

Auxiliary File and Scripting Tips



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

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

Available Online Help

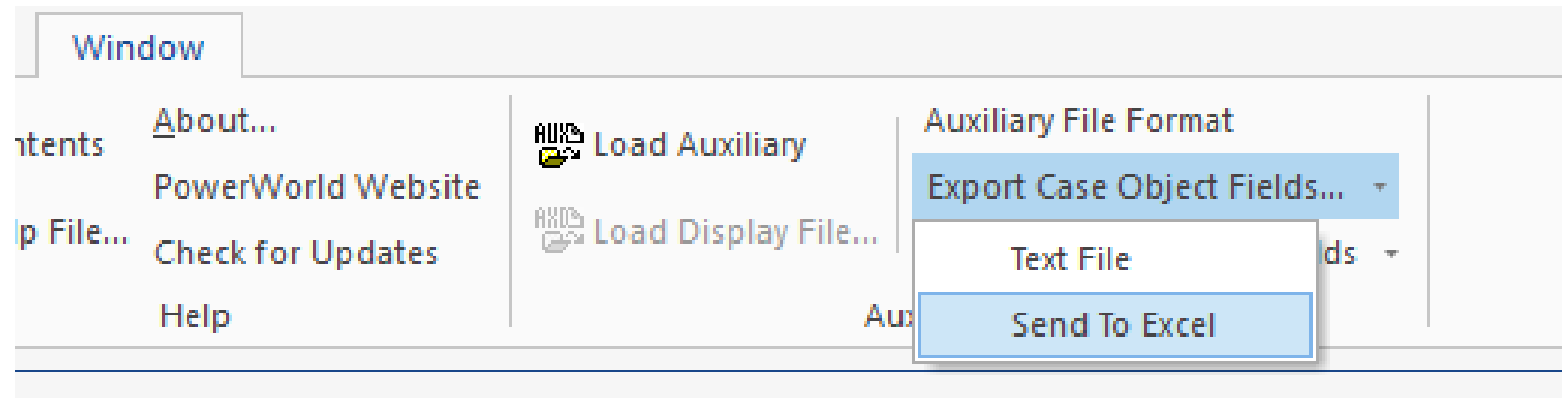


- <https://www.powerworld.com/WebHelp/Content/Other Documents/Auxiliary-File-Format.pdf>
 - Most up-to-date description of all available script commands and other auxiliary file related information
- <https://www.powerworld.com/WebHelp/>
 - Most up-to-date help information for all of Simulator
 - Contains SimAuto function documentation

Documentation: Export Case Object Fields



- The only comprehensive documentation about all of the object types and fields that are available
- From the **Window** Ribbon → **Export Case Object Fields...** →
 - **Text File**, or
 - **Send to Excel**



Export Case Object Fields (Excel)



< Included in Difference Case comparison

* Required to create object
1, 2, 3 Primary key
A, B, C Secondary key

For most fields, the description found here is the most comprehensive

	A	B	C	D	E	F	
1	Object Type	SUBDATA Allowed	Key/Required	Variable Name	Concise Variable Name	Type of Variable	Description
19011	Gen						
19012		BidCurve					
19013		ReactiveCapability					
19014				ABCPhaseAngle	FaultCurAngleA	Real	Phase A
19015				ABCPhaseAngle:1	FaultCurAngleB	Real	Phase B
19016				ABCPhaseAngle:2	FaultCurAngleC	Real	Phase C
19017				ABCPhaseI	FaultCurMagA	Real	Phase A
19018				ABCPhaseI:1	FaultCurMagB	Real	Phase B
19019				ABCPhaseI:2	FaultCurMagC	Real	Phase C
19020			<	AllLabels	AllLabels	String	This represents a comma-separated list
19021			<	AreaName	AreaName	String	It is possible for the terminal bus to bel
19022			<	AreaName:1	BusAreaName	String	It is possible for the terminal bus to bel
19023			<	AreaNum	AreaNumber	Integer	It is possible for the terminal bus to bel
19024			<	AreaNum:1	BusAreaNumber	Integer	It is possible for the terminal bus to bel
19025			<	BAName	BAName	String	It is possible for the terminal bus to bel
19026			<	BAName:1	BusBAName	String	It is possible for the terminal bus to bel
19027			<	BANumber	BANumber	Integer	It is possible for the terminal bus to bel
19028			<	BANumber:1	BusBANumber	Integer	It is possible for the terminal bus to bel
19029			<	BreakerDelay	BreakerDelay	Real	Breaker time delay in seconds
19030				BreakerGroupNum	BreakerGroupNum	Integer	ID of the Bus's breaker group of the bus
19031				BusCat	BusCat	String	Shows how the bus is being modeled in
19032				BusgenericSensP	SensdValuedPinj	Real	Sensitivity: Injection dValue/dP of Bus
19033				BusgenericSensQ	SensdValuedQinj	Real	Sensitivity: Injection dValue/dQ of Bus
19034				BusGenericSensV	SensdValuedVset	Real	Sensitivity: Injection dValue/dVsetpoin
19035				BusKVActual	kV	Real	Voltage: kV Actual of the bus
19036				BusLossSensMW	LossSensMW	Real	Sensitivity of the MW losses with respe
19037				BusMCMW	BusMargCostMW	Real	OPF: Marginal MW Cost. May be interpr
19038			<	BusName	BusName	String	Name of the bus
19039			*A*<	BusName_NomVolt	BusNameNomkV	String	Name_Nominal kV of the bus
19040			<	BusNomVolt	NomkV	Real	The nominal kv voltage specified as part
19041			*1*<	BusNum	BusNum	Integer	Number of the bus
19042				BusObjectOnline	Online	String	YES only if the generator Status = CLOSE
19043				BusOwnerName	BusOwnerName	String	Name of the Owner of the attached bus

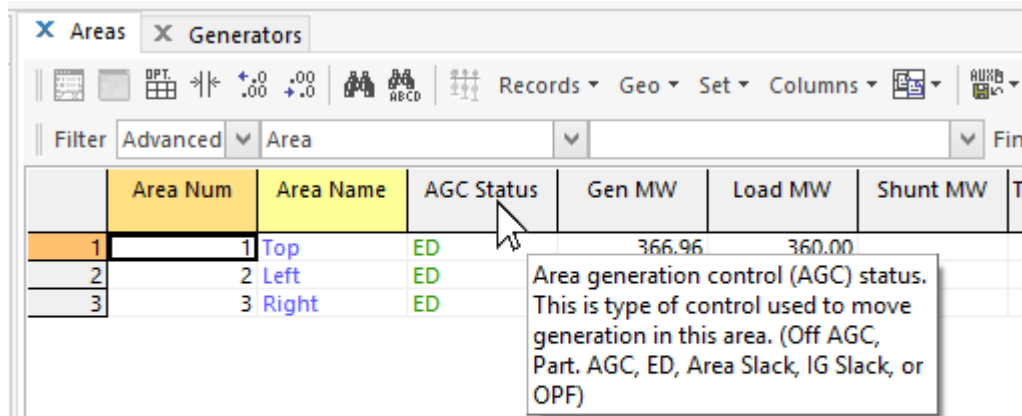
Export Case Object Fields



- SimAuto functions for obtaining field details
 - GetFieldList(ObjectType)
 - Return all fields and details for a particular object type
 - GetSpecificFieldList(ObjectType, FieldList)
 - Return specified fields and details for a particular object type

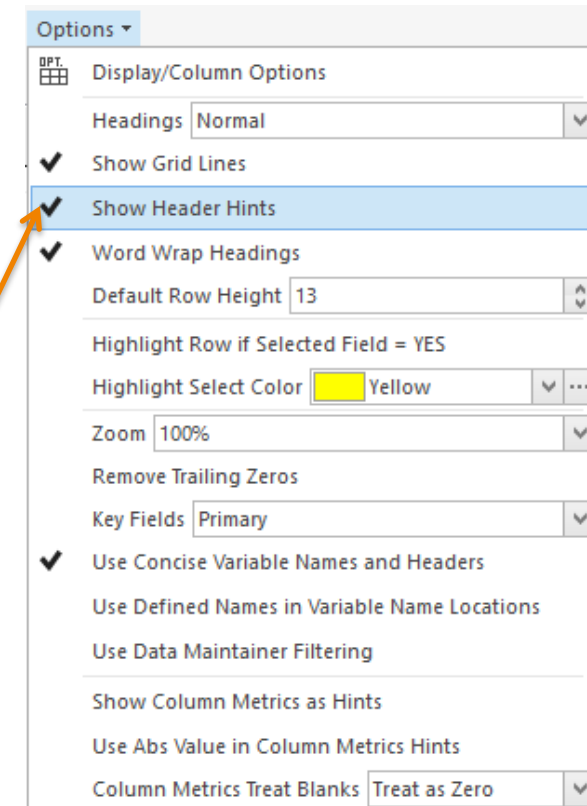
Hover Hints

- Hover over column headers in case information displays to see the description of fields



