Format

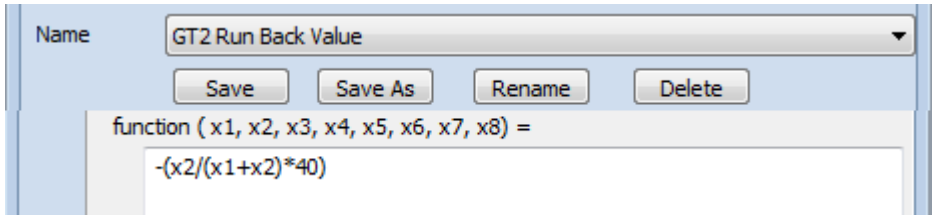
```
MODELFILTER (Name,Logic,Memo)
{
"Path 1 > 95% OR Path 2 > 102%" "OR" ""
}
MODELFILTERCONDITION (ModelFilter,CondNum,Criteria,Logic)
{
"Path 1 > 95% OR Path 2 > 102%" 1 "Path 1A to 1B > 95%" ""
"Path 1 > 95% OR Path 2 > 102%" 2 "Path 1B to 1C > 95%" ""
"Path 1 > 95% OR Path 2 > 102%" 3 "Path 1C to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

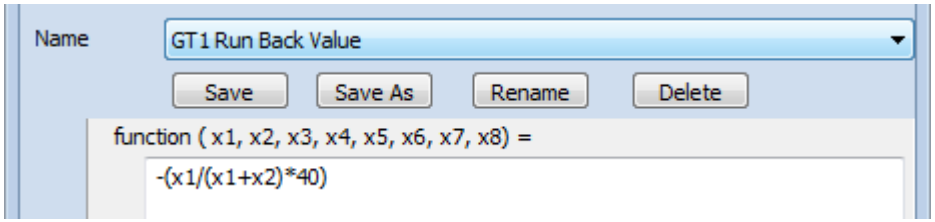
Treeville RAS Control Action #1: Model Expression

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)





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 "

"GT2 Run Back Value"       "Expression" "-(x2/(x1+x2)*40)" ""
               "Gen '8195' '1'" "MW" "YES" "Gen '8196' '2'" "MW" "YES"
               "" "" "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 "
}
```

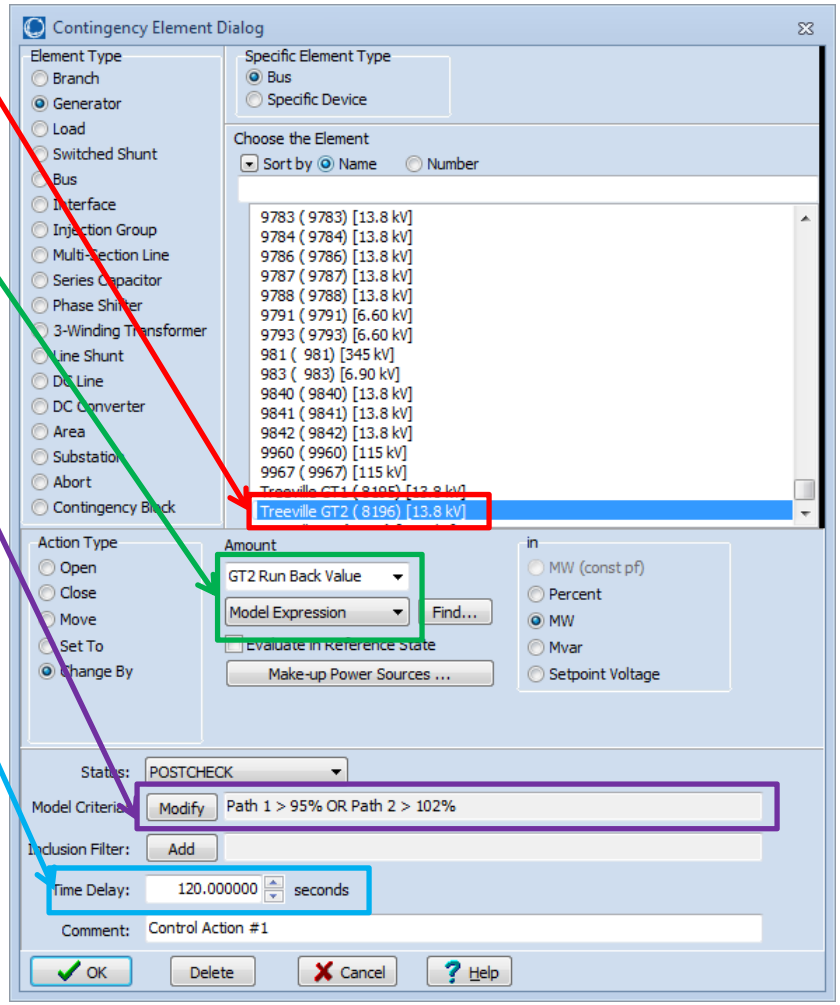
Treeville RAS Control Action #1: RemedialAction Gas Units

