

Introduction to PowerWorld Simulator: Interface and Common Tools



I4: Auxiliary File Format: DATA Section



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

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

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
- The auxiliary file format also extends to oneline diagrams (*.axd) except for a different set of script commands

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 **online diagram**

Example: Using Aux Files to Customize Simulator Environment



- Automatically load aux file each time a case is loaded
- Allows standardization of options
- Use to customize
 - Area/Zone/Owner Filters
 - Advanced Filters
 - Custom Expressions
 - Custom Data Grids
 - Many more
- Sample file series
 - *I04_Auxiliary File Data Section\aux30?0*.aux*
 - *aux3000Defaults.aux* loads the other files
 - Intended to work with *ACTIVSg25k.pwb*

Automatically Load File



Load a file for
PRESENT case
or ANY case

PowerWorld Simulator Options

Select option category

- Power Flow Solution
- Environment
- Online
- File Management**
- Case Information Displays
- Message Log
- Distributed Computing

File Management

PowerWorld Files | EPC and RAW Files | hdbexport Files

☐ Automatic Loading of Auxiliary File

☐ Automatically load an Auxiliary File when the present case is opened

Auxiliary File

(All settings below are only saved to the Registry)

☐ Automatic Loading of Auxiliary File with ANY case

☐ Automatically load an auxiliary file when ANY case is opened

Auxiliary File

Automatic Archiving of PWB files

☐ Enable Auto-Archive of PWB Files

Delimiter in Archive File Name

☒ _ (underscore) ☐ ~ (tilde)

☐ - (dash) ☐ ; (semicolon)

☐ other

☐ Save Unlinked Elements of contingency, interface and injection group records in the PWB file

Note: The Automatic Autosave will only archive cases under Edit Mode

Maximum Number of Archive Files

Note: If File Location is empty then the files will be saved in the current case file location.

File Location

Auto Load Directory

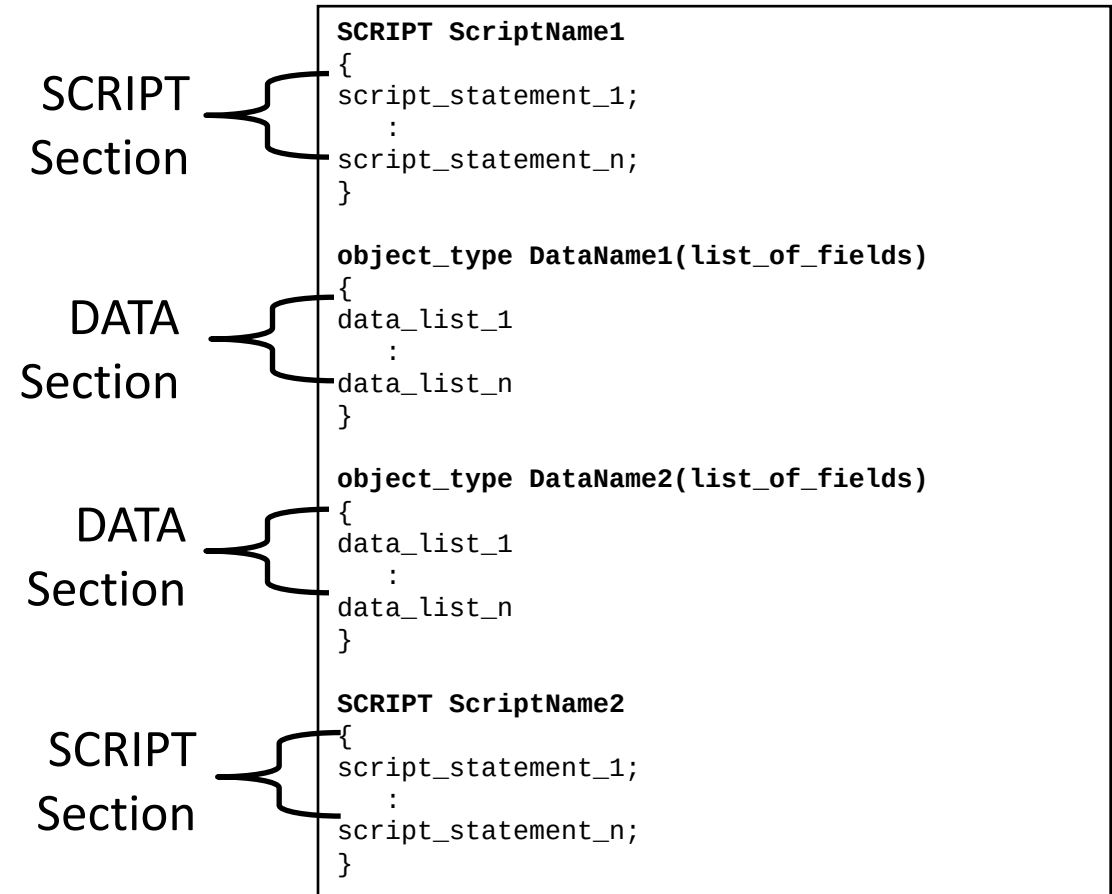
Location

Refresh Interval s

Auxiliary File Format Overview



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



Auxiliary File

SCRIPT Sections



- Start with the word **SCRIPT**
- An optional script name may follow
 - For use with the LoadScript Action
- Then a block of script actions follow enclosed in curly braces { }
- Each script statement must end in a semicolon ;
- All the script actions allowed will be covered in a later set of slides.

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

Auxiliary File DATA Sections



- Start with a valid Object_Type string*
- An optional data name may follow
 - For use with the LoadData Action
- Following this is a list of fields enclosed in parenthesis
 - (list_of_fields)

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

*Note: In the legacy format, DATA sections start with “DATA” string. Files in this format can still be loaded. DATA sections are saved in this format if the option “Use Concise Variable Names and Headers” in the Case Info Toolbar is unchecked.

Creating an Auxiliary File



- Most grids in Simulator have an option to save data to an auxiliary file
 - Save Auxiliary Files toolbar button
 - Right-click and choose **Save As → Auxiliary File**
- Save Auxiliary Files toolbar button

The screenshot shows the 'Model Explorer: Branches Input' window. The 'Explore' pane on the left shows the 'Branches Input' grid selected. The main window displays a table of branch data. A right-click context menu is open over the table, showing options to save the data to an auxiliary file. An orange arrow points from the text 'Save Auxiliary Files toolbar button' to the 'Save Auxiliary File (all records and columns)' option in the menu.

Branch	BusNumFrom	BusNameFrom	BusNumTo	BusNameTo	Circuit	Status
1	1	One	2	Two	1	Closed
2	1	One	3	Three	1	Closed
3	2	Two	3	Three	1	Closed
4	2	Two	4	Four	1	Closed
5	2	Two	5	Five	1	Closed
6	2	Two	6	Six	1	Closed
7	3	Three	4	Four	1	Closed
8	4	Four	5	Five	1	Closed
9	7	Seven	5	Five	1	Closed
10	6	Six	7	Seven	1	Closed
11	6	Six	7	Seven	2	Closed

Save Auxiliary File (all records and columns) Ctrl+Alt+A

Save Auxiliary File (selected records) Shift+Alt+A

Save Auxiliary File (selected records and columns) Shift+Ctrl+A

CSV (Comma delimited)...>Variable Names

CSV (Comma delimited)...>Column Headers

CSV (only selected records/columns)...>Variable Names

CSV (only selected records/columns)...>Column Headers

HTML...

Bitmap...

JPEG...

GE EPC format (only selected records)

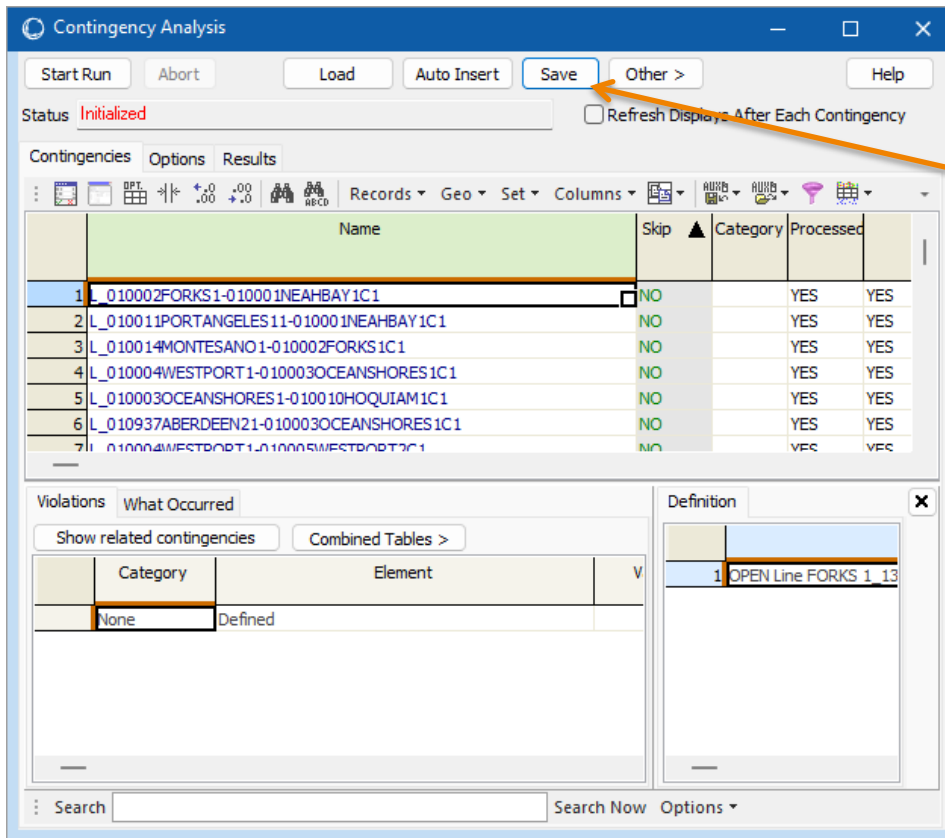
GE DYD format (only models of selected records)

