# Simulator Data Structures and Auxiliary Files

2001 South First Street
Champaign, Illinois 61820
+1 (217) 384.6330

support@powerworld.com
http://www.powerworld.com

# Overview

- This section uses a set of scripts that automate case development and sensitivity analysis (TLR or shift factors) to introduce data structures and scripting language in PowerWorld Simulator
- References
  - Auxiliary File Format (pdf document)
  - Simulator's Model Explorer and Case Information Displays
  - Export Case Object Fields…

# Auxiliary Files

- PowerWorld Simulator DATA can be stored and edited in a text file format
- A scripting language is available for modifying data and automatically running PowerWorld Simulator commands
- The auxiliary (*.aux) file format accomplishes both functions

# Auxiliary File Format Overview

- Has two types of "Sections"
  - There is no limit to the number of sections in a file

SCRIPT Section

```
SCRIPT ScriptName1
{
script_statement_1;
    :
script_statement_n;
}
```

DATA Section

```
object_type DataName1(list_of_fields)
{
data_list_1
    :
data_list_n
}
```

DATA Section

```
object_type DataName2(list_of_fields)
{
data_list_1
    :
data_list_n
}
```

SCRIPT Section

```
SCRIPT ScriptName2
{
script_statement_1;
    :
script_statement_n;
}
```

Section header

Curly braces to open and close sections

Semicolon after each SCRIPT statement

# Auxiliary Files

- Auxiliary Script Files (*.aux) may be used to standardize settings or automate batch processes
- No looping or flow control (e.g. if…then...else; do…while; for…next)
  - minor exceptions with *SolvePowerFlow* and related functions
  - this is where external programming environments (e.g. Python, Visual Basic, C++, Matlab, etc.) and SimAuto add value
  - SimAuto can also facilitate exchange with external databases (*GetParameters* and *ChangeParameters* families of functions)
- SimAuto applications can
  - Load Aux files (*ProcessAuxFile* function)
  - Run individual script statements (*RunScriptCommand* function)

# What Can You DO with Auxiliary Files?

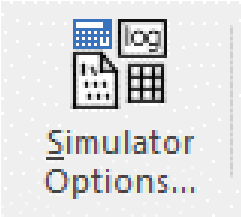
- Quality Assurance: Standardize settings and controls for multiple cases and studies
- Customize Simulator environment
- Document
  - Describe an analysis procedure for a manager or client
  - Create a detailed project record
  - Enable reproducibility
- Automate detailed calculations and storage of the results
- Automate building and editing of a one-line diagram

# Aux Automation Example

# Aux Automation Example

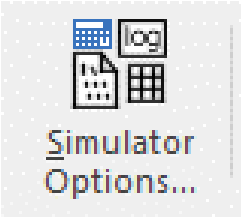
- Open ..*Case Development Aux\ACTIVSg10k.raw*
- Series of files with names *aux20?0*.aux*
  - Set Solution Options (*aux2010*.aux*)
  - Load Filters, Expressions, and Calculated Fields that are used in subsequent processes (*aux2020*.aux*)
  - Set Limit Monitoring (*aux2030*.aux*)
  - Auto-insert Contingencies (*aux2050*.aux*)
  - Load Generator information (*aux2060*.aux*)
  - Set Generator, Area, and Super Area and AGC (*aux2070*.aux*)
  - Perform sensitivity calculations and store results (*aux2080*.aux*)