	Area Num	Area Name	AGC Status	Gen MW	Load MW	Shunt MW	To
1	1	Top	ED	366.96	360.00		
2	2	Left	ED				
3	3	Right	ED				

Hints can be enabled/disabled by toggling **Show Header Hints** option



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 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

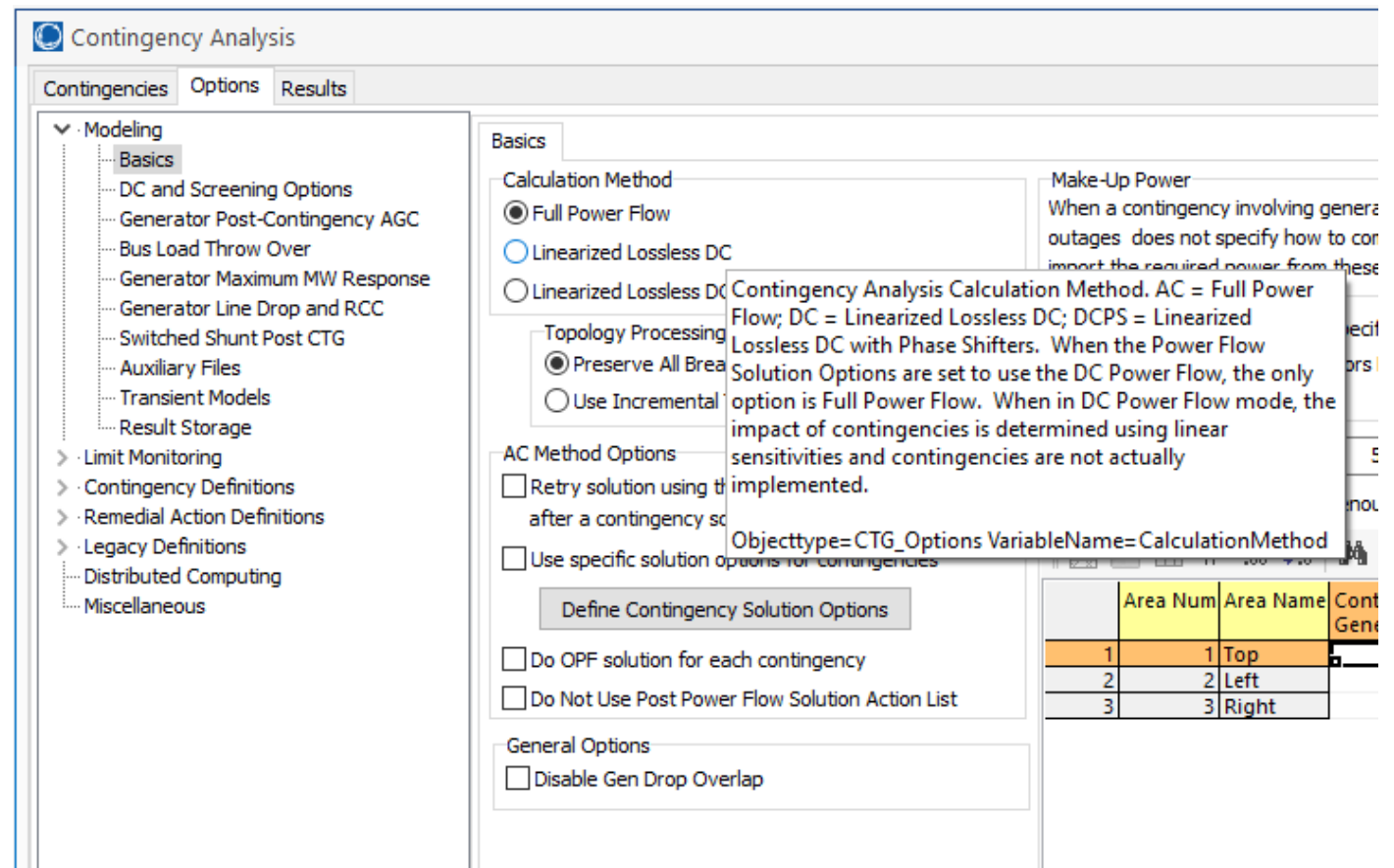
Column Metrics Treat Blanks Treat as Zero ▾



Hover Hints



- Most dialogs have hints that pop up when hovering over option fields
- These hints provide the object type and variable name for setting this option in an auxiliary file data section or script command



Specifying Field Variable Names



- Variable Names are used within auxiliary file DATA and SCRIPT sections to identify fields
- For help in creating auxiliary files, column headers within the Model Explorer can be set to show the variable names instead

The screenshot shows a table titled 'Buses' with the following columns: Bus, Number (1*<), Name (*<), AreaName (<), CustomFloat:1 (<), CustomFloat:2 (<), CustomFloat:0 (<), and NomkV*<. The table contains 7 rows of data. The 'Options' menu is open, showing 'Display/Column Options'. The 'Headings' dropdown is set to 'Variable Names'. The 'Columns' dropdown is also visible in the table header.

Bus	Number (1*<)	Name (*<)	AreaName (<)	CustomFloat:1 (<)	CustomFloat:2 (<)	CustomFloat:0 (<)	NomkV*<
1	1	One	Top				138.00
2	2	Two	Top				138.00
3	3	Three	Top				138.00
4	4	Four	Top				138.00
5	5	Five	Top				138.00
6	6	Six	Left				138.00
7	7	Seven	Right				138.00

Specifying Field Variable Names



- Variables follow the naming convention `variablename:location` where `location` is an integer indicating the exact variable to be used
- This convention is considered the legacy convention where the same `variablename` was re-used by tacking on the `location`
- Starting with Simulator version 19, *concise* variable names have been created in an attempt to get rid of the `location` in as many variables as possible and to make the names more obvious

Specifying Field Variable Names

Legacy vs. Concise



- The legacy variables to indicate the MW flow at the from end and to ends of a branch are:
 - `LineMW:0` (:0 is typically omitted because :0 is assumed if no location specified)
 - `LineMW:1`
- The same variables in the concise format are:
 - `MWFrom`
 - `MWTo`

Specifying Field Variable Names

Legacy vs. Concise



- Both versions of a field variable name can be shown in the Model Explorer

The screenshot shows the Model Explorer interface with a table of bus data. The table has columns: Bus, Number (1*<), Name (*<), AreaName (<), CustomFloat:1 (<), CustomFloat:2 (<), CustomFloat:0 (<), and NomkV*<. The data rows are numbered 1 through 7, with names One through Seven and area names Top or Right. The NomkV values are all 138.00. The Options menu is open, showing various settings. The 'Use Concise Variable Names and Headers' option is checked and highlighted with a blue bar. An orange arrow points from the text below to this option.

Bus	Number (1*<)	Name (*<)	AreaName (<)	CustomFloat:1 (<)	CustomFloat:2 (<)	CustomFloat:0 (<)	NomkV*<
1	1	One	Top				138.00
2	2	Two	Top				138.00
3	3	Three	Top				138.00
4	4	Four	Top				138.00
5	5	Five	Top				138.00
6	6	Six	Left				138.00
7	7	Seven	Right				138.00

Options

- Display/Column Options
- Headings Variable Names
- ☒ 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 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

Toggle option to switch between legacy and concise. In Simulator version 20 this is set to show concise names by default.