Other

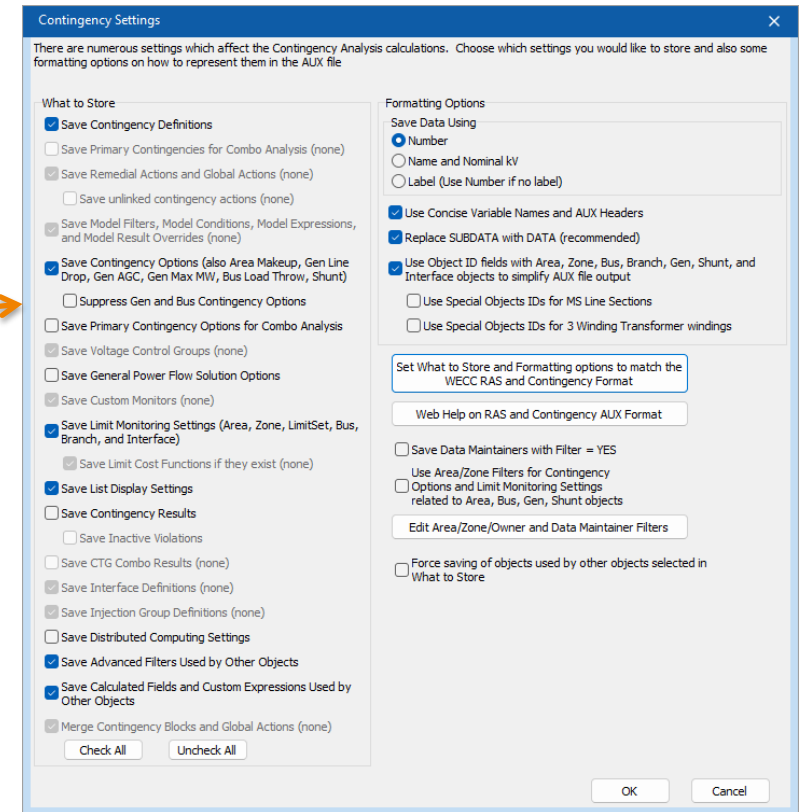
Creating an Auxiliary File



- Some dialogs have a **Save** button that will create an auxiliary file with all relevant information (e.g. Contingency Analysis)



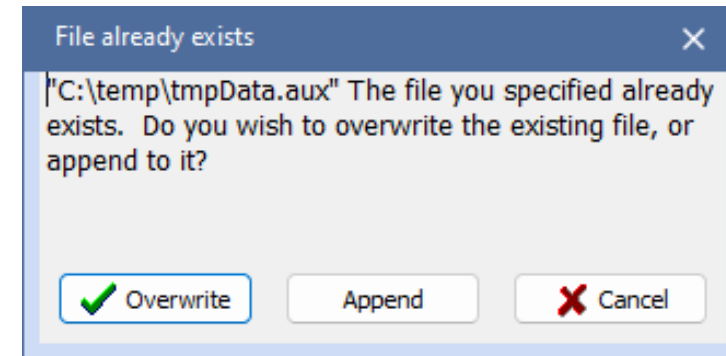
Click Save,
then another
dialog
appears to
select which
details to
include



Creating an Auxiliary File



- Anytime you Save information to an auxiliary file, the subsequent file dialogs allow options to create a new file, overwrite an existing file, or append to an existing file



- You can thus progressively build files with multiple DATA sections

DATA Section

Object_Types



- Identifies the type of object or data element the DATA section describes or models
- Some examples include:

Area	DCTransmissionLine	PartPoint
ATC_Options	Direction	PVCurve_Options
ATCExtraMonitor	Equiv_Options	QVCurve_Options
ATCGeneratorChange	Filter	ReactiveCapability
ATCInterfaceChange	Gen	Scale_Options
ATCLineChange	InjectionGroup	Schedule
ATCScenario	Interface	Shunt
ATCZoneChange	InterfaceElement	Sim_Environment_Options
BidCurve	Limit_Monitoring_Options	Sim_Misc_Options
Branch	LimitSet	Sim_Simulation_Options
Bus	LimitViol	Sim_Solution_Options
ColumnInfo	Load	StudyMWTransactions
Condition	LoadScale	Substation
Contingency	ModelCondition	SuperArea
Ctg_Options	ModelFilter	TransferLimiter
CTGElement	MSLine	Transformer
CustomExpression	MWTransaction	XFCorrection
DataGrid	OPF_Options	Zone
	OwnerObject	

DATA Section

list_of_fields



- Lists the types of values the records in the DATA section contain
- The order of the fields dictate the order that fields will be read from the DATA section
- Simulator recognizes over 10,000 total fields
 - Only a small subset will be used with a particular type of object
 - A list of available fields for each object type may be obtained on the Windows ribbon tab. On the Auxiliary Files ribbon group, choose **Export Case Object Fields → Text File** or **Send To Excel**

DATA Section

list_of_fields Parameter



- Specifications for a list_of_fields
 - Enclosed in parentheses (brackets []) for legacy format)
 - May take up several lines of text
 - You may use the Simulator comment string //
 - Simulator will ignore all text to the right of the double slash
 - Blank lines, or lines which begin with the comment string will be ignored
 - Field Names must be separated by commas
 - Field Names may be optionally augmented with a field location integer (by default, we assume Location is 0)
 - Format is Fieldname:Location
 - CustomInteger:0 means get the first custom integer
 - CustomInteger:1 means get the second custom integer

DATA Section

Key Fields



- Each object type has a few fields that serve as key fields for Simulator.
- These fields must be included in the list_of_fields.
 - Buses: Number
 - Lines: BusNumFrom, BusNumTo, Circuit
 - Loads: BusNum, ID
- You can get a list of the key fields by going to the Windows Ribbon Tab. On the Auxiliary Files Ribbon Group, choose **Export Case Object Fields → Text File** or **Send To Excel**
 - The key fields will be denoted in this output with asterisks (*1*, *2*, and *3*)

DATA Section

Secondary Key Fields



- Secondary Key fields
 - Often represent a combination of Name/Nominal kV for bus-related objects
 - The secondary key fields will be denoted in this output with asterisks (*A*, *B*, and *C*)
- When pasting into Simulator, we first look to see if the key fields are available. If they are not, then we look for the secondary key fields and use them instead.
- NOTE: The fields that make up Secondary Keys are generally not immutable and may not be unique
 - A case may have multiple buses with the same name and nominal kV.
 - If you load an aux for an object type where this occurs, Simulator will change the data for only one such object with matching Secondary Key.
 - The Secondary Key should be used with caution and only as a heuristic for transferring data between different cases where the Primary Key is unreliable (e.g. bus numbers have changed).