# Solution Options

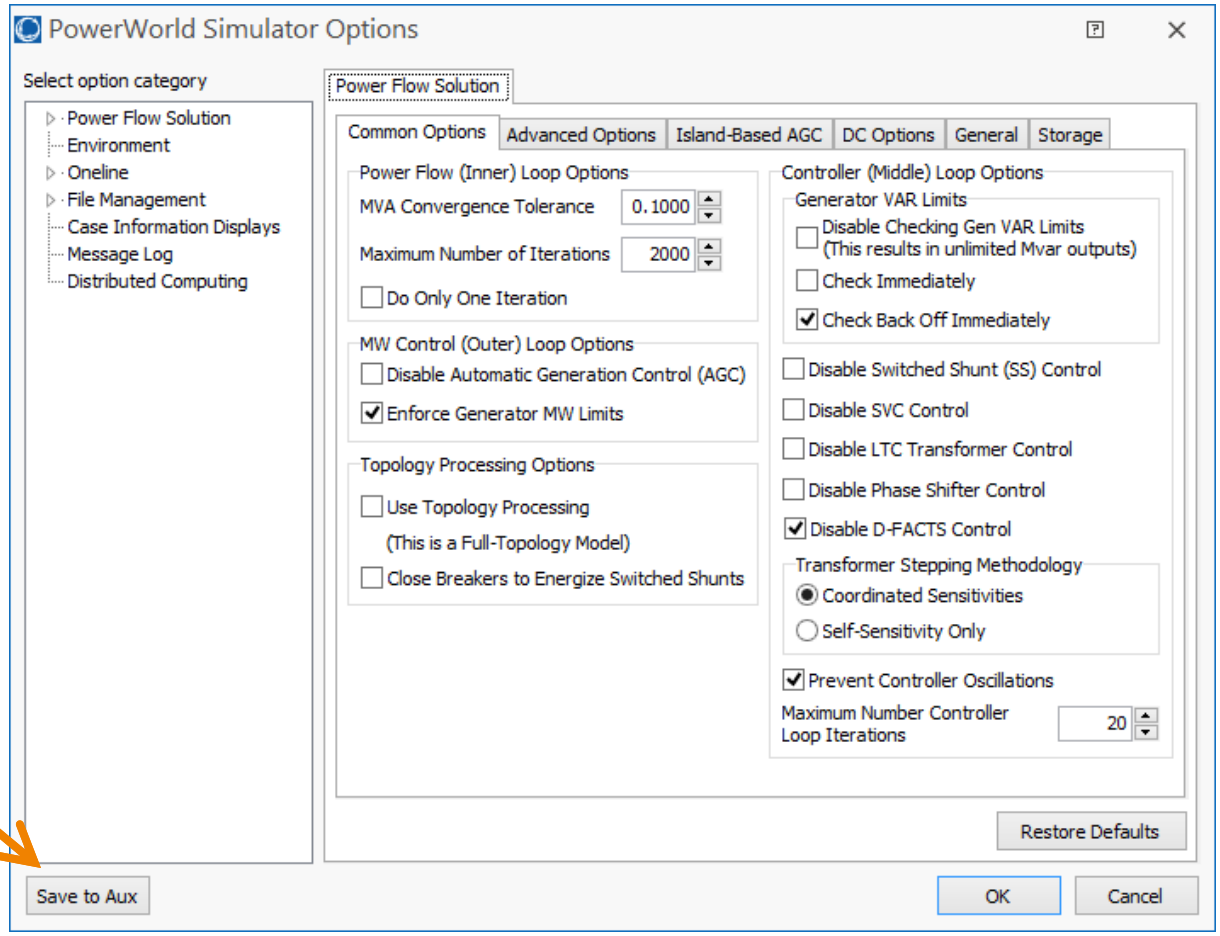



- Load the file *aux2010SolutionOptions.aux*
- This will load options stored in several DATA sections into the present case
- This file was created by saving the options from the Simulator Options dialog in the graphical user interface (GUI); more on next slide…

# Solution Options




- Export directly from GUI
- **Case Information → Simulator Options…**
- Save to Aux



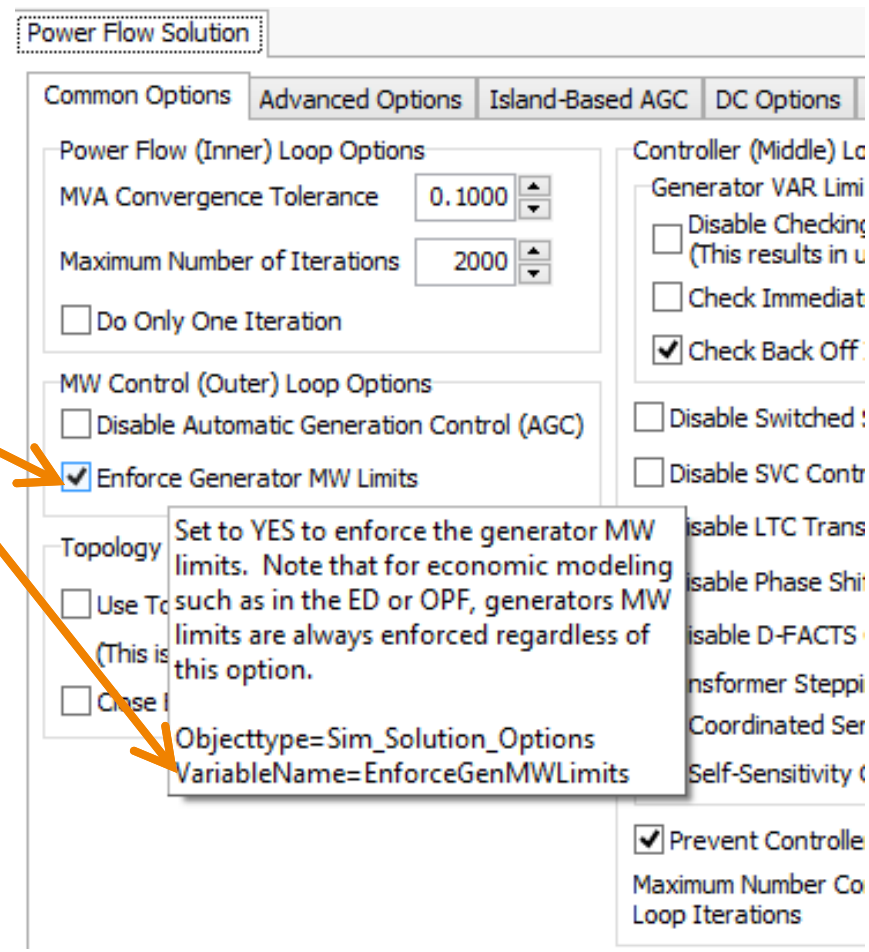
# Solution Options: DATA Section

- This DATA section lists each option and value of object *Sim_Solution_Options*
- *EnforceGenMWLimits* corresponds to a check box in the GUI (next slide)

```
Sim_Solution_Options_Value (Option,Value)
{
"AGCToleranceMVA"                    "5"
"ChkDFACTS"                          "NO"
"ChkMWAGC"                           "YES"
"ChkPhaseShifters"                   "YES"
"ChkShunts"                          "YES"
"ChkSVCs"                            "YES"
"ChkTaps"                            "YES"
"ChkVarBackoffImmediately"           "YES"
"ChkVarImmediately"                  "NO"
"CloseCBToEnergizeShunts"            "NO"
"ConsolidationUse"                   "NO"
"CTGInterfaceEnforcement"            "Never"
"DCApprox"                           "NO"
"DCLossComp"                         "NO"
"DCModelType"                        "RIgnore“
   :
"EnforceGenMWLimits"                 "YES“
   :
}
```

# Sim_Solution_Options

- Mouse over the checkbox in the dialog for a hint and the corresponding Object and Variable names
- Can also alter this value with the *SetData* script command (more info to follow)



# Filters and Expressions

- Load the file *aux2020FiltersExpressions.aux*
- Makes use of several SCRIPT sections and statements

Comment delimiter //

CreateData script action

```
SCRIPT
{
//--------------------
// Custom Filters
//--------------------
CreateData(Condition,
[ObjectType,Filter,CondNum,ObjectField,ConditionType,Value,OtherValue,Absolute],
["Area", "Study System", 1, "Number", "between", "3", "7", "NO "]);
:
}
```

- This statement creates a condition for the Area filter "Study System" (and the filter itself in the process)
  - consists of the areas with numbers between 3 and 7
  - these are California areas in our case

# Auxiliary File Reference

- Go to **Window → Auxiliary File Format**
- Opens a PDF document that describes auxiliary file structures, including a reference with structure and syntax of SCIRPT actions
- Search for *CreateData* action

**CreateData(objecttype, [fieldlist], [valuelist]);**

Use this action to create particular objects.

objecttype : The objecttype being created.