Choose Generator Unit 2

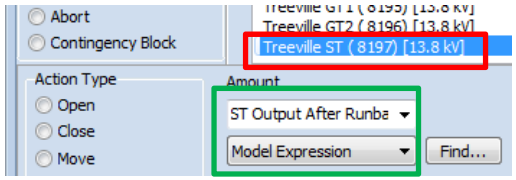
Model Expression for Unit 2

Model Criteria points to ModelFilter

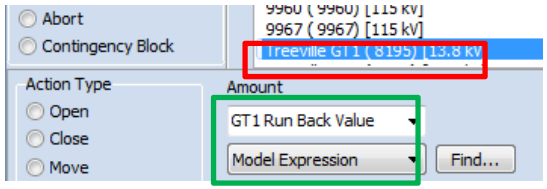
Time Delay



Same but for Steam Unit



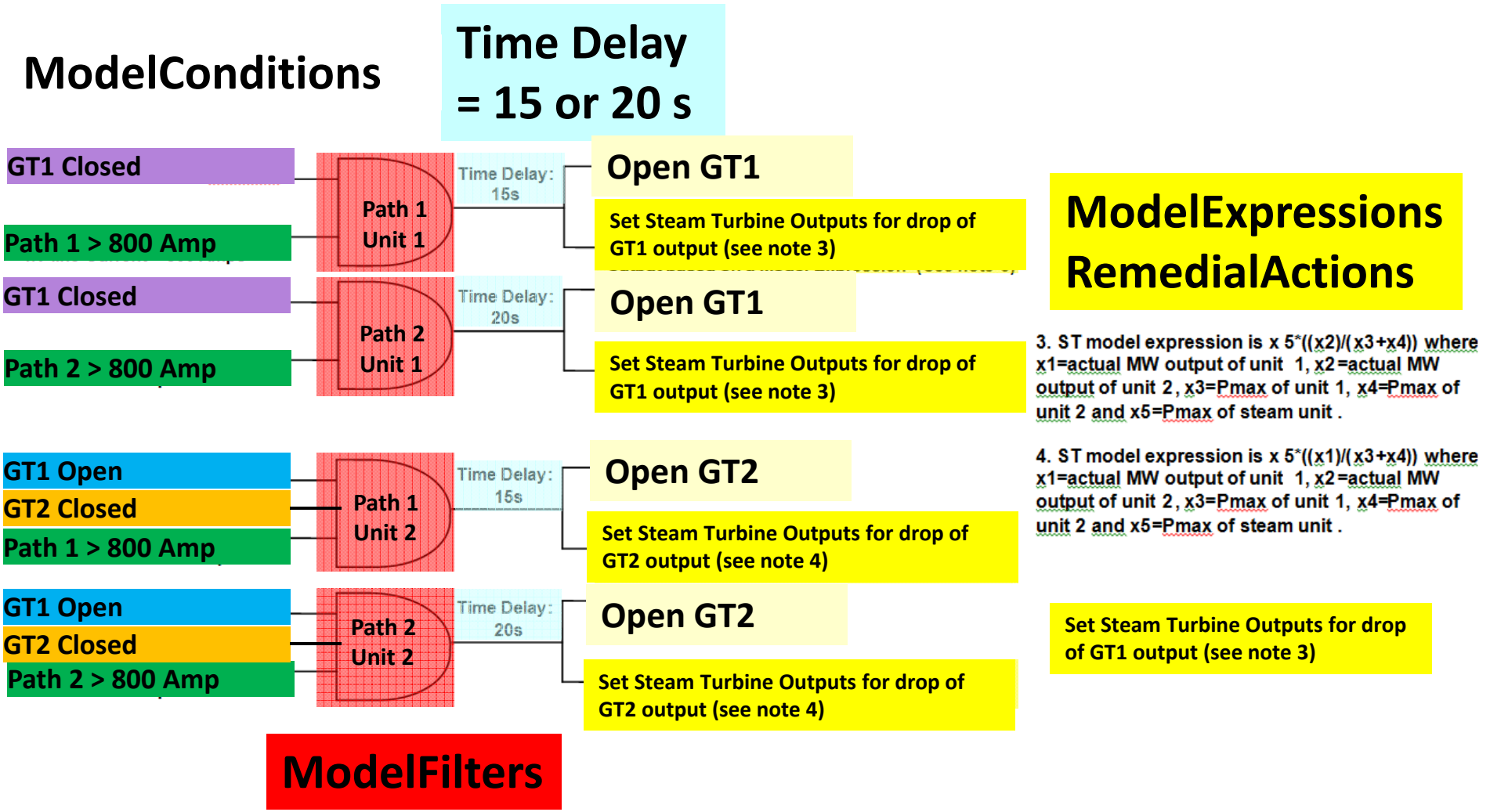
Same but for Unit 1



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



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 "
}
```

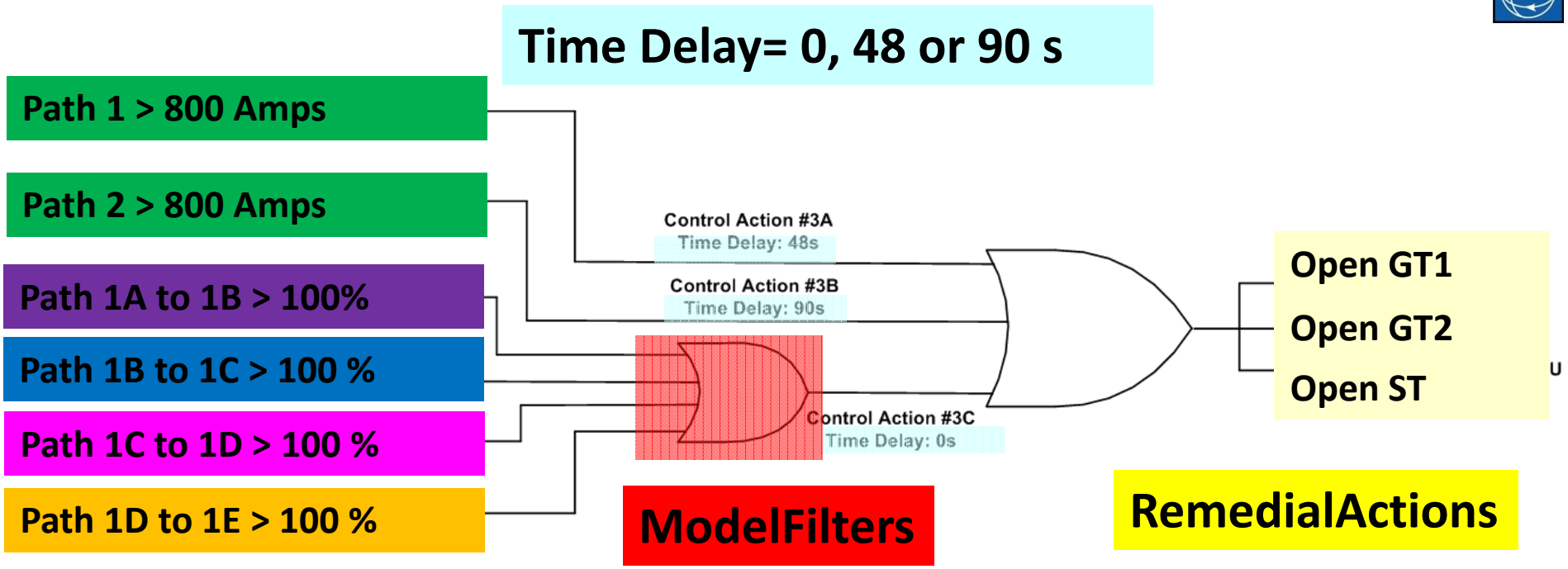
```
MODELFILTER (Name,Logic,Memo)
{
"Path 1 Unit 1" "AND" ""

"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" ""
}
```

Treeville 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)
{
"Treeville Generation Run-Back Scheme" "NO " ""
}
REMEDIALACTIONELEMENT (RemedialAction,Object,Action,Criteria,CriteriaStatus,TimeDelay,InclusionFilter,