Specifying Field Variable Names



- Location indicators continue to exist for some variable names
- Typically these are fields for which a dynamic number exists
 - Fields that are available for most objects
 - CustomExpression, CustomFloat, CustomInteger, CustomString, CustomExpressionStr, CalcFied, DataCheck, and DataCheckAggr
 - Some example fields that are available for specific objects
 - Bus – MargCostMWValue, MargControl, SensdValuedPinjMult
 - Branch – PTDFMult, LODFMult, LODFInterfaceMult
 - Gen – BidMW, BidMWHr, SensMultMeterMultControl

Specifying Field Variable Names



- To better identify dynamic fields, the `Location` can be replaced with `Location_by_name` for fields that have user-defined names
- This could help prevent conflicts in data where the same `Location` by number is being referenced in different auxiliary files but these should really be different fields
- The fields can be read and written by Simulator when used in auxiliary files
 - `CustomExpression`, `CustomFloat`, `CustomInteger`, `CustomString`, `CustomExpressionStr`, `CalcFied`, `DataCheck`, and `DataCheckAggr`
 - `"CustomFloat:My Header"`
 - `"CalcField:My calculated field name"`
 - `"DataCheckAggr:Bus 'DataCheck Name'"`

Specifying Field Variable Names



Buses

Filter: Advanced Bus Find... Remove Quick Filter

Bus	Number (1*)	Name (*)	AreaName (<)	CustomFloat:Another Header (<)	CustomFloat:Extra Header (<)	CustomFloat:My H
1	1	One	Top	1.000	11.000	
2	2	Two	Top	2.000	12.000	
3	3	Three	Top	3.000	13.000	
4	4	Four	Top	4.000	14.000	
5	5	Five	Top	5.000	15.000	
6	6	Six	Left	6.000	16.000	
7	7	Seven	Right	7.000	17.000	

Options

Display/Column Options

Headings: Variable Names

- ☒ 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 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

Bus (Number, Name, AreaName, CustomFloat:Another Header, CustomFloat:Extra Header, CustomFloat:My Header, NomKV, Vpu, kV, Vangle, LoadMW, LoadMvar, GenMW, GenMvar, ShuntMvar, Acto, Acto, AreaNumber, ZoneNumber)

```
{
  1 "One"      "Top"      1.000  11.000  21.000  138.00  1.05000  144.900  6.09 "" "" 101.85  5.25 "" 0.00  0.00  1  1
  2 "Two"      "Top"      2.000  12.000  22.000  138.00  1.04000  143.520  4.22  40.00  20.00  170.08  33.24 "" 0.00  0.00  0.00  1  1
  3 "Three"    "Top"      3.000  13.000  23.000  138.00  0.99269  136.991  0.99  110.00  40.00 "" "" 0.00  0.00  1  1
  4 "Four"     "Top"      4.000  14.000  24.000  138.00  1.00000  138.000  1.46  80.00  30.00  95.03  19.99 "" 0.00  0.00  0.00  1  1
  5 "Five"     "Top"      5.000  15.000  25.000  138.00  1.00665  138.917  -0.83 130.00  40.00 "" "" 0.00  0.00  1  1
  6 "Six"      "Left"     6.000  16.000  26.000  138.00  1.04000  143.520  2.81  200.00  0.00  200.33  -6.59 "" 0.00  0.00  2  1
  7 "Seven"    "Right"    7.000  17.000  27.000  138.00  1.04000  143.520  0.00  200.00  0.00  200.65  51.29 "" 0.00  0.00  3  1
}
```

Toggle Use Defined Names in Variable Name Locations to show the user-defined name within the case info grid and to use this when saving an auxiliary file

Specifying Field Variable Names

Saving All Fields for a Variable



- What if you don't know how many fields exist for a particular variable name?
 - Multiple PTDF or LODF calculation produces a matrix of results based on the inputs selected
- Replace the `Location` identifier with the keyword `ALL`
 - `PTDFMult:ALL` or `LODFMult:ALL`
- Available with script commands that allow specifying fields to save to file
 - `CTGSaveViolationMatrices`, `SaveData`, `SaveDataWithExtra`, `SaveObjectFields`, and `SendToExcel`

Specifying Field Variable Names

Saving All Fields for a Variable



- Available with SimAuto functions that get fields
 - GetParametersMultipleElement, GetParametersMultipleElementFlatOutput, GetParametersSingleElement, GetSpecificFieldList, SendToExcel, and WriteAuxFile
- Save the LODF matrix results to a CSV file

```
SaveData("myfile.CSV", CSVColHeader, Branch, [BusNumFrom, BusNumTo,  
Circuit, LODFMult:ALL], [], , [], NO);
```

Hint: use `filetype = CSVColHeader` to identify the monitored branches by the normal names instead of variable names

Specifying Field Variable Names

Saving All Fields for an Object Type



- What if you just want everything?
- Replace entire list of fields with keyword ALL
- Available with script commands that allow specifying fields to save to file
 - CTGSaveViolationMatrices, SaveData, SaveDataWithExtra, SaveObjectFields, and SendToExcel
- Available with SimAuto functions that get fields
 - GetParametersMultipleElement, GetParametersMultipleElementFlatOutput, GetParametersSingleElement, GetSpecificFieldList, SendToExcel, and WriteAuxFile

Specifying Field Variable Names

Saving All Fields for an Object Type



- Available with SimAuto functions that get fields
 - GetParametersMultipleElement, GetParametersMultipleElementFlatOutput, GetParametersSingleElement, GetSpecificFieldList, SendToExcel, and WriteAuxFile
- Saving all fields for a Branch

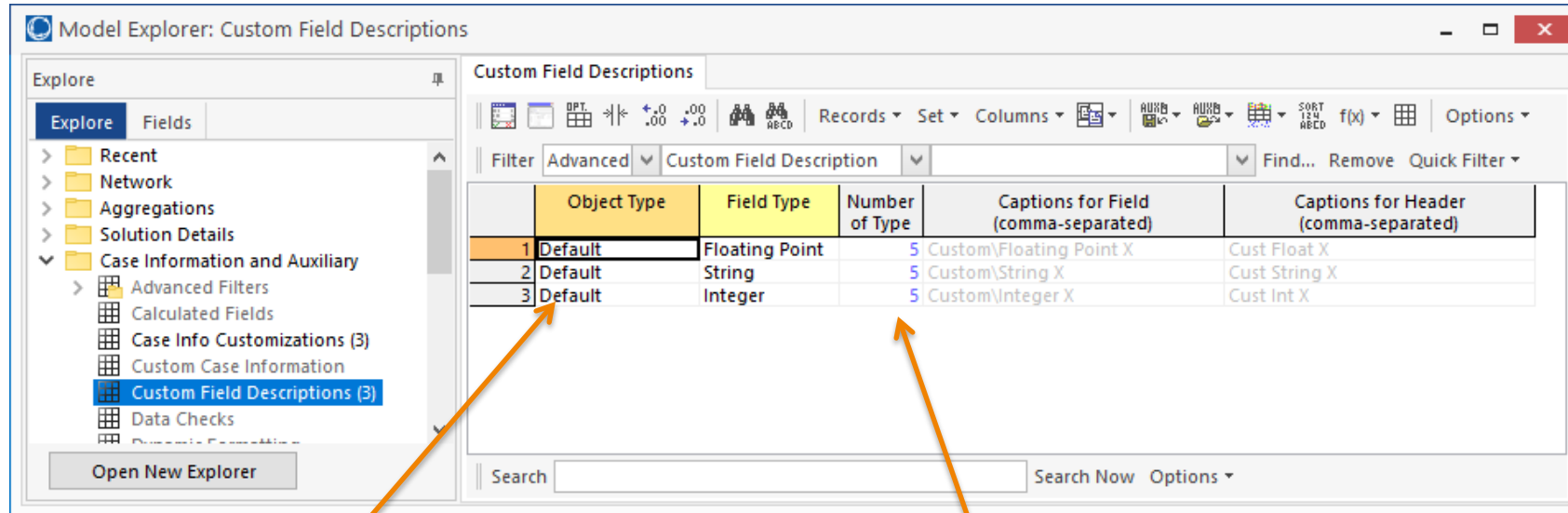
```
SaveData("myfile.CSV", CSVColHeader, Branch, [ALL], [], , [], NO);
```

Custom Fields



- Simulator has many fields, but sometimes that is not enough
- Custom fields are available for data that does not fit anywhere else
- Custom fields can also be useful for temporary storage of values
 - e.g. store the pre-change flow to compare against post-change flow
- By default 5 Custom Floating Point, Custom Integer, and Custom String fields are available for most objects
- Number and names of each type of field can be customized on an object type basis

Custom Fields: Custom Field Descriptions

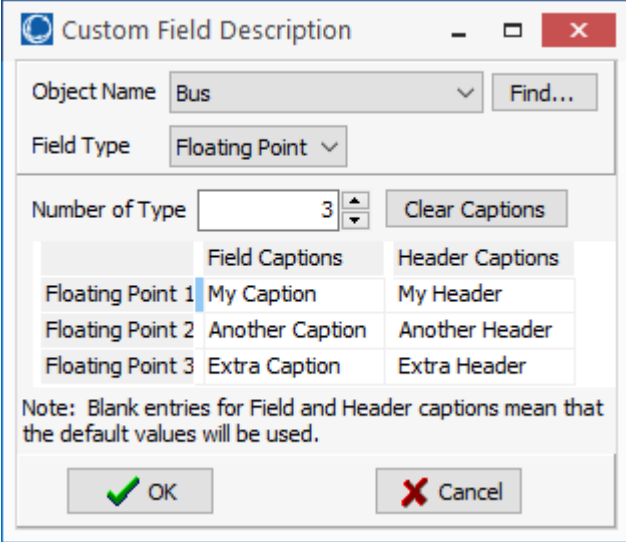


Default object types are always listed.
These apply to all object types.