[fieldlist] : A list of fields to set with the object. **The key fields and required fields must be specified**.

[valuelist] : A list of values corresponding to the respective fields.

# Model Explorer

- Encapsulates most Case Information Displays
- Provides means of navigating through almost all of the data in the model
- Available from a few places
  - Case Information Ribbon Tab
  - Tools Ribbon Tab
  - Quick Access Toolbar

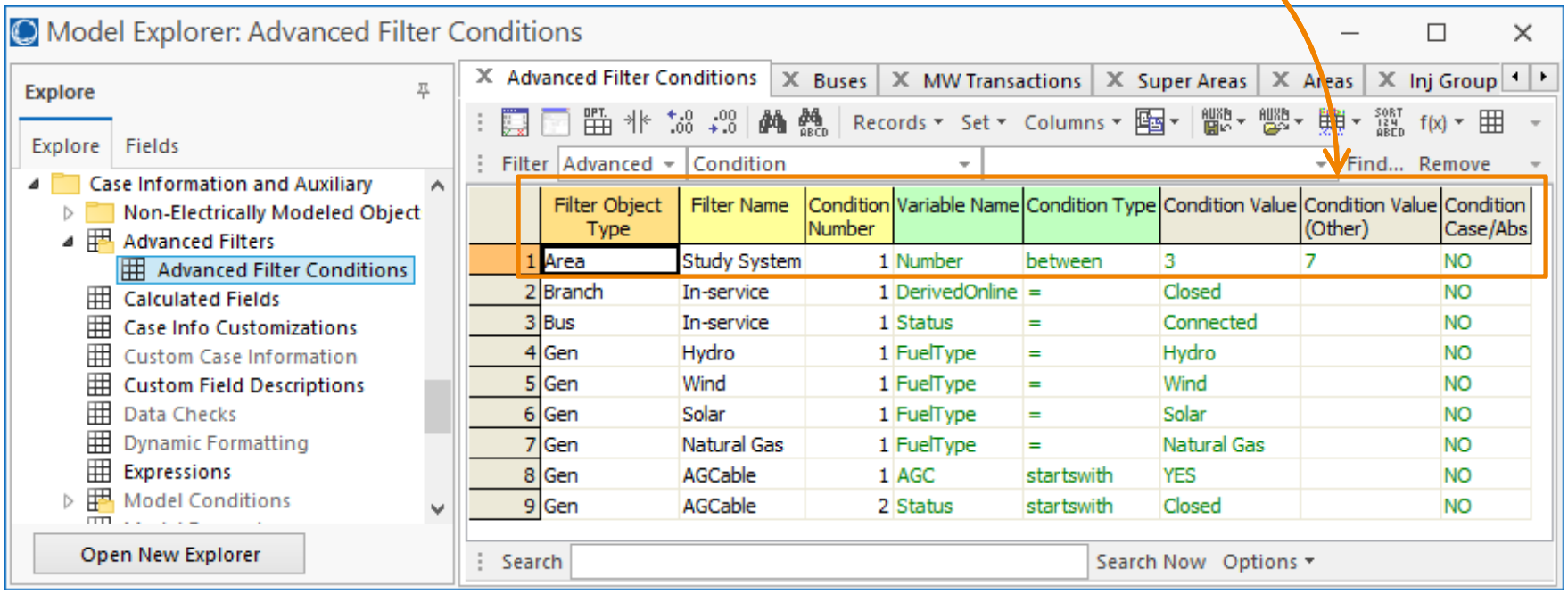
# Model Explorer

- In the Model Explorer, go to **Case Information and Auxiliary → Advanced Filters → Advanced Filter Conditions**
  - The  Advanced Filter Conditions are actually SUBDATA of the Advanced Filter object
  - Most Simulator SUBDATA can also be expressed as DATA, as they are with Advanced Filter Conditions

# Model Explorer

```
CreateData(Condition,
[ObjectType,Filter,CondNum,ObjectField,ConditionType,Value,OtherValue,Absolute],
["Area", "Study System", 1, "Number", "between", "3", "7", "NO "]);
```

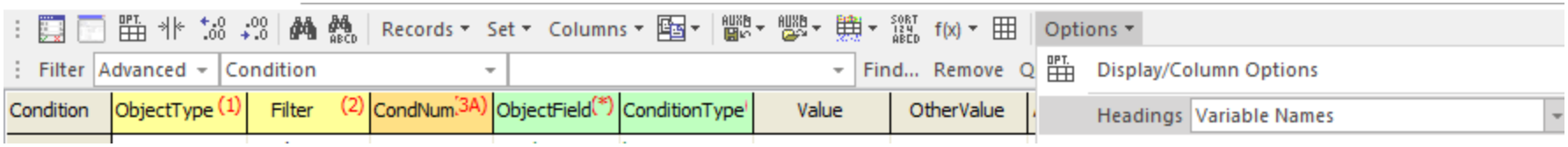
This script statement created this object




| | Filter Object Type | Filter Name | Condition Number | Variable Name | Condition Type | Condition Value | Condition Value (Other) | Condition Case/Abs |
|---|---|---|---|---|---|---|---|---|
| 1 | Area | Study System | 1 | Number | between | 3 | 7 | NO |
| 2 | Branch | In-service | 1 | DerivedOnline | = | Closed | | NO |
| 3 | Bus | In-service | 1 | Status | = | Connected | | NO |
| 4 | Gen | Hydro | 1 | FuelType | = | Hydro | | NO |
| 5 | Gen | Wind | 1 | FuelType | = | Wind | | NO |
| 6 | Gen | Solar | 1 | FuelType | = | Solar | | NO |
| 7 | Gen | Natural Gas | 1 | FuelType | = | Natural Gas | | NO |
| 8 | Gen | AGCable | 1 | AGC | startswith | YES | | NO |
| 9 | Gen | AGCable | 2 | Status | startswith | Closed | | NO |