   Comment)
{
"Treeville Generation Run-Back Scheme" "GEN 8195 1" "OPEN"
            "Path 1 Unit 1" "POSTCHECK" 15.000000 "" "Control Action #2A"
"Treeville Generation Run-Back Scheme" "GEN 8196 2" "OPEN"
            "Path 1 Unit 1" "POSTCHECK" 15.000000 "" "Control Action #2A"
"Treeville Generation Run-Back Scheme" "GEN 8195 1" "OPEN"
            "Path 2 Unit 1" "POSTCHECK" 20.000000 "" "Control Action #2B"
"Treeville Generation Run-Back Scheme" "GEN 8196 2" "OPEN"
            "Path 2 Unit 1" "POSTCHECK" 20.000000 "" "Control Action #2B“
"Treeville Generation Run-Back Scheme" "GEN 8197"   "SETTO 'ST Output for GT1 Outage' MW"
            "Path 1 Unit 2" "POSTCHECK" 15.000000 "" "Control Action #2A"
"Treeville Generation Run-Back Scheme" "GEN 8197"   "SETTO 'ST Output for GT2 Outage' MW"
            "Path 1 Unit 2" "POSTCHECK" 15.000000 "" "Control Action #2A"
"Treeville Generation Run-Back Scheme" "GEN 8197"   "SETTO 'ST Output for GT1 Outage' MW"
            "Path 2 Unit 2" "POSTCHECK" 20.000000 "" "Control Action #2B"
"Treeville 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



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

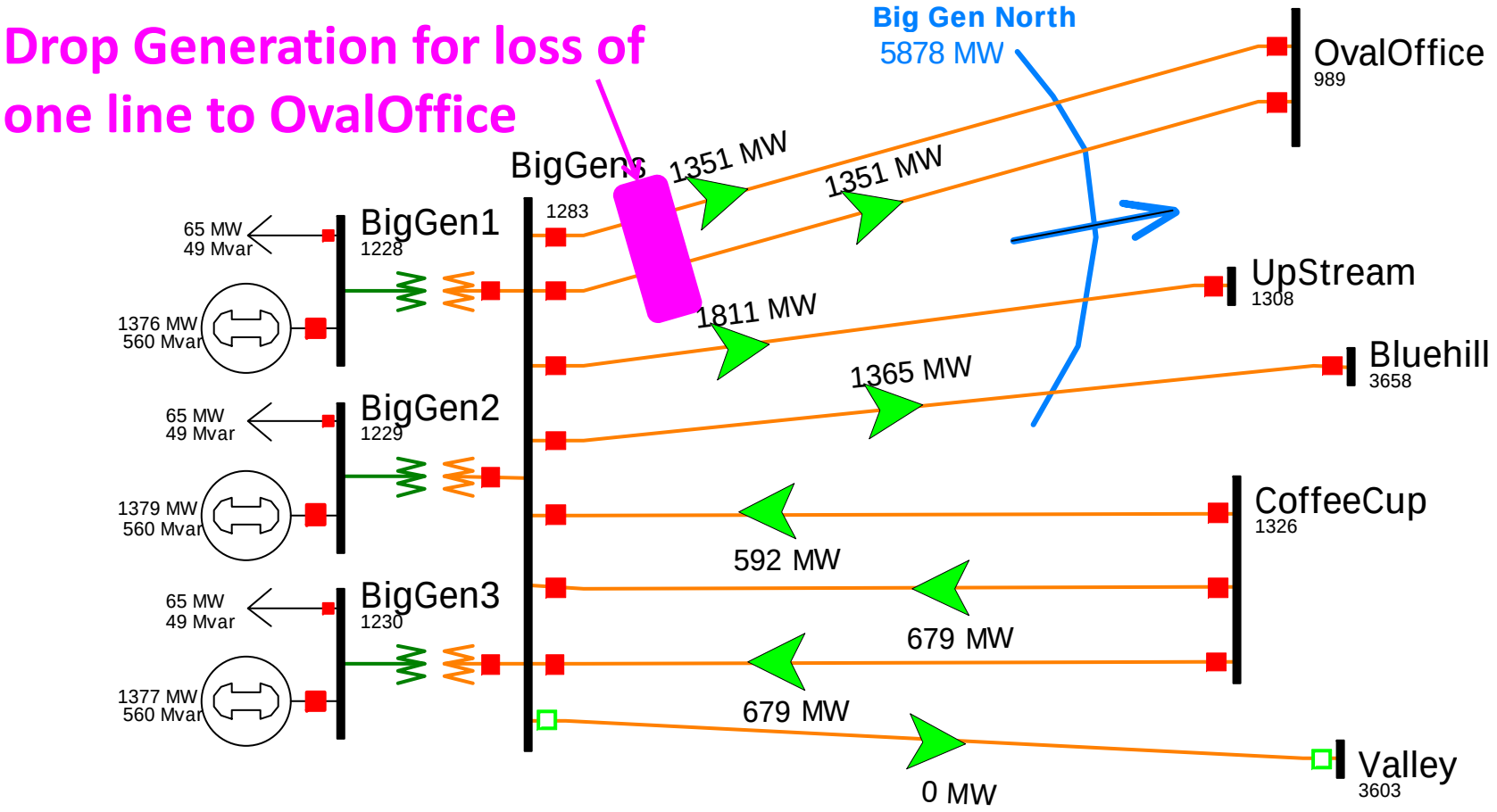
```
REMEDIALACTION (Name,Skip,Memo)
{
"Treeville Generation Run-Back Scheme" "NO " ""
}
REMEDIALACTIONELEMENT (RemedialAction,Object,Action,
          Criteria,CriteriaStatus,TimeDelay,InclusionFilter,Comment)
{
// Trip GT1
"Treeville Generation Run-Back Scheme" "BRANCH 8195 8194 1" "OPEN" "Path 1 > 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!