Number of default fields of a particular type can be changed without additional customization.

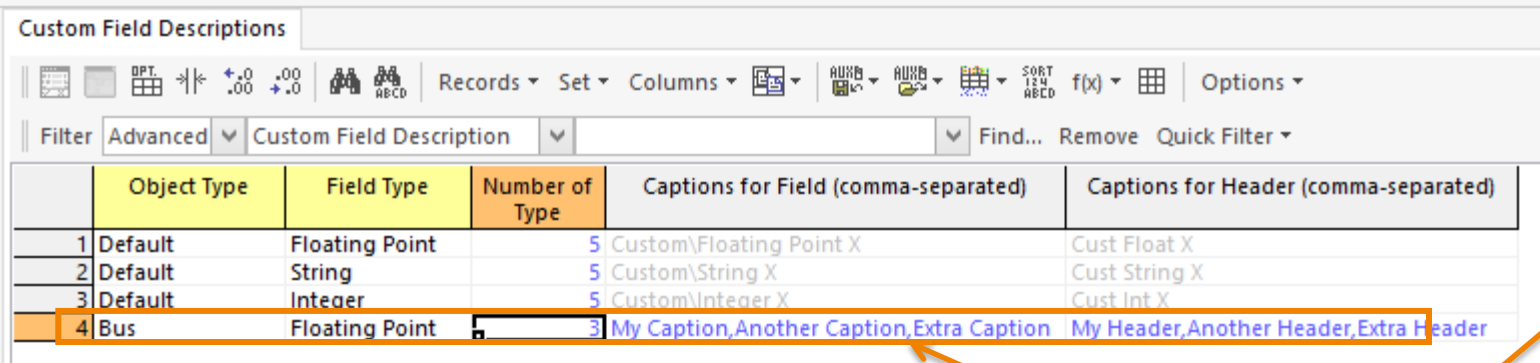
Custom Field Descriptions

- To customize for a particular object type right-click and choose **Insert** (or **Show Dialog** for existing)
 - Choose the **Object Name**
 - Choose the **Field Type**
 - Choose the **Number of Type**
 - Define Field Captions
 - Define Header Captions



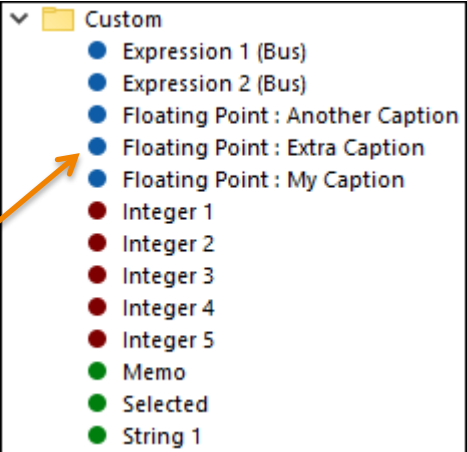
The dialog box is titled "Custom Field Description". It contains the following fields and controls:

- Object Name:** A dropdown menu with "Bus" selected and a "Find..." button.
- Field Type:** A dropdown menu with "Floating Point" selected.
- Number of Type:** A numeric input field with "3" and a "Clear Captions" button.
- Field Captions:** A table with 3 rows and 1 column. The rows are labeled "Floating Point 1", "Floating Point 2", and "Floating Point 3". The captions are "My Caption", "Another Caption", and "Extra Caption".
- Header Captions:** A table with 3 rows and 1 column. The rows are labeled "My Header", "Another Header", and "Extra Header".
- Note:** Blank entries for Field and Header captions mean that the default values will be used.
- Buttons:** "OK" and "Cancel".



The table shows the list of custom field descriptions. The row for "Bus" is highlighted in orange.

	Object Type	Field Type	Number of Type	Captions for Field (comma-separated)	Captions for Header (comma-separated)
1	Default	Floating Point	5	Custom\Floating Point X	Cust Float X
2	Default	String	5	Custom\String X	Cust String X
3	Default	Integer	5	Custom\Integer X	Cust Int X
4	Bus	Floating Point	3	My Caption,Another Caption,Extra Caption	My Header,Another Header,Extra Header

- 
- The list shows the available fields for the "Bus" object type. The fields are:
- Custom
 - Expression 1 (Bus)
 - Expression 2 (Bus)
 - Floating Point : Another Caption
 - Floating Point : Extra Caption
 - Floating Point : My Caption
 - Integer 1
 - Integer 2
 - Integer 3
 - Integer 4
 - Integer 5
 - Memo
 - Selected
 - String 1

List of available fields reflects the user-defined Field Captions

Custom Field Descriptions



Column headings reflect the user-defined Header Captions

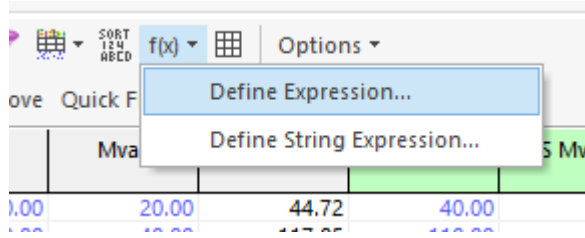
The screenshot shows the 'Model Explorer: Buses' window. On the left, the 'Fields' pane lists available fields under a 'Custom' folder, including 'Expression 1 (Bus)', 'Expression 2 (Bus)', 'Floating Point : Another Caption', 'Floating Point : Extra Caption', 'Floating Point : My Caption', 'Integer 1', 'Integer 2', and 'Integer 3'. The 'Floating Point : My Caption' field is selected. The main pane displays a table with the following columns: 'Number', 'Name', 'Area Name', 'Another Header', 'Extra Header', 'My Header', and 'M'. The table contains 7 rows of data. An orange arrow points from the text 'Column headings reflect the user-defined Header Captions' to the 'Another Header' column header.

	Number	Name	Area Name	Another Header	Extra Header	My Header	M
1	1	One	Top				
2	2	Two	Top				
3	3	Three	Top				
4	4	Four	Top				
5	5	Five	Top				
6	6	Six	Left				
7	7	Seven	Right				

Expressions



- Custom Expressions (Custom String Expressions)
 - Allow calculation to be performed on fields of the object type for which it is defined
 - Model Field, Model Expression, and Model String Expression fields can also be included in the expression function
- Model Expressions (Model String Expressions)
 - Allow calculation on specific fields of specific objects
- Provide a way of performing calculations and updating data within Simulator
- Useful with contingency actions and remedial actions for specifying the value of the applied action



Expressions

Click **New** to add a new expression and then give it a meaningful name

Assign fields to expression variables

Fields can be a **Field** of the particular object type or access fields for particular model objects

Define the function for the expression

Define Expressions

Custom\Expression - Increase MW by 10%

New Save Delete Add Column...

Expression Name
Increase MW by 10%

Variable	Field Selection	Find...	Field Type	Treat blanks as zero	Evaluate In Reference State
x1=	MW\MW	Find...	Field	☒	☐
x2=	Choose a Field	Find...	Field	☐	☐
x3=	Choose a Field	Find...	Field	☐	☐
x4=	Choose a Field	Find...	Field	☒	☐
x5=	Choose a Field	Find...	Field	☒	☐
x6=	Choose a Field	Find...	Field	☒	☐
x7=	Choose a Field	Find...	Field	☒	☐
x8=	Choose a Field	Find...	Field	☒	☐

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

x1*(1.10)

Evaluate

OK Cancel



Expressions



- Example of increasing load by 10%
 - Create expression *Increase MW by 10%*
 - Update the load by the result of the expression using script command

```
SetData(Load, [MW], ["@CustomExpression"], ALL);
```

```
SetData(Load, [MW], ["@CustomExpression:Increase MW by  
10%"], ALL);
```

Supplemental Data



- Original intent was to allow extra information to be stored that is associated with display objects
- Can be used as a user-defined container object similar to other aggregation objects like areas, zones, substations, etc.
 - Details of this will be left for other discussions
- Within the context of script commands and auxiliary files can be used as a place to store user-defined variables
 - Reference these using the **Special Keywords in Script Commands** syntax, **Specifying Field Values in Script Commands** syntax, and as part of Custom Expressions and Model Expressions