# Model Explorer

- From the Case Info Toolbar, choose **Options → Headings → Variable Names**




- Also check the box **Use Concise Variable Names and Headers**

# Model Explorer

- Display now shows Object Name and Variable Names exactly as they appear in the *CreateData* SCRIPT statement

Variable Names (concise)

Object Name

| Condition | ObjectType (1) | Filter (2) | CondNum (3A) | ObjectField (*) | ConditionType | Value | OtherValue | Absolute |
|---|---|---|---|---|---|---|---|---|
| 1 | Area | Study System | 1 | Number | between | 3 | 7 | NO |
| 2 | Branch | In-service | 1 | DerivedOnline | = | Closed | | NO |

- Key Fields are highlighted yellow
  - uniquely identify each Object
  - must be included for most script actions (exceptions including referencing ALL objects or those that meet a filter)
- Required Fields are highlighted green
  - must be included to create new objects

# Generator Parameters

- Load the files
  - *aux2030LimitMonitoring.aux*
  - *aux2050Contingencies.aux*
  - *aux2060GeneratorFuelType.aux*

Generator Key Fields: BusNum and ID

```
Gen (BusNum,BusName,ID,FuelType)
{
 20283 "POLLOCK PI~2" "1 " "Hydro"
 20284 "POLLOCK PI~3" "1 " "Hydro"
 20288 "MCCLELLAN 1 " "1 " "Natural Gas"
 20293 "BURNEY 1 3  " "1 " "Wind"
 20295 "BELDEN 2    " "1 " "Hydro"
:
}
```

- The last contains a single DATA section that sets the FuelType variable for the units inside California
- FuelType is used by filters to set AGC field in subsequent file

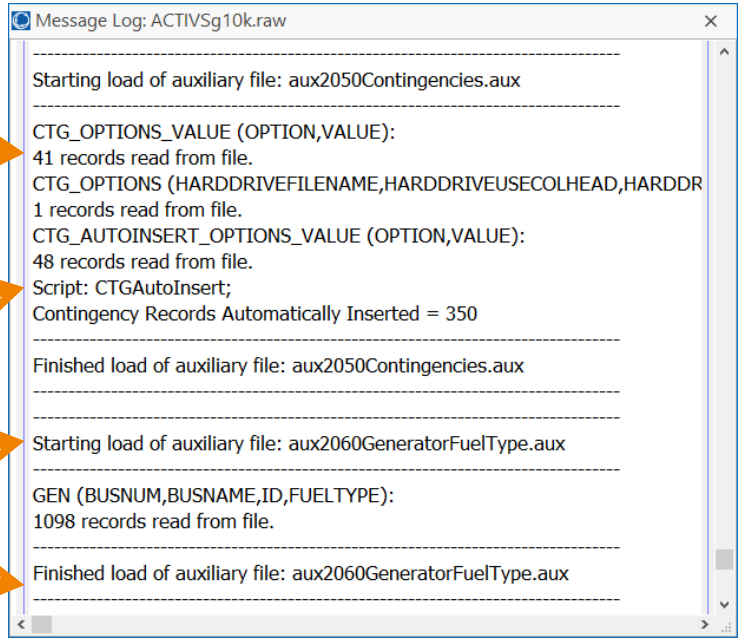
# Auxiliary File Tips

- Use the Simulator GUI!!!
- Test and debug auxiliary files by loading from the GUI and examining **Message Log**

DATA Sections: which objects and variables; how many records read and skipped, if any

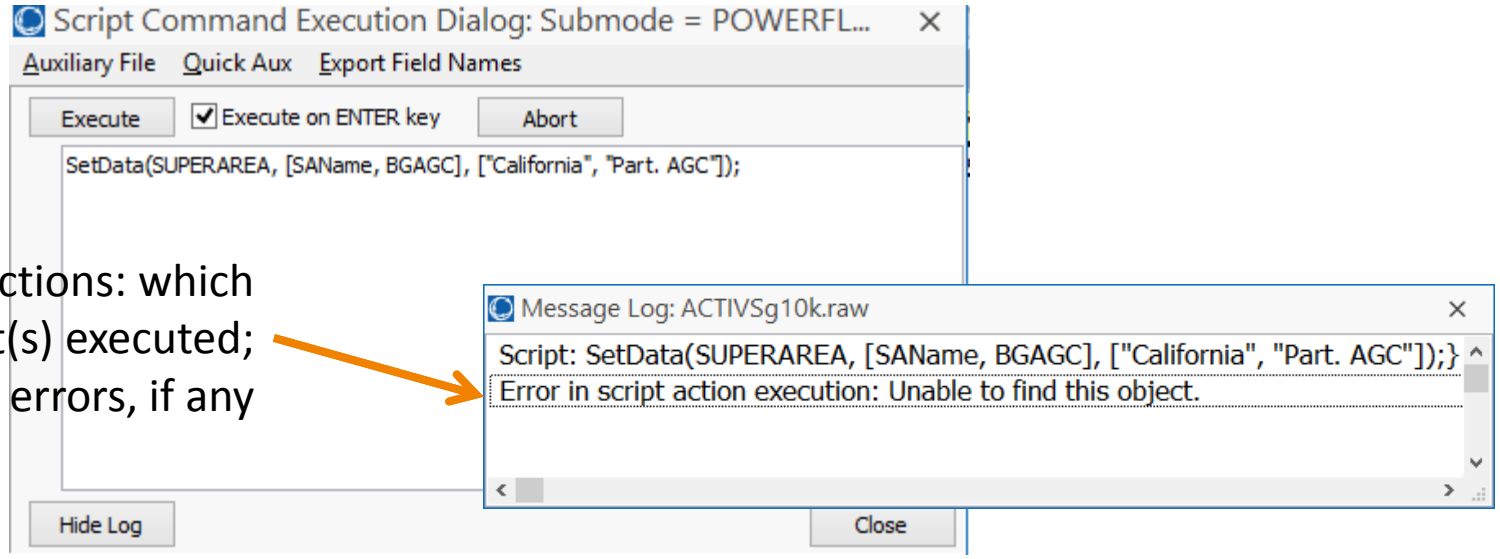
SCRIPT Sections: which statement(s) executed; errors, if any