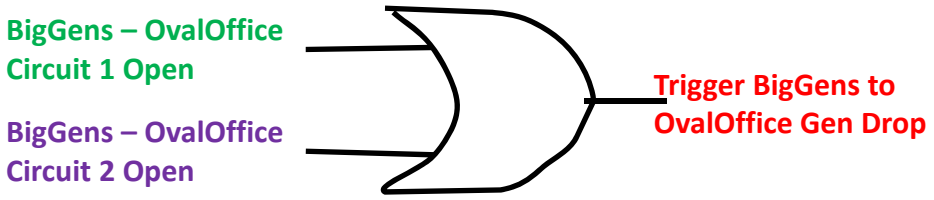
Example #2: Double Line Outage with Generation Dropping

- Consider system below



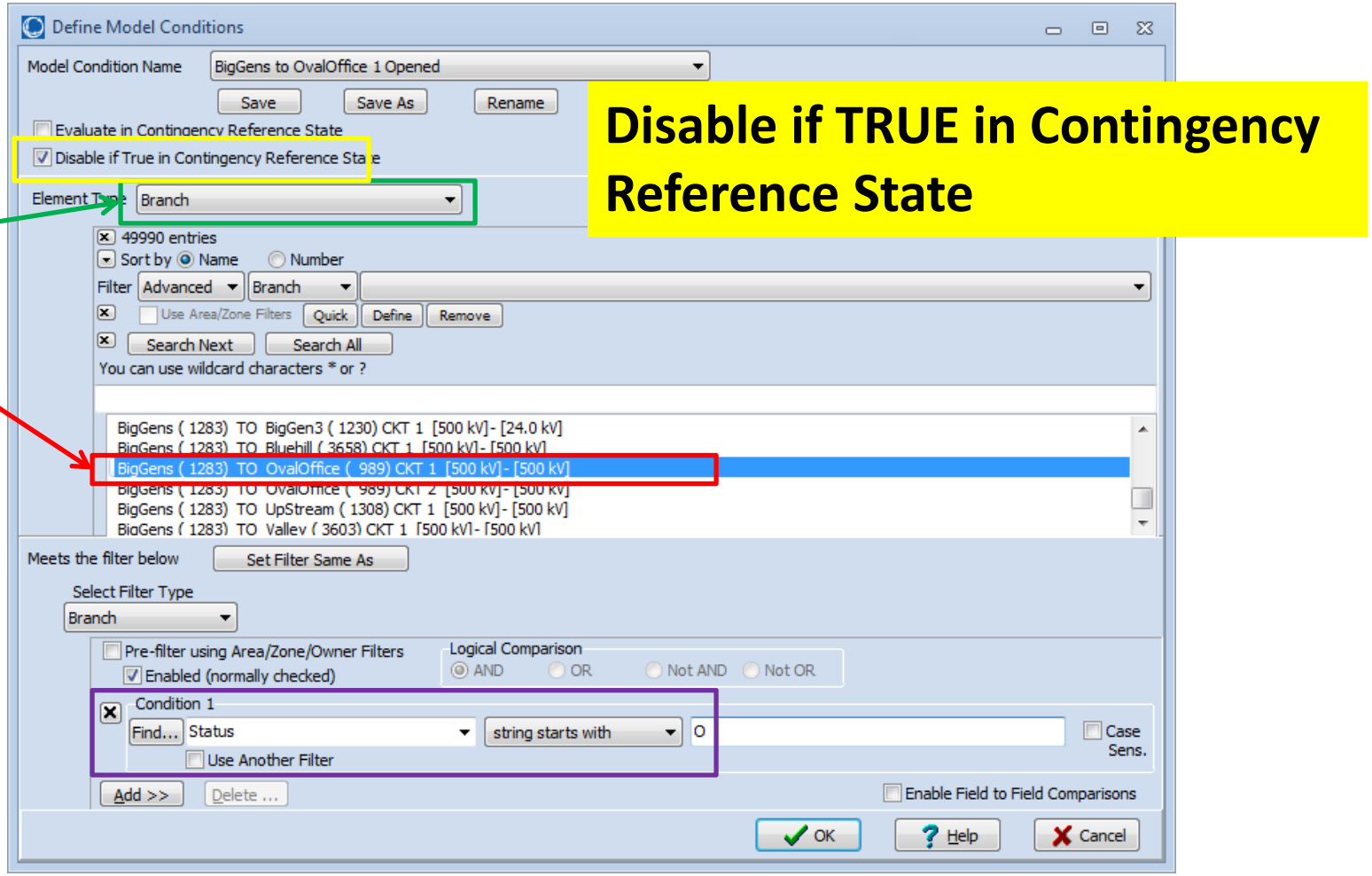
BigGens to OvalOffice Gen Drop

- 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"



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

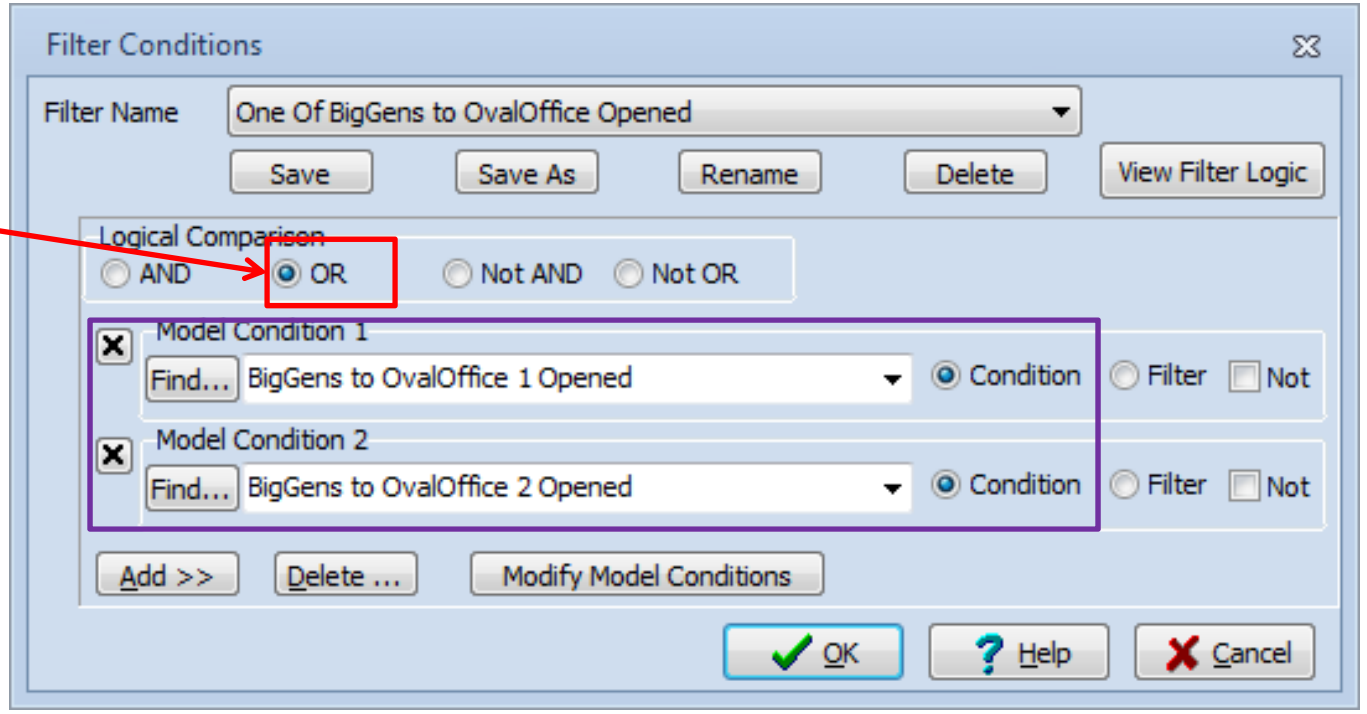
BigGens to OvalOffice Gen Drop: ModelCondition Dialog