Labels



- Unique identifier for an object of a particular type
- Refer to equipment in the model in a way unique to your organization
- Likely to change less frequently than bus numbers
- Can be used for pasting and updating
 - Label (for use in input from AUX or Paste)

DATA Section

Required Fields



- You can also create new objects by reading in data from the DATA Section of an AUX file.
- Be careful. You must provide all Key AND **Required Fields** regarding the object.
 - e.g. To create a bus, include Number, Name, Nominal kV, Area Number, and Zone Number.
- Required Fields can be highlighted in green by using the Highlight Key/Required fields options.
 - In the Display/Column Options
 - In the Headings caption of the case info display

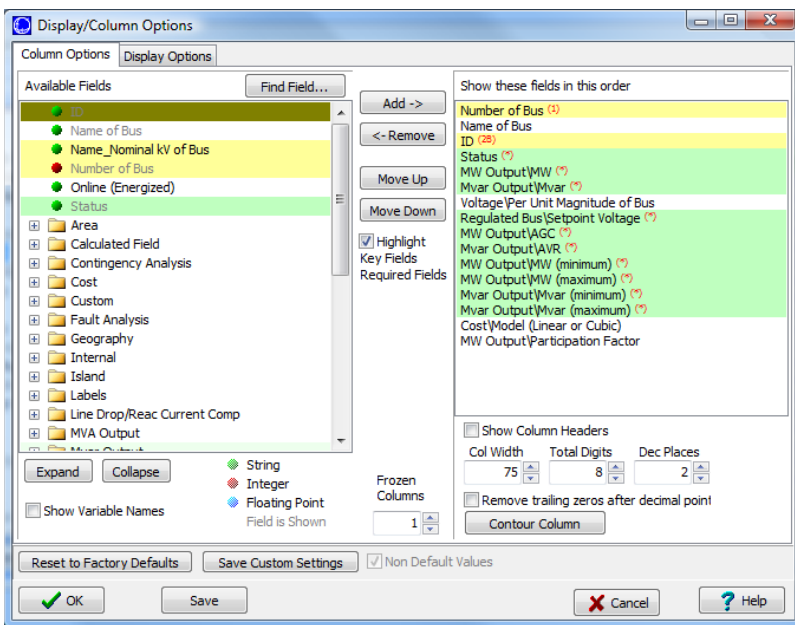
Finding Key and Required Fields in the Simulator Interface



- Key Fields are yellow, Required Fields are Green

	Number of Bus	Name of Bus	ID	Status	Gen MW	Gen Mvar	PU Volt of Bus	Set Volt	AGC	AVR	Min MW
1	30	CONE G1	H	Closed	445.00	226.73	1.050	1.05000	YES	YES	262.0
2	30	CONE G1	L	Closed	405.00	205.60	1.050	1.05000	YES	YES	238.0
3	31	CONE G2	H	Closed	445.00	-21.00	0.970	0.96671	YES	YES	262.0
4	31	CONE G2	L	Closed	405.00	-19.00	0.970	0.96671	YES	YES	238.0
5	32	KEYS G1	H	Closed	433.00	5.00	0.959	0.95862	YES	YES	262.0

- Or in the Display/Column Options Dialog



ONLY For Legacy Auxiliary File Header DATA Sections



- Start with the word DATA
- An optional data name may follow
 - For use with the LoadData Action
- Following this is a list of parameters enclosed in parenthesis
 - (object_type,[list_of_fields],file_type_specifier,create_if_not_found)
 - object_type
 - [list_of_fields]
 - file_type_specifier
 - create_if_not_found

```
DataName1(object_type, [list_of_fields],  
file_type_specifier, create_if_not_found)  
{  
  data_list_1  
  :  
  data_list_n  
}
```

ONLY For Legacy DATA Section file_type_specifier Parameter



- Simulator presently accepts two values
 - none specified
 - Fields which follow are space-delimited
 - CSV
 - Fields which follow are comma-separated

ONLY For Legacy DATA Section

create_if_not_found Parameter



- Optional field
- Specifies whether or not to create a new object if an existing one is not found
 - YES to create object
 - NO to deny creation
 - If omitted, user is prompted
 - If using the LoadAux script command, create_if_not_found will override the CreatelfNotFound script parameter

DATA Section

Data List



- Following the argument list describing the DATA section, the Data List is given
- Starts with a left curly brace {
- Ends with a right curly brace }
 - Right curly brace must appear on its own line of text
- Between the curly braces, any number of data lists can be entered
 - Each data list represents a single object in Simulator

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

DATA Section

Data List



- Specifications of a data list
 - May take up several lines of text
 - Simulator will read the number of fields specified in `list_of_fields`
 - Each new data object must start on its own line of text
 - You may use the Simulator comment string `//`
 - Simulator will ignore all text to the right of the double slash
 - Blank lines, or lines which begin with the comment string will be ignored
 - Remember that the right curly brace must appear on its own line at the end of the `data_list`
 - Fields separated by space or comma depending on `file_type_specifier`
 - Strings can be enclosed in double quotes, but this is not required
 - You should however always enclose strings that contain spaces (or commas) in quotes. Otherwise, strings containing commas would cause errors for comma-delimited files, and spaces would cause errors for space-delimited formatted files.

Example DATA Section



object_type

list_of_fields: Note enclosed in parentheses ()

```
BRANCH (BusNumFrom,BusNameFrom,BusNumTo,BusNameTo,Circuit,Status,IsXF,R,X,B, // comment
// comments can appear here
LimitMVAA,LimitMVAB,LimitMVAC)
{
//--comments can appear here-----
// Num  Name      ToNum  ToName  CKT   Status  Xfmr?   R       X       B       Alimit  Blimit  Climit
//-----
1 "ROANSpra"  964 "GIBCRK C" " 1" "Closed" "No"  0.00690  0.03980  0.01080  236.0   236.0   236.0
1 "ROANSpra"   4 "IOLA" " 1" "Closed" "No"  0.00828  0.04776  0.01296  236.0   236.0   236.0
5 "IOLA"       4 "IOLA" " 1" "Closed" "Yes" 0.00763  0.14166  0.00000   60.0    60.0    60.0
5 "IOLA"      13 "KEITH" " 1" "Closed" "No"  0.13735  0.12184  0.00175   24.0    24.0    24.0
5 "IOLA"       9 "BEDIAS" " 1" "Closed" "No"  0.01980  0.11477  0.00194  118.0   118.0   118.0
9 "BEDIAS"    25 "NTHZULCH" " 1" "Closed" "No"  0.03173  0.18395  0.00312   88.0   118.0   118.0
13 "KEITH"    17 "CARLOSSW" " 1" "Closed" "No"  0.08233  0.07303  0.00105   24.0    24.0    24.0
17 "CARLOSSW" 29 "BOONVIL" " 1" "Closed" "No"  0.21753  0.19295  0.00277   24.0    24.0    24.0
17 "CARLOSSW" 21 "CARLOS" " 1" "Closed" "No"
// data can appear on more than one line. It just read a field for each entry in the list_of fields
0.06540  0.10810  0.00160   35.0    35.0    35.0
25 "NTHZULCH" 48 "HLTOPLKS" " 1" "Closed" "No"  0.04232  0.24532  0.00416   88.0   118.0   118.0
//-----
}
```

Start
and End
Data
Lists

Note: "GIBCRK C" needs double quotes because it has a space in the name
IOLA may optionally have quotes

DATA Section

SubData



- For some object_types, the DATA section does not provide an adequate method of defining information.
- For these special instances, SUBDATA sections are optionally added to each data list.

```
object_type (list_of_fields)
{
value_list_1
    <SUBDATA subobject_type1>
        precise format describing an object_type1
        precise format describing an object_type1
        :
    </SUBDATA>
    <SUBDATA subobject_type2>
        precise format describing an object_type2
        precise format describing an object_type2
        :
    </SUBDATA>
value_list_2
    :
value_list_n
}
```

DATA Section

SubData



- Starts with tag `<SUBDATA subobject_type>`
- Ends with tag `</SUBDATA>`
- Inside the tags, you must follow specified format for the respective `subobject_type`.
 - Values must appear in a precisely defined order
 - A list of the subdata sections for each kind of `object_type` follow
 - For detailed description of the precisely defined format for each subdata section, see the online help

SubData Sections allowed for each kind of object_type



- Contingency
 - CTGElement, LimitViol, and Sim_Solution_Options
- CTG_Options
 - Sim_Solution_Options
- Filter
 - Condition
- Gen
 - BidCurve and ReactiveCapability
- Load
 - BidCurve
- MultiSectionLine
 - Bus