Start and Finish of Loaded Files



# Auxiliary File Tips

- Use the **Script Command Execution Dialog** to test and debug individual statements
- Choose **Script** from the **Tools** Ribbon



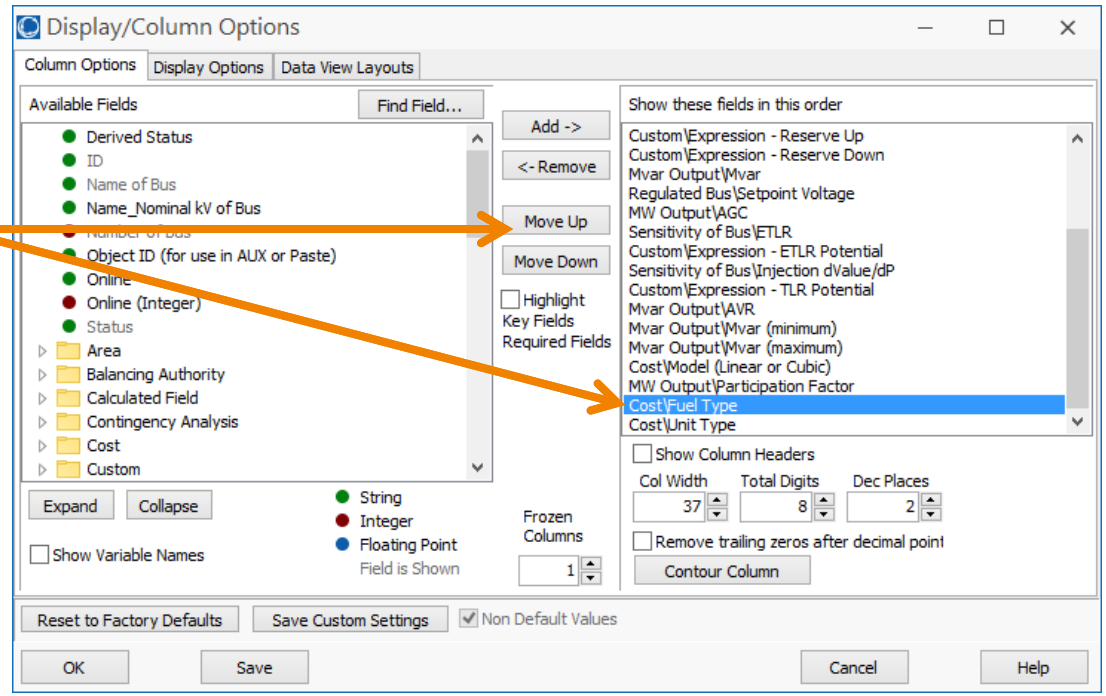

SCRIPT Sections: which statement(s) executed; errors, if any

# Creating a DATA Section from Case Info Display

- Suppose we wish to create a file with generator fuel types, as the one just loaded
- In the Model Explorer, go to **Network → Generators**
- Tip: save only key fields and the records and columns necessary to make needed changes
  - Example: if setting generator Fuel Type is the objective, do not include other fields such as Gen MW, Gen Max MW, etc.
  - Extra fields may be specific to one case and not appropriate for other cases

# Creating a DATA Section

- To choose fields to save to the Aux file, choose the **Display/Column Options** from the Case Info Toolbar
- Move *Cost\Fuel Type* up so that it directly follows *ID*
- Click **OK**



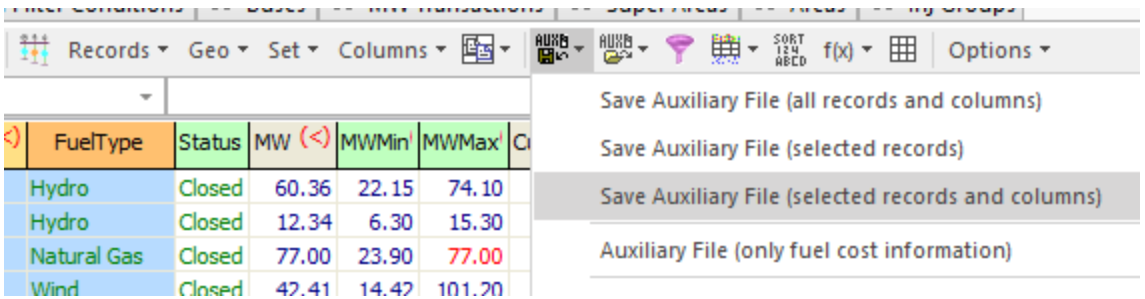
# Creating a DATA Section

- Click and Drag to select the first 4 fields for at least one record; these include
  - Key fields: BusNum and ID
  - FuelType: what we want to store
  - BusName: not needed, but aids readability
- From the Case Info Toolbar, choose **Columns → Select Column(s)**

| Gen | BusNum (1<) | BusName (<) | ID (2B<) | FuelType | Status | MW (<) | MV |
|---|---|---|---|---|---|---|---|
| 1 | 20283 | POLLOCK PI | 1 | Hydro | Closed | 60.36 | |
| 2 | 20284 | POLLOCK PI | 1 | Hydro | Closed | 12.34 | |
| 3 | 20288 | MCCLELLAN | 1 | Natural Gas | Closed | 77.00 | |
| 4 | 20293 | BURNEY 1 3 | 1 | Wind | Closed | 42.41 | |
| 5 | 20295 | BELDEN 2 | 1 | Hydro | Closed | 85.30 | |