Supplemental Data



- Two objects need to be defined
 - **Supplemental Classification**
 - This is used as the category to group the data
 - Example – *My Custom Options*
 - **Supplemental Data**
 - Individual pieces of information that belong to a Supplemental Classification
 - Assign to a Supplemental Classification and provide a Name
 - Example – **Classification** = *My Custom Options*, **Name** = *My Working Directory*

Supplemental Data



Model Explorer: Supplemental Data

Explore

- Recent
- Network
- Aggregations
- Solution Details
- Case Information and Auxiliary
 - Advanced Filters
 - Calculated Fields
 - Case Info Customizations (4)
 - Custom Case Information
 - Custom Field Descriptions (3)
 - Data Checks
 - Dynamic Formatting
 - Expressions
 - Model Conditions
 - Model Expressions
 - Model Filters
 - Model Result Overrides
 - Model String Expressions
 - String Expressions
 - Supplemental Data (3)**
 - User-Defined Case Info Displays

Supplemental Data

Filter: Advanced Supplemental Data

Supplemental Classification Supplemental Data Supplemental Data Contained Objects

	Classification	Name	Cust String 1	Cust String 2	Cust String 3	Cust
1	My Custom Options	My Contingency Directory	c:\mydir\ctgdef\			
2	My Custom Options	My Output Directory	c:\mydir\output\			
3	My Custom Options	My Working Directory	c:\mydir\			

Use custom fields for assigning the values of the user-defined variables

Special Keywords in Script Commands



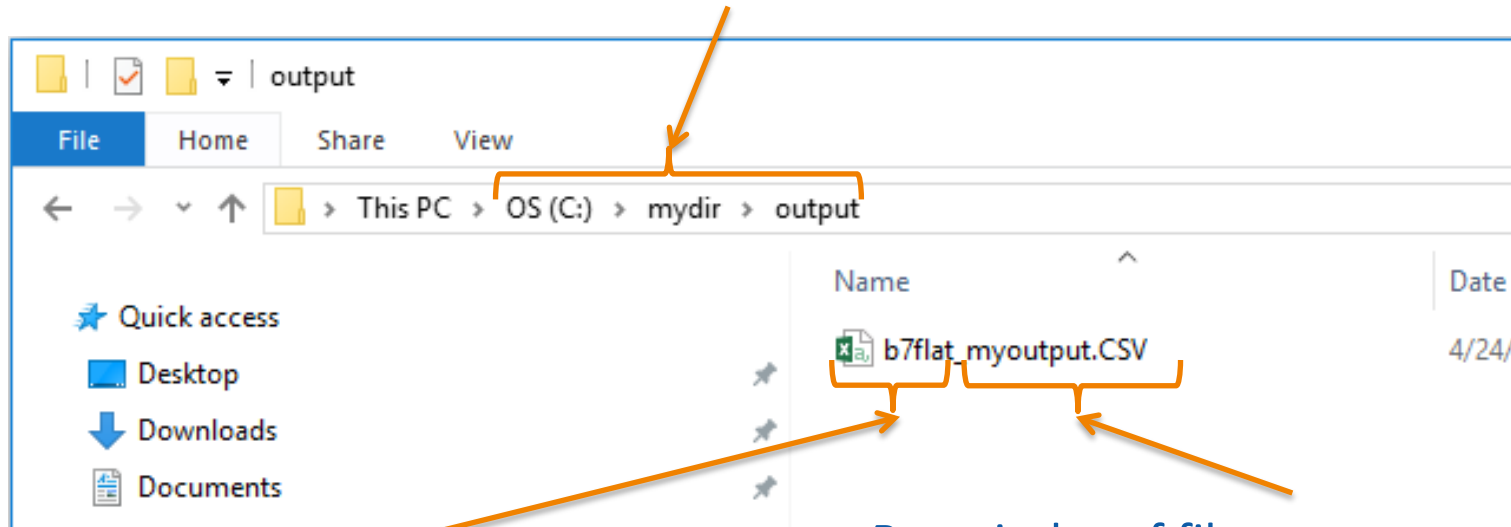
- Allowed as part of file name parameters and other text parameters
 - Generally, these cannot be used for specifying the values to set for specific fields
- The following keywords will be replaced with their actual value when used by a script command
 - @DATETIME, @DATE, @TIME, @BUILDDATE, @VERSION, and @CASENAME
 - @MODELFIELD<objecttype 'key1' 'key2' 'key3' variablename:digits:rod>

Special Keywords in Script Commands



```
SaveData("@MODELFIELD<SupplementalData 'My Custom Options' 'My  
Output Directory' CustomString>@CASENAME_myoutput.CSV", CSV, Bus,  
[Number, NomkV, Vpu, Vangle], [],, [], NO);
```

Directory is specified by @MODELFIELD<SupplementalData 'My Custom Options' 'My Output Directory' CustomString>



@CASENAME includes *b7flat* from *b7flat.pwb*

Remainder of file name, *_myoutput.CSV*, is remainder of the script command parameters that is not a keyword

Special Keywords in Field Values



- These are the exceptions because generally the special keywords are only used in file names and text fields as parameters in script commands
- The fields where this is allowed currently only include fields that specify output file names or directories
- The keywords will be converted at the time that the fields are accessed to determine the name of the directory or file
- The following fields can include the special keywords (these will also show up in relevant dialog fields in the GUI)
 - Transient_Options: ExpDirectory
 - PVCurve_Options: PVCOutFile, PVCStoreStatesWhere
 - QVCurve_Options: QVOutputDir
 - CTG_Options: CTGPostSolAuxFile, PostPostAuxFile, CTGResultStorageFile:1
 - Contingency: PostCTGAuxFile
 - Sim_Environment_Options: SEOSpecifiedAUXFile:0, SEOSpecifiedAUXFile:1, SEOSpecifiedAUXFile:2
 - MessLog_Options: LogAutoFileName

Special Keywords in Field Values Example



PV CURVES

- > Setup
- > Quantities to track
 - Limit violations
 - PV output
 - QV setup
- > PV Results
- > Plots

PV output

☒ Save results to file ☐ Transpose results

☐ Single Header File

State Archiving

☐ Do not save system states

☒ Save c

☐ Save a

Specify a

State Archiving and Plot Storage

Directory Location

Hover hints over the dialog fields indicate the object type and variable name to use in aux files

Name of the PV results output file.
Objecttype=PVCurve_Options VariableName=PVCOutFile

Directory where archived state and plot files should be stored.
Objecttype=PVCurve_Options
VariableName=PVCStoreStatesWhere

```
PVCurve_Options (PVCOutFile,PVCStoreStatesWhere)
{
"@MODELFIELD<SupplementalData 'PV Analysis' '4' CustomString:0>" "@MODELFIELD<SupplementalData 'PV Analysis' '3' CustomString:0>"
}
```

Specifying Field Values



- When setting the value of a field other fields can be referenced by using special formatting
- The following can be used in script commands and data sections
 - `"@variablename:location:digits:decimals"`
 - Sets the field to another field within the same object
 - `"&ModelExpressionName:digits:decimals"`
 - Sets the field to the result of a model expression
 - `"&objecttype 'key fields' variablename:location:digits:decimals"`
 - Sets the field to the value of the specified field for the specified object

Specifying Field Values



- `digits` specifies the total number of digits
 - Default is 32
- `decimal` specifies the digits to the right of the decimal
 - Default is 16
- When using concise variable names both `digit` and `decimal` must be specified if specified at all
 - `location` can be omitted whether or not `digit` and `decimal` are specified
- When using legacy variable names `location`, `digit`, and `decimal` can be omitted but the previous parameters must be specified for a given parameter to be included

Specifying Field Values Examples



```
SetData(Load, [MW], ["@CustomExpression"], ALL);
```

Set the MW value of all loads to the custom expression result for that load

```
Load (BusNum,BusName,ID,MW)
{
    2 "Two"    "1"  "@CustomExpression"
    3 "Three"  "1"  "@CustomExpression"
    4 "Four"   "1"  "@CustomExpression"
    5 "Five"   "1"  "@CustomExpression"
    6 "Six"    "1"  "@CustomExpression"
    7 "Seven"  "1"  "@CustomExpression"
}
```

```
SetData(Load, [MW], ["&Gen 2 '1' MW:8:2"], "<Device>Load 2 '1'");
```

Set the MW value of the load at bus 2 with ID = 1 to the MW output of the generator at bus 2 with ID = 1

```
Load (BusNum,BusName,ID,MW)
{
    2 "Two"    "1"  "&Gen 2 '1' MW:8:2"
}
```