BigGens to OvalOffice Gen Drop: ModelFilter Dialog

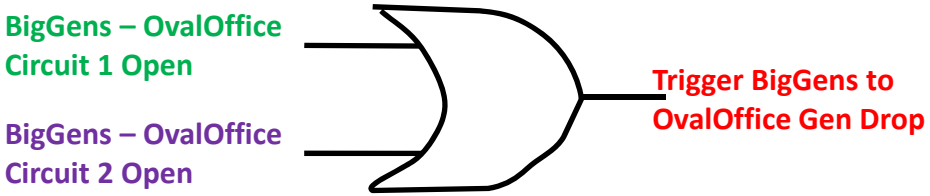
OR Logic

List of Model Conditions

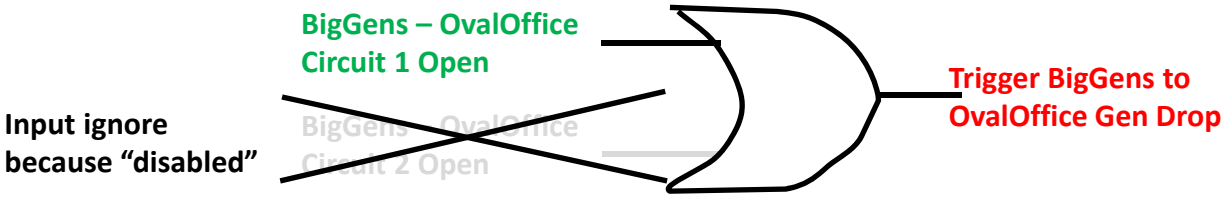


Disable if TRUE in Contingency Reference State Meaning

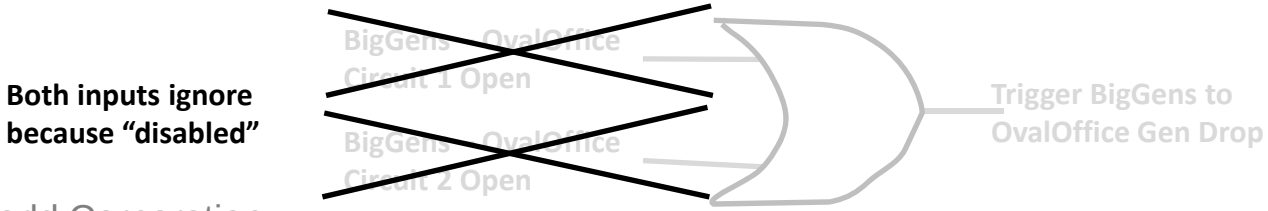
- If both lines are inservice then logic looks like



- 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



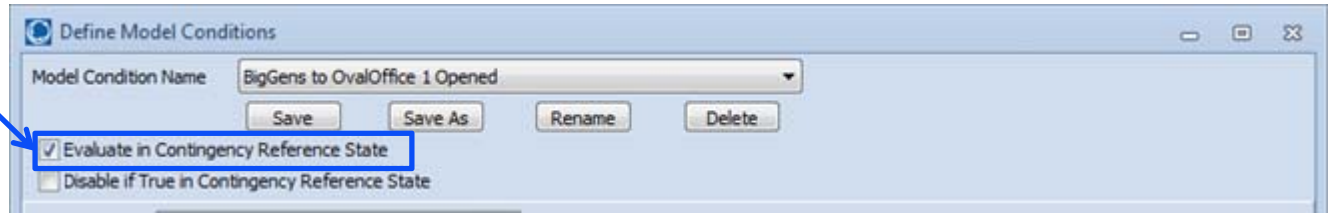
BigGens to OvalOffice Gen Drop: ModelCondition, ModelFilter File