SubData Sections allowed for each kind of object_type



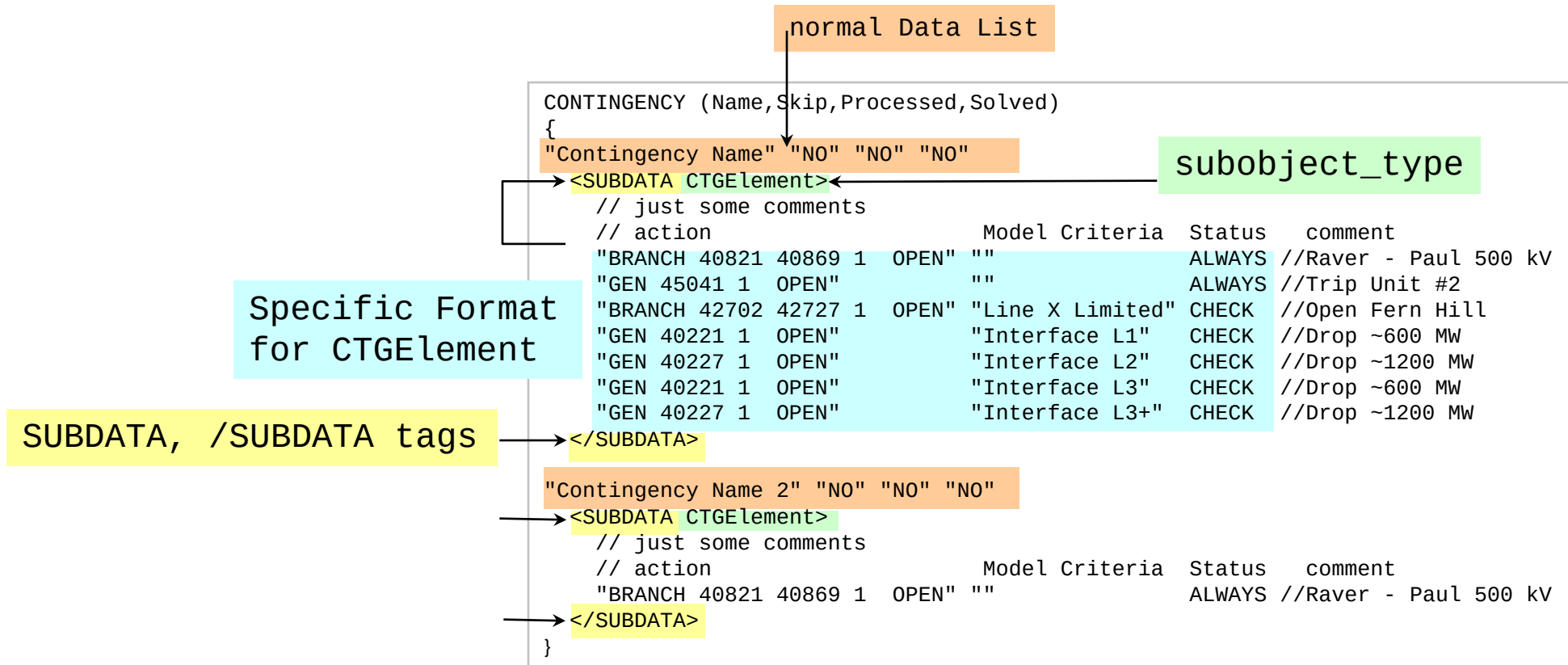
- InjectionGroup
 - PartPoint
- Interface
 - InterfaceElement
- ModelCondition
 - Condition
- ModelFilter
 - ModelCondition
- Owner
 - Bus, Load, Gen, Branch

SubData Sections allowed for each kind of object_type

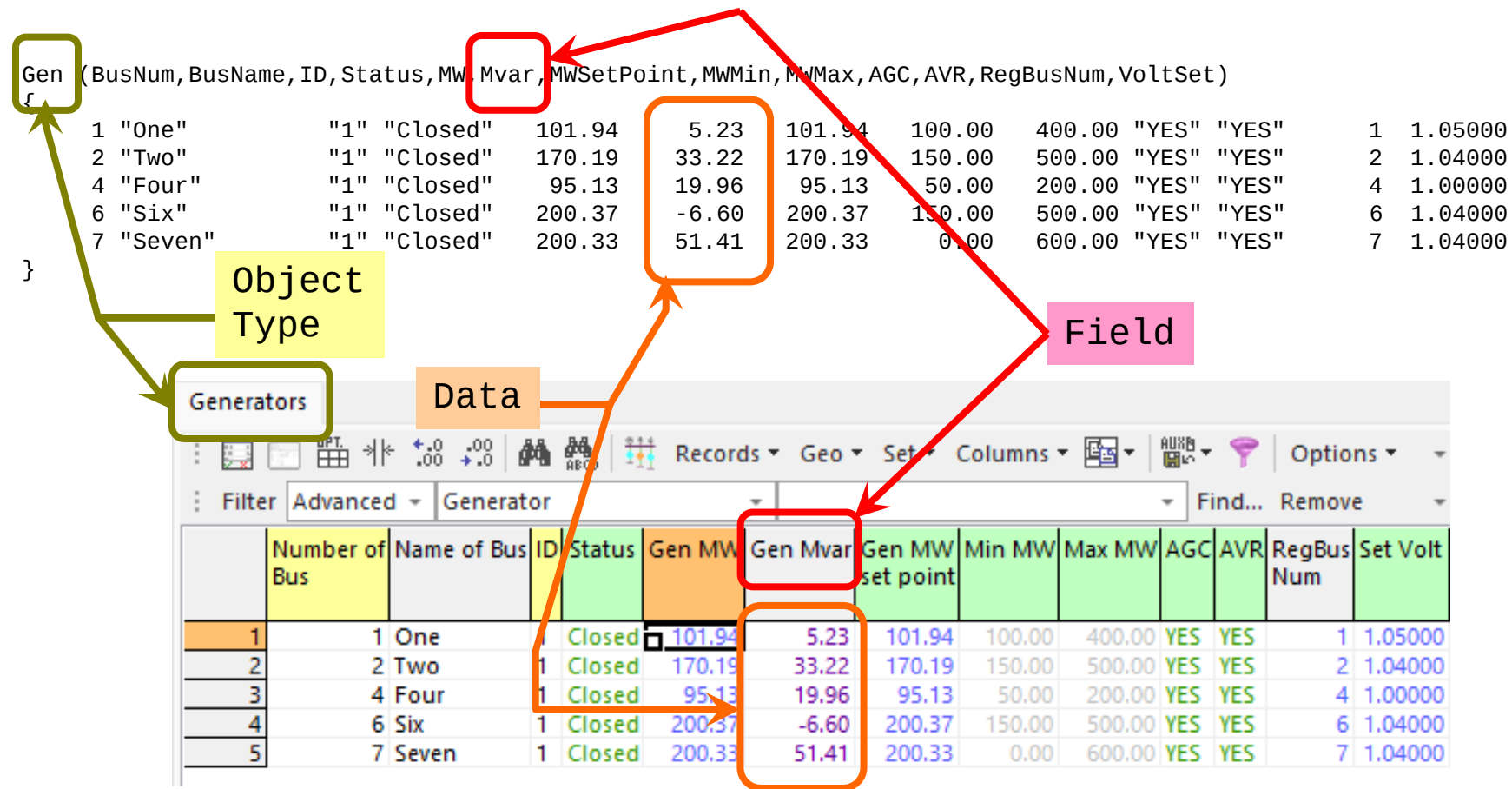


- DataGrid
 - ColumnInfo
- ATCScenario
 - TransferLimiter, ATCExtraMonitor
- ... and many more