Filtering



- Many script commands take a filter as input to determine the objects on which the command acts
- Valid filter parameters are typically
 - "*FilterName*"
 - Name of Advanced Filter, Device Filter, or Secondary Filter (filtering across object types)
 - AREAZONE
 - Filter based on the area/zone/owner filter
 - SELECTED
 - Filter based on the Selected field = YES
 - ALL
 - Special filter for some script commands to include all objects of a specified object type

Filtering



- Device Filter
 - Instead of creating an Advanced Filter to return a particular object, reference the object directly through the device filter syntax
 - "<DEVICE>objecttype 'key1' 'key2' 'key3'"

```
SetData(Load, [MW], [&Gen 2 '1' MW:8:2"], "<Device>Load 2 '1'");
```

Set the MW value of the load at bus 2 with ID = 1 to the MW output of the generator at bus 2 with ID = 1

Filtering



- Filtering Across Object Types (Secondary Filter)
 - Reference an Advanced Filter for a different object type than the object being filtered
 - Allows reuse of filters
 - Example: define a bus filter and then use this to filter generators, loads, switched shunts, branches, or any other object that connects to a bus
 - If the object being filtered contains more than one of the filter object type OR is assumed
 - If a bus object is being filtered based on an area filter, the bus meets the filter if the area of the bus meets the filter
 - If an area object is being filtered based on a bus filter, the area meets the filter if ANY single bus in the area meets the filter
 - "<objecttype>filtername"

```
SetData(Load, [MW], ["@CustomExpression"], "<Bus>Nom kV > 138");
```

Set the MW value of a load if the terminal bus of the load meets the filter

User Interface During Scripting



- Auxiliary file scripting is a batch process with no looping structure or condition checks
- The intention is to load an auxiliary file and walk away
- There are a few script commands and special syntax that allow user interaction with a GUI dialog during the scripting

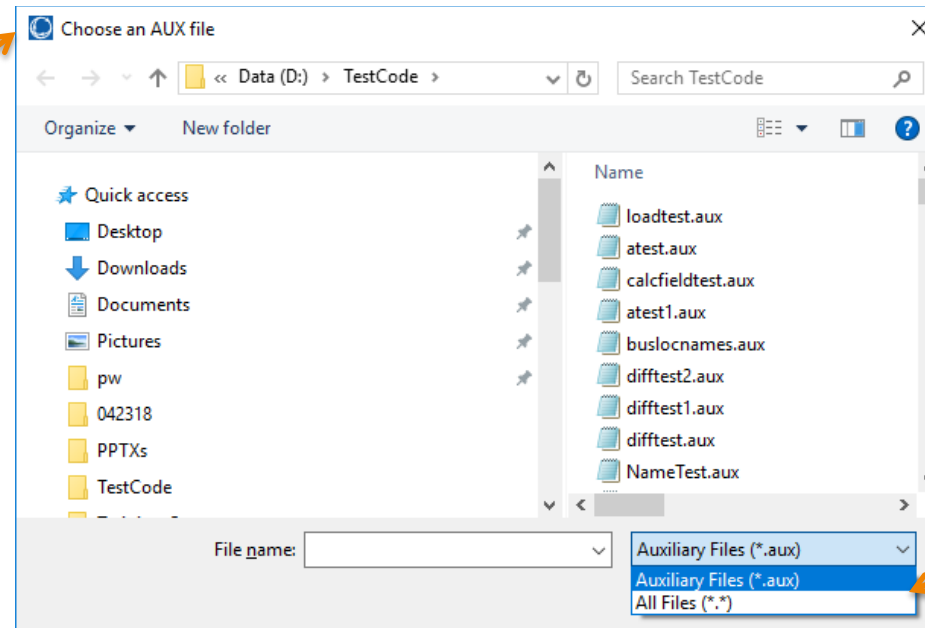
GUI During Scripting



- Special syntax within the filename parameter in script commands will open a dialog to choose a file
 - "<PROMPT 'Caption' 'FileTypes'>"

```
LoadAux("<PROMPT 'Choose an AUX file' 'Auxiliary Files (*.aux)|*.aux|All Files (*.*)|*.*'>");
```

**Specified
Caption**



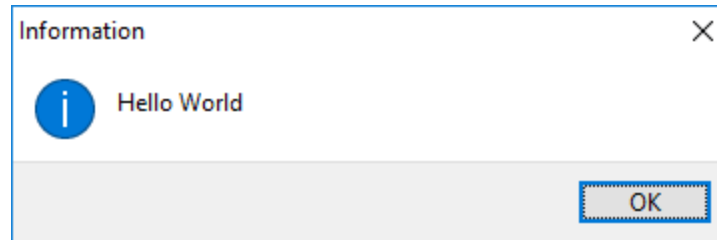
**Specified
FileTypes**

GUI During Scripting



- `MessageBox("text");`
 - Provide a simple text message to the user with no user input to the scripting

```
MessageBox("Hello World");
```



GUI During Scripting



- `OpenDataView("ObjectIDString", "DataGridIDString");`
 - Use the Data View feature to display a dialog during scripting that will allow fields to be changed for a single object in a manner similar to how they are changed in a case information display
 - See the help documentation for full details on how to customize these data views
 - https://www.powerworld.com/WebHelp/Default.htm#MainDocumentation_HTML/Data_View.htm

GUI During Scripting



```
OpenDataView("Bus 1", "DataGrid 'Bus'");
```

- Opens a dialog containing the same fields that are displayed in the Bus case information display in the Model Explorer
- Color coding for field values is the same as in a case information display
- User can change values for the fields that can be edited

The screenshot shows a window titled "Data View for Object" with a search bar containing "Bus '1'". Below the search bar are buttons for "Type", "Refresh", and "New". The main area contains a list of parameters with their corresponding values, some of which are color-coded (blue for text, purple for numbers). The parameters and values are:

Parameter	Value
Number	1
Name	One
AreaName	Top
NeighborsInservice	1
Neighbors	2
NomkV	138.00
Vpu	1.05000
kV	144.900
Vangle	6.09
LoadMW	
LoadMvar	
GenMW	101.85
GenMvar	5.25
ShuntMvar	
ActG	0.00
ActB	0.00

At the bottom, there are buttons for "Layouts", "Modify", a dropdown menu showing "DataGrid: Bus", "Options", and a "Close" button.

GUI During Scripting



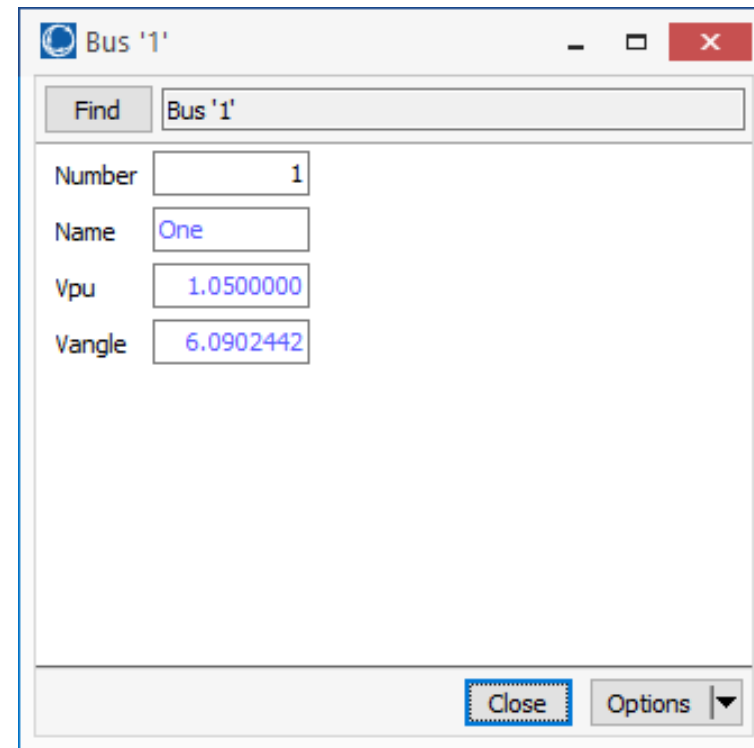
- `ObjectFieldsInputDialog("ObjectIDString", [fieldlist], lots of optional parameters);`
 - Create a custom dialog for displaying fields for a particular object without having to define a DataGrid object
 - See help documentation for details on how to specify the lots of optional parameters
 - <https://www.powerworld.com/WebHelp/Content/Other Documents/Auxiliary-File-Format.pdf>

GUI During Scripting



```
ObjectFieldsInputDialog("Bus 1",[Number, Name, Vpu, Vangle]);
```