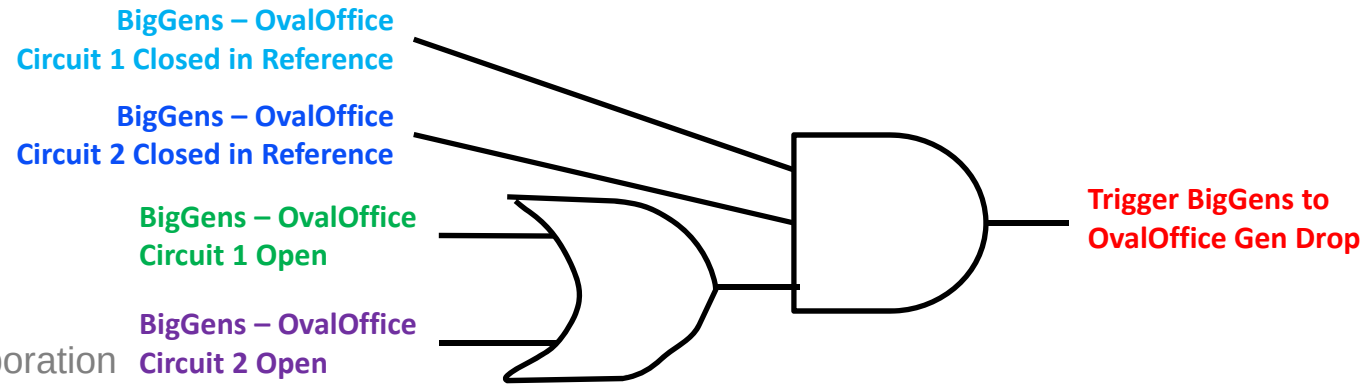
```
MODELCONDITION (Name,Object,FilterObjectType,FilterLogic,EvaluateInRef,DisableIfTrueInRef,Memo)
{
"BigGens to OvalOffice 1 Opened" "Branch '1283' '989' '1'" "Branch" "AND" "NO" "YES" ""
"BigGens to OvalOffice 2 Opened" "Branch '1283' '989' '2'" "Branch" "AND" "NO" "YES" ""
}
MODELCONDITIONCONDITION (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

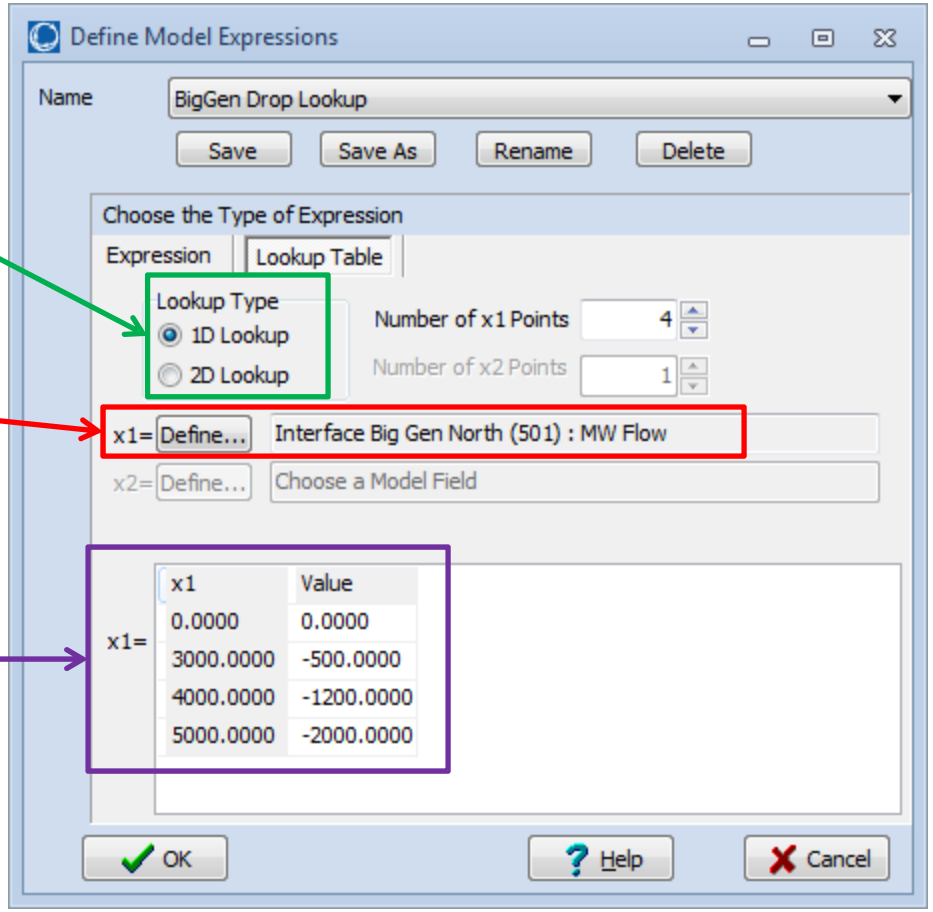
- 1D Lookup Table using a ModelExpression

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

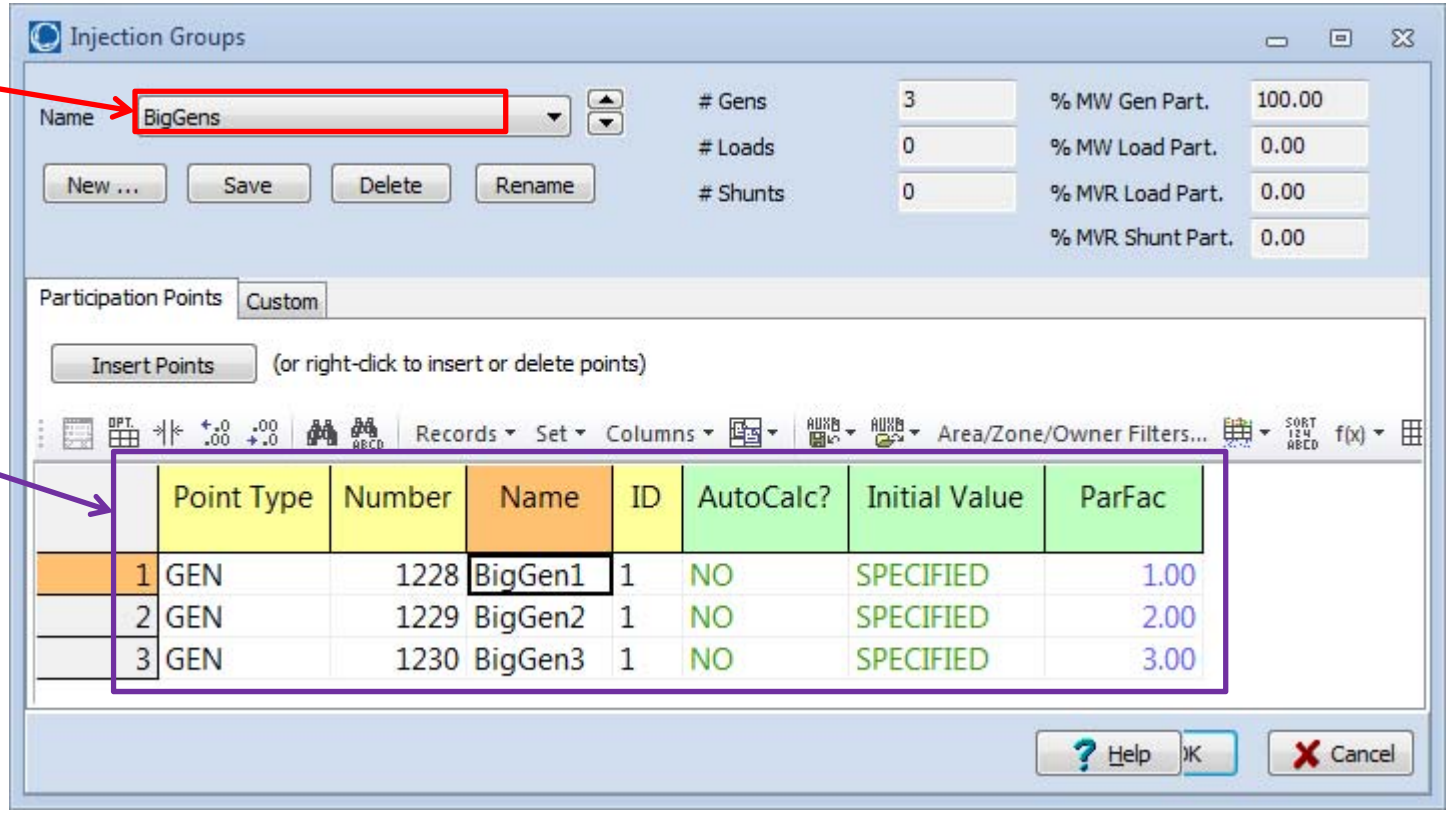