# Creating a DATA Section

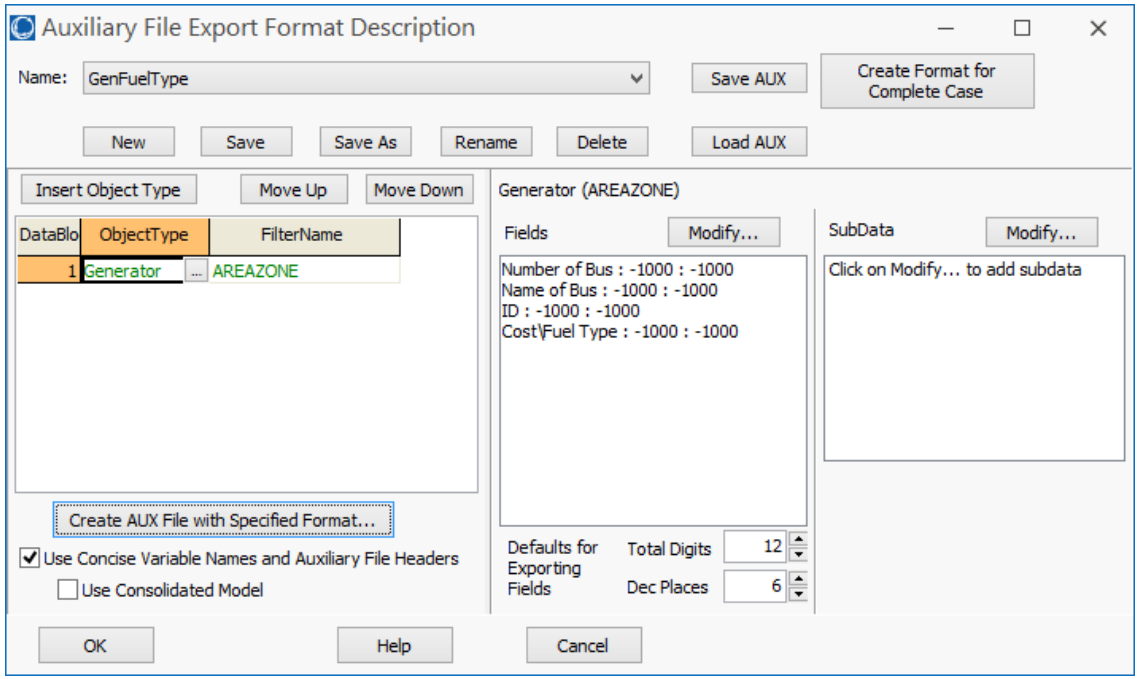
- From the Case Info Toolbar, choose **Aux Save → Save Auxiliary File (selected records and columns)**




- Create a new file name and click **Save**
- For some Object Types, Simulator may prompt if you would like to save certain types of sub data (e.g. BidCurve and ReactiveCapability for Generators)

# Creating a DATA Section

- Can also create an AUX Export Format Description (**Case Information → AUX Export Format Desc…**)
- Insert Object Type Generator
- Use FilterName AREAZONE
- Choose fields
- **Save As**



# More on Simulator Object Fields

- Re-open the **Display/Column Options**
- A list of all available object fields appears on the left, arranged in folders
- You may customize the display to show a desired set of fields, order, and other characteristics
- Click **Find Field…** to search by string
- Special folders for Key Fields and Required Fields – verify they are shown, if needed

# More on Simulator Object Fields

- Customizations may be viewed in the Model Explorer under **Case Information and Auxiliary → Case Info Customizations**
- These also occur in *aux2020FiltersExpressions.aux*
- List of Columns, width, and decimals are given as SUBDATA ColumnInfo

```
DataGrid (Name,FrozenCols)
{
"Area" 4
   <SUBDATA ColumnInfo>
     "Number"                45  8 2
     "Name"                  75  8 2
     "AGC"                   54  8 2
     "GenMW"                 75  8 2
     "LoadMW"                75  8 2
     "ShuntMW"               75  8 2
     "ExportMWSched"         75  8 2
     "ExportMW"              75  8 2
     "ACE"                   75  8 2
     "EconDispLambda"        75  8 2
     "LossMW"                75  8 2
     "CalcField:0"           58  8 0
     "CalcField:1"           47  8 0
     "CalcField:2"           55  8 0
     "AutoControlShunt"      75  8 2
     "AutoControlXF"         75  8 2
     "SlackBus"              59  8 2
     "ExportMWUnspecified"   75  8 2
     "AGCTolerance"          75  8 2
   </SUBDATA>
:
}
```