- Opens a dialog containing the fields defined in the script command
- Color coding for field values is the same as in a case information display
- User can change values for the fields that can be edited



GUI During Scripting



```
ObjectFieldsInputDialog("Branch 1 2 1",  
[BusNumFrom, BusNumTo, Circuit, LimitMVAA,  
LimitMVAB, LimitMVAC, R, X, Status,  
MWFrom, MWTo, MvarFrom, MvarTo, MVAFrom,  
MVATo], "My Custom Branch Dialog", "My  
Branch", [], [3, 9], ["Input Tab", "Output  
Tab"], [3, 8, 9, 13],  
["Ratings", "Status", "MW", "MVA"], [6, 11],  
["Impedance", "Mvar"]);
```

Calculated Fields

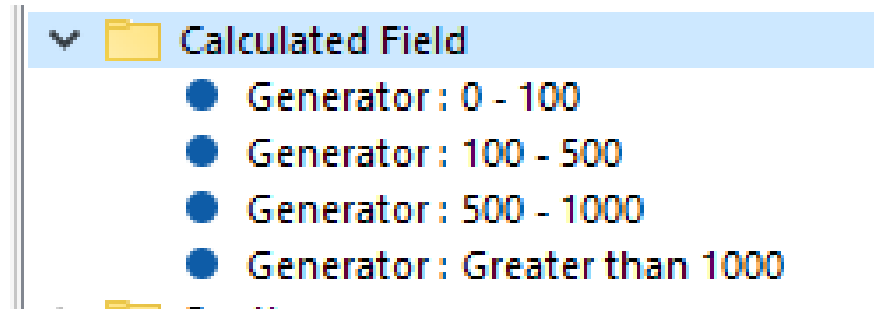


- Provide a way of performing an arithmetic operation on a group of objects of a single object type based on a specific field
- This calculation can then be applied to objects of a different object type
- The final results are based on how the object type in the calculated field definition relates to the object type to which the calculation is applied
 - Define a calculation on the branch object type and then apply this to the bus object type. The final result that is displayed with a bus will perform the calculation on all branches that have at least one terminal connected to that bus.

Calculated Fields Example



- Total Max MW of generators by range
 - Define calculated fields for the ranges of generators: 0 – 100 MW, 100 – 500 MW, 500 – 1000 MW, greater than 1000 MW
- Calculated fields will show up in the list of available fields for all applicable object types



Calculated Fields Example



Object Type on which the calculation is performed

Field and Operation being applied

Filter is applied to the objects of Object Type to determine which ones are included in the calculation

Calculated Field

Calculated Field Name: Gen Max MW 0 - 100

Save Save As Rename Delete

Object Type: Generator

Generator calculations can be used with the following objects.

- 3W Transformer
- Active Contingency
- Area
- Area AGC Model: Generic Are
- balancing Authority
- Branch
- BUS
- Bus TSSStats

Field: Find... MW Output\MW (maximum)

☐ Use Absolute Value

Operation:

- ☐ Minimum
- ☒ Maximum
- ☐ Sum
- ☐ Average
- ☐ Median
- ☐ Variance
- ☐ Standard Deviation
- ☐ Count

Treat Blank Entries:

- ☒ As zeros
- ☐ Ignore

Objects to be included in the operation:

- ☐ All
- ☐ Only objects that meet the filter referenced Choose Filter
- ☒ Only objects that meet the filter below Set Filter Same As...

Select Filter Type: Generator

☐ Pre-filter using Area/Zone/Owner Filters

☒ Enabled (normally checked)

Logical Comparison:

- ☒ AND
- ☐ OR
- ☐ Not AND
- ☐ Not OR
- ☐ XOR
- ☐ One TRUE

Condition 1:

Find... MW Output\MW (maximum) greater than or equal to 0 ☐ ABS

☐ Use Another Filter

Condition 2:

Find... MW Output\MW (maximum) less than 100 ☐ ABS

☐ Use Another Filter

Add >> Delete ...

☐ Enable Field to Field Comparisons

OK Help Cancel

Calculated Fields Example



When used with Area object type, the calculated results give the total Max MW of all generators in each area within the specified MW range

The screenshot shows the 'Model Explorer: Areas' window. On the left, the 'Fields' panel is open, displaying a list of available fields. Under the 'Calculated Field' category, four fields are listed: 'Generator : Gen Max MW 0 - 100', 'Generator : Gen Max MW 100 - 500', 'Generator : Gen Max MW 500 - 1000', and 'Generator : Gen Max MW Greater than 1000'. The 'Areas' panel on the right shows a table with columns for 'Area Num', 'Gen Max MW 0 - 100', 'Gen Max MW 100 - 500', 'Gen Max MW 500 - 1000', and 'Gen Max MW Greater than 1000'. The table contains 21 rows of data, with row 7 highlighted in orange.

	Area Num	Gen Max MW 0 - 100	Gen Max MW 100 - 500	Gen Max MW 500 - 1000	Gen Max MW Greater than 1000	SI
1	1	90.000	373.000	544.000		
2	2	88.000	150.000			
3	3	94.000	437.000	808.000	1379.000	
4	4	84.000	330.000			
5	5	98.730	290.000			
6	6	85.000				
7	7	83.200	308.000			
8	8	99.750	405.000	750.000	1080.000	
9	9	81.000	270.000	950.000		
10	10	96.000	420.000	710.000	1200.000	
11	11	99.500	317.100	825.700	1200.000	
12	12	90.000	478.500	530.000		
13	13	92.500	178.300			
14	14	99.500	435.000	527.400		
15	15	97.000	259.200			
16	16	87.000	330.000	823.000		
17	17	45.000				
18	18	70.000	300.000			
19	19	99.500	490.000	575.000		
20	20	90.600	383.000	780.000		
21	21	93.700	479.000	605.000		

Calculated Fields Example



When used with Zone object type, the calculated results give the total Max MW of all generators in each zone within the specified MW range

The screenshot shows the 'Model Explorer: Zones' window. On the left, the 'Fields' panel is open, displaying a list of available fields. Under the 'Calculated Field' section, four options are listed: 'Generator : Gen Max MW 0 - 100', 'Generator : Gen Max MW 100 - 500', 'Generator : Gen Max MW 500 - 1000', and 'Generator : Gen Max MW Greater than 1000'. The 'Zones' panel on the right shows a table with columns for 'Zone Num' and four calculated MW ranges. The table contains 24 rows of data.

Zone Num	Gen Max MW 0 - 100	Gen Max MW 100 - 500	Gen Max MW 500 - 1000	Gen Max MW Greater than 1000
1				
2	11			
3	100	90.000	373.000	544.000
4	101			
5	102			
6	103			
7	104			
8	105			
9	106			
10	107			
11	108			
12	109			
13	110	88.000	150.000	
14	111			
15	112			
16	113			
17	114			
18	116			
19	117			
20	120			
21	121	30.200		
22	122	40.800	268.000	
23	123	40.800		
24	130			

Calculated Fields



- Other examples
 - Return the buses in the island with the most buses
 - Set the system slack bus based on generator criteria
 - Maximum percent flow on all branches in an area or zone
 - Trip injection group with largest output
 - GIC flows in a substation
 - Losses by owner and nominal kV level
 - PV results with the lowest transfer level

Auxiliary File Export Format Descriptions



- Allows you to define a list of DATA sections and fields you would like to write out to an auxiliary file
- These formats themselves can then be saved to an auxiliary file and used whenever needed
- The following options define an aux export format
 - Object Type: Specify type of object
 - Filter Name: *All*, *Selected*, *AreaZone*, the name of an Advanced Filter in quotes, or a Filter Condition. 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
- Script command can be used to save an auxiliary file in a defined format
 - `SaveDataUsingExportFormat("filename", filetype, "FormatName");`