Example DATA Section with SubData



Correlation of Case Info Display to an Auxiliary DATA Section

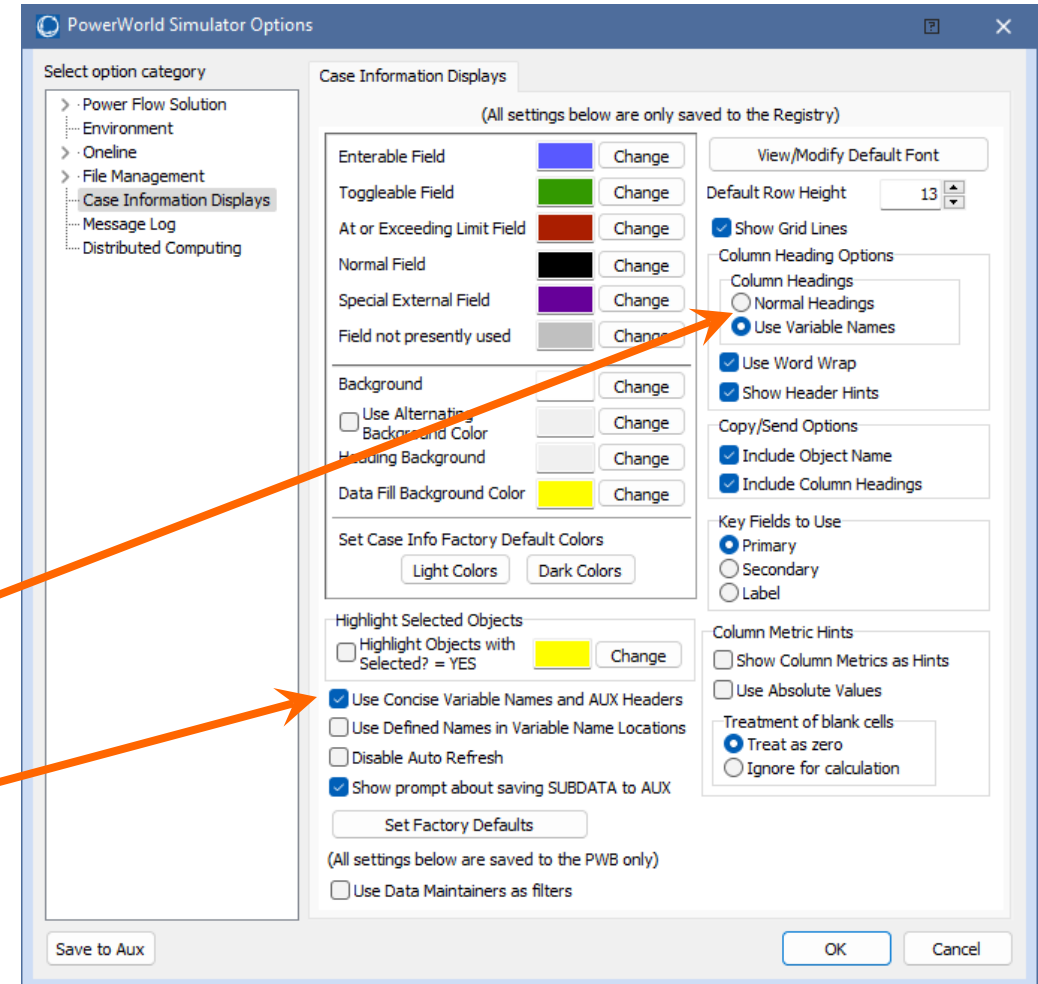


Option For changing Case Info Display Column Heading to Variables

- Choose **Options** → **Simulator Options**
 - Then go to the Case Information Displays option category
 - Change the **Column Headings** to *Use Variable Names*

*Use Normal Headings or
Variable Names*

*Use Concise Variable
Names and AUX Headers*



Option For changing Case Info Display Column Heading to Variables



- You can also change the option in the toolbar

The screenshot displays a software interface with a data table and an 'Options' menu. The table has columns: Number of Bus, Name of Bus, ID, Status, Gen MW, Gen Mvar, Gen MW set point, Min MW, Max MW, AGC, AVR, RegBus Num, Set Volt, and Gen Mvar set point. The 'Options' menu is open, showing 'Display/Column Options'. An orange arrow points from the text 'Options> Headings: choose Normal or Variable Names' to the 'Headings' dropdown menu, which is currently set to 'Normal'. Another orange arrow points from the text 'Use Concise Variable Names and Headers' to the 'Use Concise Variable Names and Headers' checkbox, which is checked.

Options> Headings: choose *Normal* or *Variable Names*

Use Concise Variable Names and Headers

Search Search Now Options ▾

170 MW

Options

Display/Column Options

Headings: Normal

☒ Show Grid Lines

Show Header Hints

☒ Word Wrap Headings

Default Row Height: 13

Highlight Row if Selected Field = YES

Highlight Select Color: Yellow

Zoom: 100%

Remove Trailing Zeros

Key Fields: Primary

Use Alternating Background Colors

☒ Use Concise Variable Names and Headers

Use Defined Names in Variable Name Locations

Use Data Maintainer Filtering

Show Column Metrics as Hints

Use Abs Value in Column Metrics Hints

Column Metrics Treat Blanks: Treat as Zero

Use Variable Names



- Provides a graphical way for you to see the variable names

Generators

Filter: Advanced Generator

	Number of Bus	Name of Bus	ID	Status	Gen MW	Gen Mvar	Min MW	Max MW	AGC	AVR	RegBus Num	Set Volt	Min Mvar	Max Mvar	Enforce MW Limits	Part. Factor	Cost Model
1	1	One	1	Closed	101.85	5.25	100.00	400.00	YES	YES	1	1.05000	-9900.00	9900.00	YES	1.00	Cubic
2	2	Two	1	Closed	170.08	33.24	150.00	500.00	YES	YES	2	1.04000	-9900.00	9900.00	YES	1.00	Cubic
3	4	Four	1	Closed	95.03	19.99	50.00	200.00	YES	YES	4	1.00000	-9900.00	9900.00	YES	1.00	Cubic
4	6	Six	1	Closed	200.33	-6.59	150.00	500.00	YES	YES	6	1.04000	-9900.00	9900.00	YES	1.00	Cubic
5	7	Seven	1	Closed	200.65	51.29	0.00	600.00	YES	YES	7	1.04000	-99999.00	99999.00	YES	1.00	Cubic

Generators

Filter: Advanced Generator

Gen	BusNum(<)	BusName(<)	ID(<)	Status(<)	MW(<)	Mvar(<)	MWMin(<)	MWMax(<)	AGC(<)	AVR(<)	RegBusNum(<)	VoltSet(<)	MvarMin(<)	MvarMax(<)	EnforceMWLimit(<)	PartFact(<)	CostModel(<)
1	1	One	1	Closed	101.85	5.25	100.00	400.00	YES	YES	1	1.05000	-9900.00	9900.00	YES	1.00	Cubic
2	2	Two	1	Closed	170.08	33.24	150.00	500.00	YES	YES	2	1.04000	-9900.00	9900.00	YES	1.00	Cubic
3	4	Four	1	Closed	95.03	19.99	50.00	200.00	YES	YES	4	1.00000	-9900.00	9900.00	YES	1.00	Cubic
4	6	Six	1	Closed	200.33	-6.59	150.00	500.00	YES	YES	6	1.04000	-9900.00	9900.00	YES	1.00	Cubic
5	7	Seven	1	Closed	200.65	51.29	0.00	600.00	YES	YES	7	1.04000	-99999.00	99999.00	YES	1.00	Cubic

```

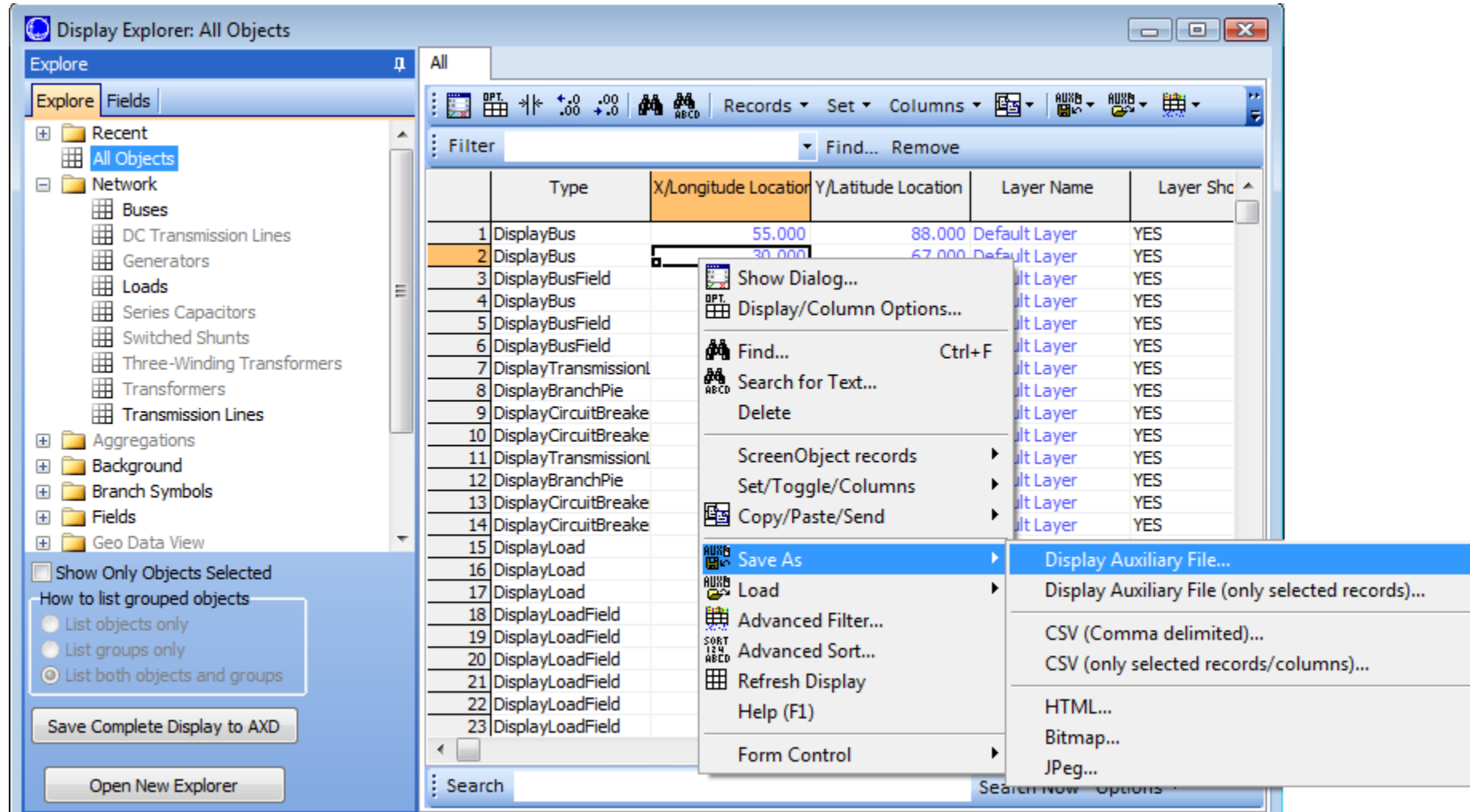
Gen (BusNum, BusName, ID, Status, MW, Mvar, MWMin, MWMax, AGC, AVR, RegBusNum, VoltSet, MvarMin, MvarMax, EnforceMWLimit, PartFact, CostModel)
{
  1 "One" "1" "Closed" 101.85 5.25 100.00 400.00 "YES" "YES" 1 1.05000 -9900.00 9900.00 "YES" 1.00 "Cubic"
  2 "Two" "1" "Closed" 170.08 33.24 150.00 500.00 "YES" "YES" 2 1.04000 -9900.00 9900.00 "YES" 1.00 "Cubic"
  4 "Four" "1" "Closed" 95.03 19.99 50.00 200.00 "YES" "YES" 4 1.00000 -9900.00 9900.00 "YES" 1.00 "Cubic"
  6 "Six" "1" "Closed" 200.33 -6.59 150.00 500.00 "YES" "YES" 6 1.04000 -9900.00 9900.00 "YES" 1.00 "Cubic"
  7 "Seven" "1" "Closed" 200.65 51.29 0.00 600.00 "YES" "YES" 7 1.04000 -99999.00 99999.00 "YES" 1.00 "Cubic"
}
    
```