# More on Simulator Object Fields

- To Export a complete reference, choose **Window → Export Case Object Fields… → Send to Excel** (or Text File)
- Details for DataGrid object

| | A | B | C | E | F | G |
|---|---|---|---|---|---|---|
| 1 | Object Ty | SUBDATA | Key/Requ | Concise Variable Name | Type of Va | Description |
| 3947 | DataGrid | | | | | |
| 3948 | | ColumnInfo | | | | |
| 3949 | | | | FilterPre | String | Use Filters? |
| 3950 | | | | RowHeight | Integer | Row Height |
| 3951 | | | | SortOrder | String | Sort Direction |
| 3952 | | | *1* | Name | String | Name |
| 3953 | | | | Filter | String | Filter Name |
| 3954 | | | | FontColor | Integer | Font Color |
| 3955 | | | | FontName | String | Font Name |
| 3956 | | | | FontStyles | String | Font Styles |
| 3957 | | | | FrozenCols | Integer | The number of frozen c |
| 3958 | | | | FontNonDefault | String | Nondefault Font? |
| 3959 | | | | RemoveTrailingZeros | String | Set to YES to remove tr |
| 3960 | | | | FontSize | Integer | Font Size |
| 3961 | | | | SortVariable | String | Sorted by |
| 3962 | | | | Zoom | Real | The percent zoom leve |

# Auxiliary Files

- Load the file *aux2070GeneratorAGC.aux*
  - Set of SCRIPT statements that set Generator AGC status based on fuel type and set Area and SuperArea AGC modes
- TIP: Use script actions to initialize standard or default values for ALL objects of a given type
  - e.g. *SetData(objecttype, [fieldlist], [valuelist], ALL);*
  - Objects that need different values or to be handled specially (e.g. study areas) can be identified by primary key or filter in specific statements
  - Improves robustness and compatibility with different cases having different objects and topology – easier to manage files and code over a life cycle
  - Usually makes code more compact and easier to read

# Tip: Initialize Values

Initialize for ALL

Settings for specific Generators by exception, using Filters

Setting for specific Area by exception, using Primary Key. Code will adapt easily to future cases with additional (or fewer) areas

Assigning Areas to SuperArea by Filter, rather than listing each individual area, means we only need to maintain the filter definition in one place.

```
SCRIPT
{
// Initialize AGC for ALL units
SetData(GEN, [GenAGCAble], [YES], ALL);

// Set all Hydro, Wind, and Solar Units to NO AGC
//   based on Gen Filters
SetData(GEN, [GenAGCAble], [NO], "Hydro");
SetData(GEN, [GenAGCAble], [NO], "Wind");
SetData(GEN, [GenAGCAble], [NO], "Solar");

// Initialize all areas to Participation Factor Control,
//   AGC Tolerance = 5 MW
SetData(AREA, [BGAGC, ConvergenceTol], ["Part. AGC", 5], ALL);

// Set area with Island Slack Buses to Off AGC
SetData(AREA, [AreaNum, BGAGC], [9, "Off AGC"]);
:
// Create California SuperArea on Participation Factor control
//    and place Study System areas in California SuperArea
CreateData(SUPERAREA, [SAName, BGAGC], ["California", "Part. AGC"]);
SetData(AREA, [SuperArea], ["California"], "Study System");
:
}
```

# Auxiliary File Tip Summary

- Use the Simulator GUI!!!
  - Model Explorer, Case Information Displays, and Analysis Dialogs for exploring objects and variables and saving to Auxiliary File
  - Message Log and Script Execution Dialog
- Use the References!
  - Auxiliary File Format (pdf document)
  - Export Case Object Fields…

# Auxiliary File Tip Summary

- Build files by saving Case Info Displays and settings to auxiliary files
  - Use text editor to review, make changes, and add comments
  - Can append new DATA sections to existing auxiliary files
  - Add SCRIPT sections where appropriate
  - Most Options dialogs in Simulator have a button for Saving to Aux
- DATA sections: save only key fields and the records and columns necessary to make needed changes
  - Example: if setting generator Fuel Type is the objective, do not include other fields such as Gen MW, Gen Max MW, etc.
  - Extra fields may be specific to one case and not appropriate for other cases
- Add comments to document your process

# Limit Monitoring Settings

- Open **Tools → Limit Monitoring…**
- Can Save an Auxiliary File directly from this dialog (**Save Monitoring Settings**)
- This is a great start, but we might wish to clean it up using tips and principles discussed
  - Individual Areas and Zones are listed in DATA sections
  - What if our next case has 17 areas, but we still wish to monitor only Areas 3-7? This file would leave Area 17 Monitored for all voltages (default values).

# Limit Monitoring Settings

Excerpt from file created from Limit Monitoring dialog. In a future case with Area 17, Area 17 would be monitored.

```
:
Area
(Number,MonitorLimits,MonitorMinkV,Monitor
MaxkV)
{
         1 "NO "   100.0000  9999.0000
         2 "NO "   100.0000  9999.0000
         3 "YES"   100.0000  9999.0000
         4 "YES"   100.0000  9999.0000
         5 "YES"   100.0000  9999.0000
         6 "YES"   100.0000  9999.0000
         7 "YES"   100.0000  9999.0000
         8 "NO "   100.0000  9999.0000
         9 "NO "   100.0000  9999.0000
        10 "NO "   100.0000  9999.0000
        11 "NO "   100.0000  9999.0000
        12 "NO "   100.0000  9999.0000
        13 "NO "   100.0000  9999.0000
        14 "NO "   100.0000  9999.0000
        15 "NO "   100.0000  9999.0000
        16 "NO "   100.0000  9999.0000
}
:
```

This would handle extra Areas as desired (or fewer Areas, without errors).