Auxiliary File Export Format Descriptions

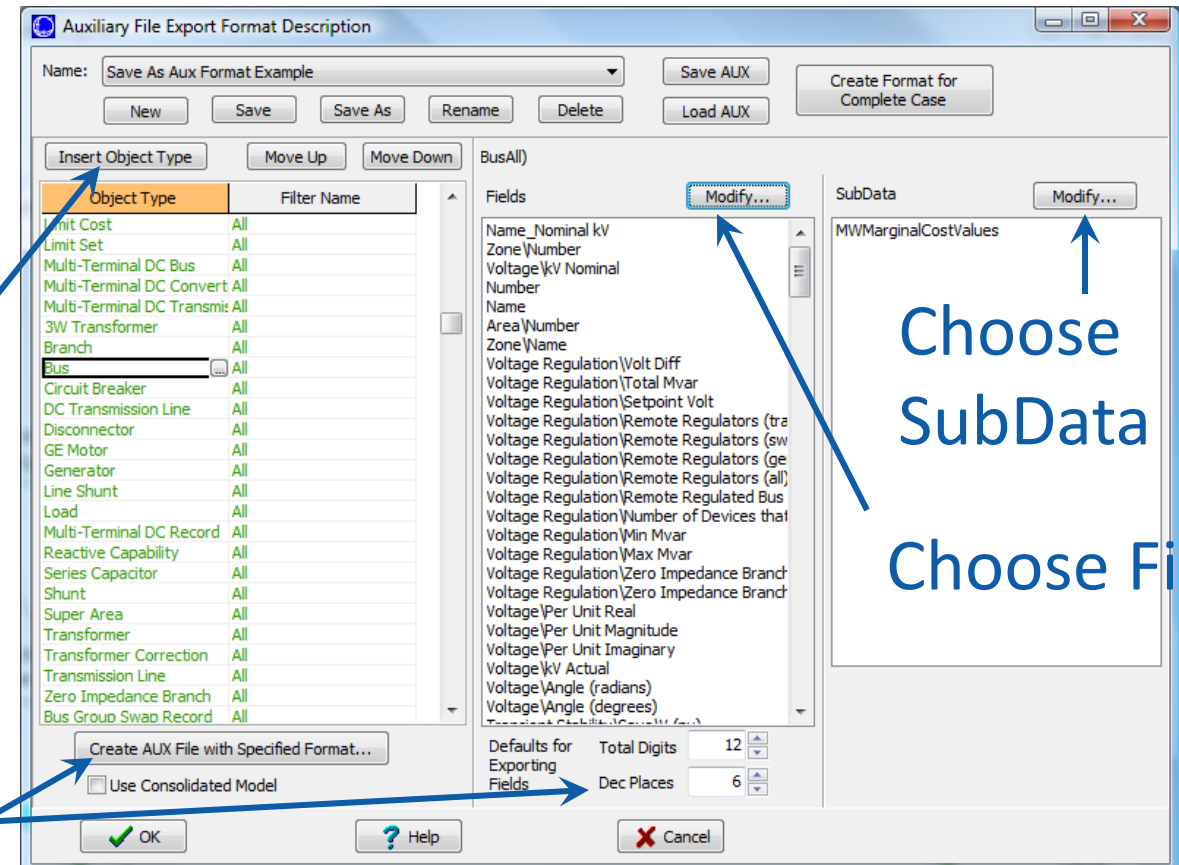


- On the Case Information Ribbon Tab, click **AUX Export Format Desc...**



Choose Object Types

Set formatting and file type

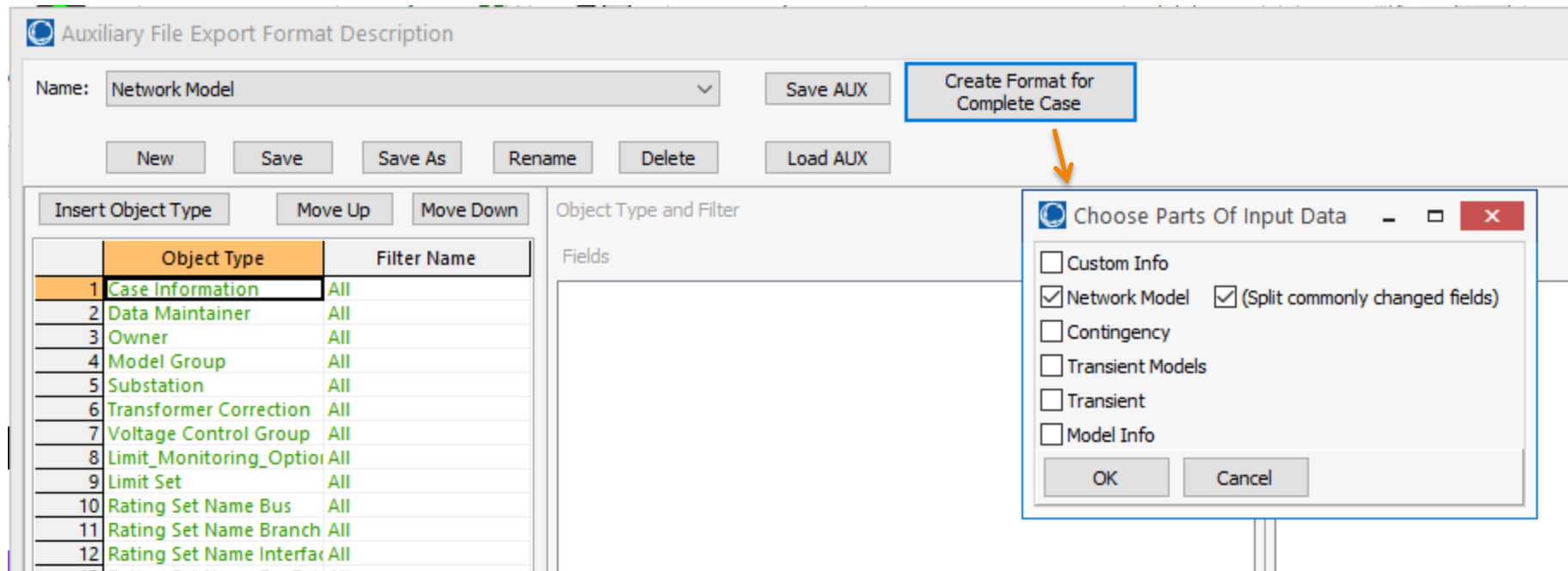


Choose SubData
Choose Fields

Auxiliary File Export Format Descriptions



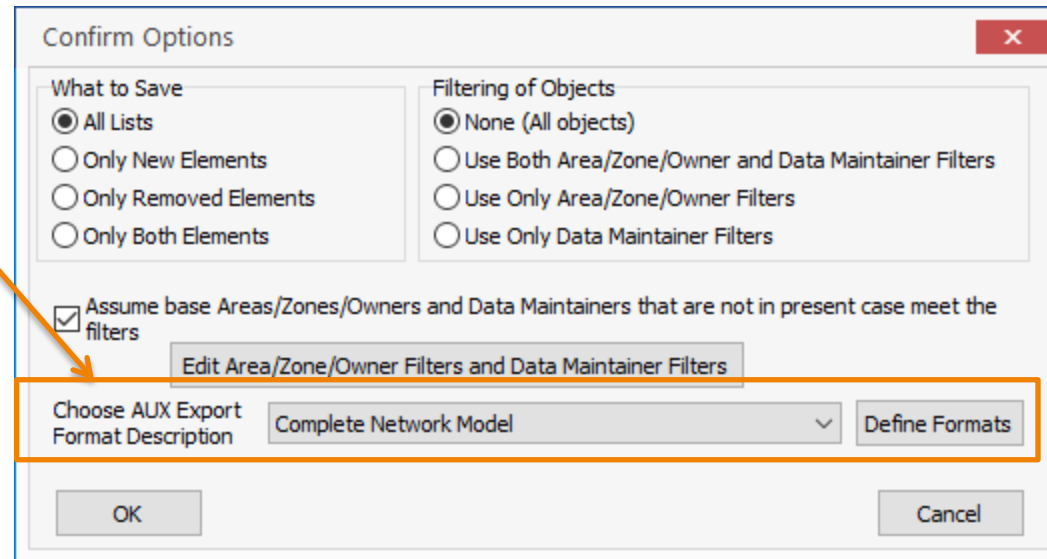
- Default formats can be used as a starting point for customizations



Auxiliary File Export Format Descriptions



- AUX export formats can be used with Present Case Topological Differences from the Base Case dialog to customize fields and objects saved



The image shows a 'Confirm Options' dialog box with the following sections:

- What to Save:**
 - ☒ All Lists
 - ☐ Only New Elements
 - ☐ Only Removed Elements
 - ☐ Only Both Elements
- Filtering of Objects:**
 - ☒ None (All objects)
 - ☐ Use Both Area/Zone/Owner and Data Maintainer Filters
 - ☐ Use Only Area/Zone/Owner Filters
 - ☐ Use Only Data Maintainer Filters
- ☒ Assume base Areas/Zones/Owners and Data Maintainers that are not in present case meet the filters
-
- Choose AUX Export Format Description:**
 - Complete Network Model (dropdown menu)
 -
-

An orange arrow points from the text 'objects saved' in the bullet point above to the 'Edit Area/Zone/Owner Filters and Data Maintainer Filters' button. An orange box highlights the 'Choose AUX Export Format Description' section.