Display Auxiliary Files



- One-line diagram objects can also be saved and edited in the auxiliary file format (*.axd file extension).
- Case Information Displays for one-line diagrams are accessed from **Onelines**
→ **List Display** → **All Display Objects...**
- Save entire one-line in axd format
 - **File** → **Save Online As...**
 - Choose *Display Auxiliary File (*.axd)* from **Save as type:**
- Open entire one-line in axd format
 - **File** → **Open Online...**
 - Choose *Display Auxiliary File* from **Files of type:**
- **Auxiliary ID** (*SOAuxiliaryID*) field used as an extra key field to uniquely identify objects

Display Objects



Example Display Auxiliary DATA Section



Object Type for this example is DisplayBus

↓

```
DISPLAYBUS (BusNum, SOAuxiliaryID, SOX, SOY, SOTickness, SOColor, SOUseFillColor, SOFillColor,
             SOSize, SOWidth, SOorientation, SOLevel, SOSameLevelDisplayPriority, SOImmobile,
             SLName, SOStyle, SODashed, SOBelongsToGroup)
{
    1 "1"  9.000000000 63.000000000      1 0 "YES"      0    6.00    0.45 "Right" "Middle" 132 "NO "
    "Default Layer" "Rectangle"          0 ""
    2 "1" 19.000000000 42.000000000      1 0 "YES"      0   11.00    0.45 "Right" "Middle" 131 "NO "
    "Default Layer" "Rectangle"          0 ""
    3 "1" 37.000000000 63.000000000      1 0 "YES"      0    5.00    0.45 "Right" "Middle" 130 "NO "
    "Default Layer" "Rectangle"          0 ""
    4 "1" 72.000000000 57.000000000      1 0 "YES"      0    9.00    0.45 "Right" "Middle" 129 "NO "
    "Default Layer" "Rectangle"          0 ""
    5 "1" 68.000000000 36.000000000      1 0 "YES"      0    6.00    0.45 "Right" "Middle" 128 "NO "
    "Default Layer" "Rectangle"          0 ""
    6 "1" 15.000000000 15.750000000      1 0 "YES"      0    8.00    0.45 "Right" "Middle" 121 "NO "
    "Default Layer" "Rectangle"          0 ""
    7 "1" 54.000000000 15.750000000      1 0 "YES"      0   11.00    0.45 "Right" "Middle" 119 "NO "
    "Default Layer" "Rectangle"          0 ""
}
```

Display Auxiliary File Example



- Case *ACTIVSg10kOPF.raw*
- *I04_Auxiliary File Data Section\WABus.axd*: adds state background and several 345 kV buses and transmission lines
- Load axd into new or existing one-line diagram

```
DisplayTransmissionLine
(BusNum,BusNum:1,LineCircuit,SOAuxiliaryID, ...)
{
  10586    10588 " 1" "1" ...
    <SUBDATA Line>
    // Coordinates are Mercator
    [-119.28191216 47.19970512] [-119.28011216 47.19970512]
    [-119.28011216 47.19970512] [-119.27111216 47.19970512]
  </SUBDATA>
  ...
}
```

Case Information Ribbon Tab:

Auxiliary File Export Format Descriptions



- Allows you to define a list of DATA sections you would like to write out to an Auxiliary File
 - Object Type: Specify type of object
 - Filter Name: All, Selected, AreaZone, or the name of an Advanced Filter in quotes. Allows you to specify objects for which to write data.
 - Fields: a list of all fields to be written for this object
 - SubData: a list of sub-data sections to write for each object
- You can then use a script command to invoke saving all this information
 - `SaveDataUsingExportFormat("filename", filetype, "FormatName");`

Case Information Ribbon Tab: Auxiliary File Export Format Descriptions Dialog

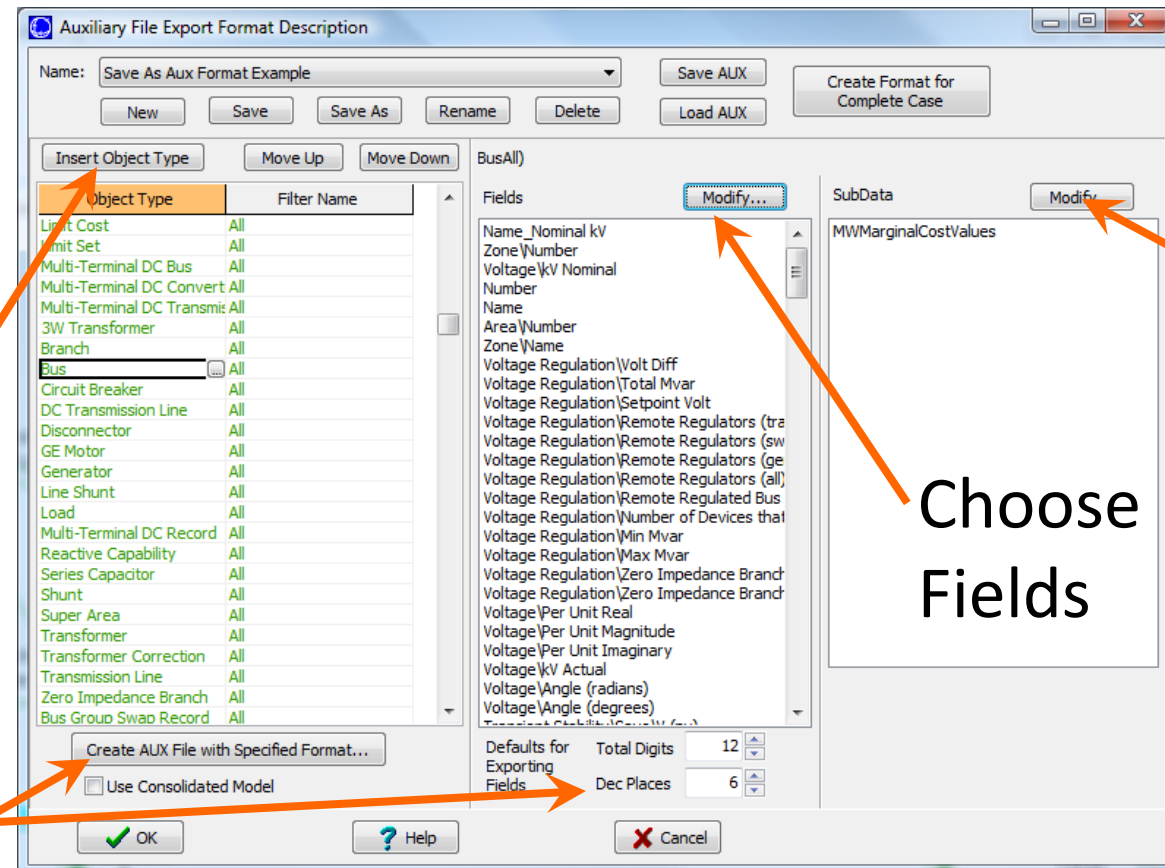


- On the Case Information Ribbon Tab, click AUX Export Format Desc...