Injection Group Load Dropping

- First you must define an Injection Group

Name

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

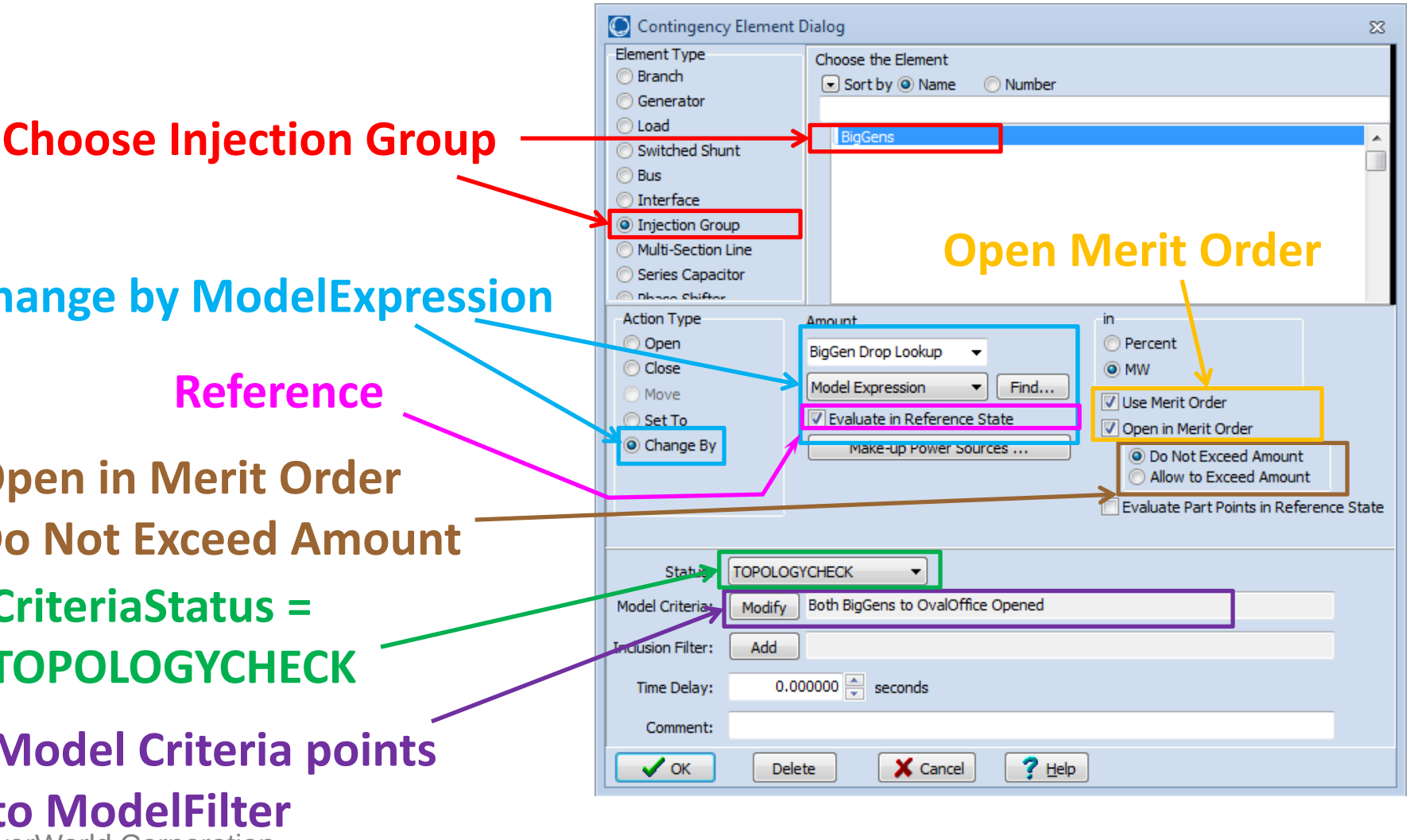
```
INJECTIONGROUP (Name)
{
"BigGens"
}
PARTPOINT (GroupName,Object,AutoCalcMethod,PartFact,AutoCalc)
{
"BigGens" "Gen '1228' '1'" "SPECIFIED" 1.00000 "NO "
"BigGens" "Gen '1229' '1'" "SPECIFIED" 2.00000 "NO "
"BigGens" "Gen '1230' '1'" "SPECIFIED" 3.00000 "NO "
}
```

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



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