- We only need to maintain the definition of the “Study System” filter.
- Code is also much more compact.

```
SCRIPT
{
//  set all Areas to not monitor
//   report nominal kV between 100-9999 kV
SetData(Area,
[BGReportLimits,BGReportLimMinKV,BGReportLimMaxKV], ["NO",
100, 9999], ALL);

// set only Study Areas to monitor
SetData(Area, [BGReportLimits], ["YES"], "Study System");
:
}
```

# One More Tip!

- Use a Master file to call secondary files (*LoadAux*)
  - Overall procedure can be maintained in the master file
  - Parameters subject to change over time (e.g. generator specifics) can be stored in the secondary files and more easily updated or replaced without disrupting other parts
  - Can suppress confirmation dialogs when creating new objects
  - Comment out statements that load files that are temporarily not needed
- *aux2000Master.aux*
- It helps to fully check and debug individual files first, though review of the log from loading Master file can still help catch errors

# Master File

```
SCRIPT
{
// Power Flow Solution Options
LoadAux("aux2010SolutionOptions.aux", Yes);

// Custom filters, expressions, calculated fields, and data grids
LoadAux("aux2020FiltersExpressions.aux", Yes);

// Limit Monitoring Settings
LoadAux("aux2030LimitMonitoring.aux", Yes);

// Contingency options and auto-insertion
LoadAux("aux2050Contingencies.aux", Yes);

// Generator Fuel Types
LoadAux("aux2060GeneratorFuelType.aux", Yes);

// Generator AGC Settings
LoadAux("aux2070GeneratorAGC.aux", Yes);

// Multiple Element TLR Calculation - perform on base case or with contingencies
//LoadAux("aux2080TLRBaseCase.aux", Yes);
//LoadAux("aux2081TLRCTGCase.aux", Yes);
}
```

Yes in second argument suppresses confirmation dialog for new objects

"Comment out" statements that load files that are temporarily not needed

# Sensitivity Analysis with Aux Files

- Remaining files calculate Multiple Element TLR/Shift Factors and save results by bus and SuperArea to csv files
  - *aux2080TLRBaseCase.aux*: calculates on Overloaded Lines and Transformers in the base case
  - *aux2081TLRCTGCase.aux*: calculates on Overloaded Lines and Transformers in contingency analysis
- Custom Expressions and Calculated Fields created earlier compute the estimated loading relief benefit of changing control outputs (generators and loads) and the sum of overloads

# Sensitivity Analysis

- Go to **Aggregations → Super Areas** in the Model Explorer
- Three Calculated Fields show
  - MW flow on the branch with the highest MW flow in the Super Area
  - Sum of base case MVA transmission overloads
  - Sum of contingency MVA transmission overloads
- These fields are also available in Areas/Zones/etc.

| | Super Area | AGC Status | Use Area PF | Num Areas | Gen MW | Load MW | Tot Sched MW | ACE MW | Maximum Branch MW | Aggregat Overload | AMVACO | Loss MW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | California | Part. AGC | NO | 5 | 59529 | 67019 | -8294 | -0 | 2698 | 175 | 0 | 804 |

# Sensitivity Analysis

- Load *aux2080TLRBaseCase.aux*
- Go to **Network → Generators** in the Model Explorer
- Sort on Custom Expression *ETLR Potential* in descending order

| | Number of Bus | Name of Bus | ID | Status | Gen MW | Min MW | Max MW | Reserve Up | Reserve Down | Gen Mvar | Set Volt | AGC | ETLR | ETLR Pote ▼ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 25938 | EL SEGUND~ | 1 | Open | 0.00 | 113.75 | 342.00 | 342.00 | 0.00 | 0.00 | 1.02268 | YES | -0.5564 | 190.2751 |
| 2 | 26063 | SUN VALLEY~ | 1 | Closed | 210.00 | 88.56 | 210.00 | 0.00 | 121.44 | 106.89 | 1.02410 | YES | 0.5054 | 106.1344 |
| 3 | 26061 | SUN VALLEY~ | 1 | Closed | 210.00 | 31.17 | 210.00 | 0.00 | 178.82 | 106.89 | 1.02410 | YES | 0.5054 | 106.1344 |
| 4 | 25829 | OXNARD 9 6 | 1 | Open | 0.00 | 149.59 | 806.00 | 806.00 | 0.00 | 0.00 | 1.04200 | YES | -0.0972 | 78.3521 |
| 5 | 26128 | BURBANK 10 | 1 | Open | 0.00 | 53.79 | 193.80 | 193.80 | 0.00 | 0.00 | 1.03900 | YES | -0.2717 | 52.6551 |
| 6 | 26133 | AVILA BEAC~ | 1 | Closed | 1161.50 | 165.83 | 1161.50 | 0.00 | 995.67 | 59.76 | 1.02312 | YES | 0.0329 | 38.2561 |
| 7 | 26134 | AVILA BEAC~ | 1 | Closed | 1161.50 | 130.25 | 1161.50 | 0.00 | 1031.25 | 59.76 | 1.02312 | YES | 0.0329 | 38.2561 |
| 8 | 26064 | SUN VALLE~1 | 1 | Closed | 60.50 | 26.43 | 60.50 | 0.00 | 34.07 | 30.79 | 1.02410 | YES | 0.5054 | 30.5768 |
| 9 | 26163 | MC KITTRIC~ | 1 | Closed | 600.00 | 194.01 | 600.00 | 0.00 | 405.99 | -66.60 | 1.00283 | YES | 0.0404 | 24.2682 |
| 10 | 26173 | LONG BEAC~ | 1 | Open | 0.00 | 58.82 | 230.00 | 230.00 | 0.00 | 0.00 | 1.04080 | YES | -0.0786 | 18.0781 |
| 11 | 29084 | SUN CITY 4 5 | 1 | Open | 0.00 | 107.66 | 409.50 | 409.50 | 0.00 | 0.00 | 1.04600 | YES | -0.0412 | 16.8698 |
| 12 | 23377 | MOSS LANDI~ | 1 | Closed | 497.62 | 328.03 | 702.00 | 204.38 | 169.59 | -28.06 | 1.02312 | YES | 0.0337 | 16.7880 |

# Sensitivity Analysis

- Sign convention in calculations and expressions
  - Generators with (+) ETLR can relieve overloads by *decreasing* output or dropping
  - Loads with (-) ETLR can relieve overloads by shedding
- Connect unit 25938 1 and set Gen MW = Max MW
- Solve Power Flow
- How does Aggregate Overload change for Super Area? (drops from 175 to 133)
- Also try load shedding or with contingencies (*aux2081TLRCTGCase.aux*)