Choose Object Types

Set formatting and file type



Choose
SubData

Choose
Fields



Special Object Types to Provide Better Access to SUBDATA Objects



- Previously, several types of data were only accessible through SUBDATA sections of AUX Files
- This limited your ability to systematically edit/query these objects
 - Could not use Copy/Paste To/From a Spreadsheet
 - Could not use SimAuto commands to access
 - Could not use Script Commands (SaveData, SetData)
- Most of these limitations have now been removed for most object types
 - New object types have been created that provide access to this data in “sub-tables”
 - [in “database speak” = this is table with a foreign key constraint]

Special Object Types Created to Provide Access to SUBDATA



New Object Type Created	Existing Object Type this SUBDATA is related to	Existing SUBDATA Section <SUBDATA XXX>
ReactiveCapability	Gen	ReactiveCapability
PartPoint	InjectionGroup	PartPoint
InterfaceElement	Interface	InterfaceElement
MTDCBus	MTDCRecord	MTDCBus
MTDCConverter	MTDCRecord	MTDCConverter
MTDCTransmissionLine	MTDCRecord	MTDCTransmissionLine
ContingencyElement	Contingency	CTGElement
ViolationCTG	Contingency	LimitViol
CTGElementBlockElement	CTGElementBlock	CTGElement
GlobalContingencyActionsElement	GlobalContingencyActions	CTGElement
PostPowerFlowActionsElement	PostPowerFlowActions	CTGElement

Interface / Interface Element In Auxiliary Files



Interface Objects with <SUBDATA InterfaceElement>

```
INTERFACE (Number, Name, MonDirection,  
          LimitMWA, LimitMWB, LimitMWC)  
{  
1 "East Side" "FROM -> TO" 0.0 0.0 0.0  
  <SUBDATA InterfaceElement>  
    "BRANCH 4 2 1" NO 0.500000  
    "BRANCH 4 3 1" NO 0.600000  
    "BRANCH 4 5 1" NO 0.700000  
  </SUBDATA>  
2 "West Side" "FROM -> TO" 0.0 0.0 0.0  
  <SUBDATA InterfaceElement>  
    "BRANCH 2 5 1" NO 0.800000  
    "BRANCH 2 4 1" NO 0.900000  
  </SUBDATA>  
}
```

Interface Objects and InterfaceElement Objects

```
INTERFACE (Number, Name, MonDirection,  
          LimitMWA, LimitMWB, LimitMWC)  
{  
1 "East Side" "FROM -> TO" 0.0 0.0 0.0  
2 "West Side" "FROM -> TO" 0.0 0.0 0.0  
}  
INTERFACEELEMENT (InterfaceName, Element,  
                  MeterFar, Weight)  
{  
"East Side" "BRANCH 4 2 1" "NO" 0.50  
"East Side" "BRANCH 4 3 1" "NO" 0.60  
"East Side" "BRANCH 4 5 1" "NO" 0.70  
"West Side" "BRANCH 2 5 1" "NO" 0.80  
"West Side" "BRANCH 2 4 1" "NO" 0.90  
}
```

These Auxiliary Files represent EXACTLY the same data

Interface / Interface Element In an Excel Spreadsheet



Microsoft Excel - Book2

Type a question for help

Arial 10 B I U

G13 fx

	A	B	C	D	E	F	G
1							
2		Interface					
3		Number	Name	Monitor Direction	Lim A MW	Lim B MW	Lim C MW
4		1	East Side	FROM -> TO	0	0	0
5		2	West Side	FROM -> TO	0	0	0
6							
7		InterfaceElement					
8		Interface Name	Description (File Format)	Metered Far End	Weighting		
9		East Side	BRANCH 4 2 1	NO	0.5		
10		East Side	BRANCH 4 3 1	NO	0.6		
11		East Side	BRANCH 4 5 1	NO	0.7		
12		West Side	BRANCH 2 5 1	NO	0.8		
13		West Side	BRANCH 2 4 1	NO	0.9		
14							

InterfaceElement

Ready NUM

Injection Group / PartPoint In Auxiliary Files



Injection Group Objects with <SUBDATA PartPoint>

```
INJECTIONGROUP (Name)
{
  "Big Gens"
    <SUBDATA PartPoint>
      "GEN" 1 "1" 1.0000 "SPECIFIED" "YES"
      "GEN" 2 "1" 1.0000 "SPECIFIED" "YES"
      "GEN" 4 "1" 1.0000 "SPECIFIED" "YES"
    </SUBDATA>
  "Little Gens"
    <SUBDATA PartPoint>
      "GEN" 6 "1" 1.0000 "SPECIFIED" "YES"
      "GEN" 7 "1" 1.0000 "SPECIFIED" "YES"
    </SUBDATA>
}
```

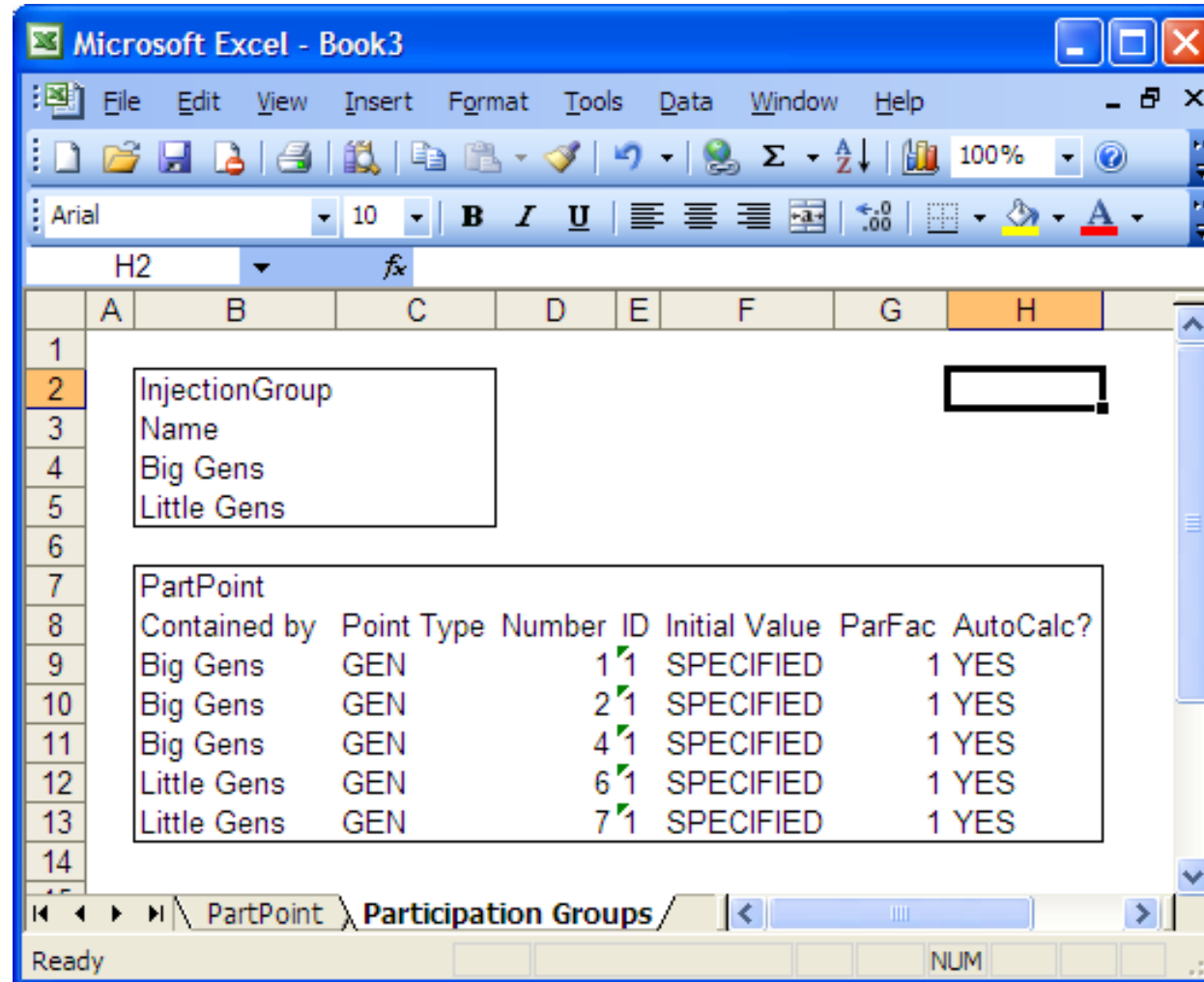
Injection Group Objects and PartPoint Objects

```
INJECTIONGROUP (Name)
{
  "Big Gens"
  "Little Gens"
}

PARTPOINT (GroupName, Object, AutoCalcMethod,
           PartFact, AutoCalc)
{
  "Big Gens"      "GEN 1 1" "SPECIFIED" 1.00 "YES"
  "Big Gens"      "GEN 2 1" "SPECIFIED" 1.00 "YES"
  "Big Gens"      "GEN 4 1" "SPECIFIED" 1.00 "YES"
  "Little Gens"   "GEN 6 1" "SPECIFIED" 1.00 "YES"
  "Little Gens"   "GEN 7 1" "SPECIFIED" 1.00 "YES"
}
```

These Auxiliary Files represent EXACTLY the same data

Injection Group / PartPoint In an Excel Spreadsheet



The screenshot shows a Microsoft Excel window titled "Microsoft Excel - Book3". The spreadsheet has columns A through H and rows 1 through 15. The "InjectionGroup" data is in cells B2:B5, and the "PartPoint" data is in cells B7:B13. A table border is drawn around the "PartPoint" data, and a formula bar is visible in cell H2.

InjectionGroup	PartPoint
Name	
Big Gens	
Little Gens	

PartPoint

Contained by	Point Type	Number	ID	Initial Value	ParFac	AutoCalc?
Big Gens	GEN	1	1	SPECIFIED	1	YES
Big Gens	GEN	2	1	SPECIFIED	1	YES
Big Gens	GEN	4	1	SPECIFIED	1	YES
Little Gens	GEN	6	1	SPECIFIED	1	YES
Little Gens	GEN	7	1	SPECIFIED	1	YES

PartPoint Participation Groups



Contingency / ContingencyElement In Auxiliary Files



Contingency Objects with <SUBDATA CTGElement>

```
CONTINGENCY (Name)
{
  "West Outage"
    <SUBDATA CTGElement>
      "BRANCH 2 5 1 OPEN" "" CHECK
      "BRANCH 2 4 1 OPEN" "" CHECK
    </SUBDATA>
  "East Outage"
    <SUBDATA CTGElement>
      "GEN 2 MOVE_P_TO 4 50 PERCENT" "" CHECK
      "BRANCH 2 5 1 OPEN" "" CHECK
      "BRANCH 4 5 1 OPEN" "" CHECK
    </SUBDATA>}
}
```

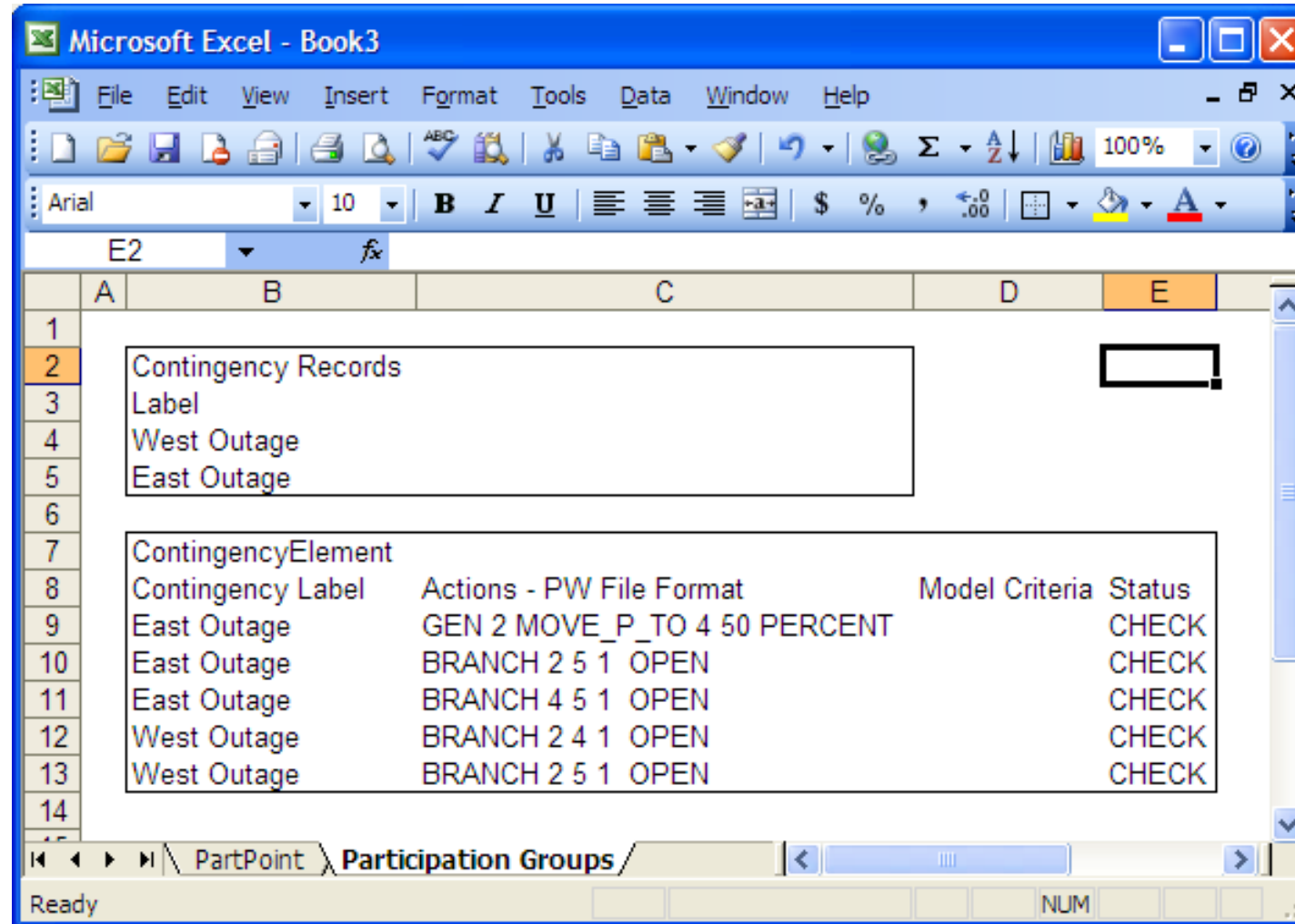
Contingency Objects and Contingency Element Objects

```
CONTINGENCY (Name)
{
  "West Outage"
  "East Outage"
}

CONTINGENCYELEMENT (Name, Object, Action,
                    Criteria, CriteriaStatus)
{
  "East Outage" "GEN 2" " MOVE_P_TO 4 50 PERCENT" "" "CHECK"
  "East Outage" "BRANCH 2 5 1" "OPEN" "" "CHECK"
  "East Outage" "BRANCH 4 5 1" "OPEN" "" "CHECK"
  "West Outage" "BRANCH 2 4 1" "OPEN" "" "CHECK"
  "West Outage" "BRANCH 2 5 1" "OPEN" "" "CHECK"
}
```

These Auxiliary Files represent EXACTLY the same data

Contingency / ContingencyElement In an Excel Spreadsheet



The screenshot shows a Microsoft Excel window titled "Microsoft Excel - Book3". The spreadsheet has columns A through E and rows 1 through 15. The data is organized into two main sections: "Contingency Records" and "ContingencyElement".

Contingency Records

Label
West Outage
East Outage

ContingencyElement

Contingency Label	Actions - PW File Format	Model Criteria	Status
East Outage	GEN 2 MOVE_P_TO 4 50 PERCENT		CHECK
East Outage	BRANCH 2 5 1 OPEN		CHECK
East Outage	BRANCH 4 5 1 OPEN		CHECK
West Outage	BRANCH 2 4 1 OPEN		CHECK
West Outage	BRANCH 2 5 1 OPEN		CHECK

The status bar at the bottom indicates "Ready" and "NUM".



Special Support Sub-Types of Special Object Types



- These special object types give you access to a combination of all the “sub-object” fields and specific device fields simultaneously

New Object Type	PartPoint	InterfaceElement
Special Sub-Types which provide access to two tables simultaneously	PartPointGen PartPointInjectionGroup PartPointLoad PartPointShunt	InterfaceElementBranch InterfaceElementBranchClose InterfaceElementBranchOpen InterfaceElementDCLine InterfaceElementGen InterfaceElementInjectionGroup InterfaceElementInterface InterfaceElementLoad InterfaceElementMSLine



Example: Special Support Sub-Types of Special Object Types



- PartPointGen
 - Provides access to all PartPoints that represent generator objects
 - Provides access to BOTH the list of fields for the PartPoint AND the Gen
- InterfaceElementBranch
 - Provides access to all InterfaceElements that represent branch objects
 - Provides access to BOTH the list of fields for the InterfaceElement AND the Branch



Special Fields Provide Access to SUBDATA



- ATCSenario object with <SUBDATA TransferLimiter>
 - Instead of creating new object type, extra key fields are added to existing TransferLimiter object type
 - ATCLineZoneChanges, ATCGenChanges, ATCInterfaceChanges
 - When fields are present, records are associated with the ATC scenario identified by the extra key fields
 - When fields are not present, records are associated with the standard single set of ATC results