Record Specification and File Format for Specifying a Power Flow Case

Prepared for	:	PowerWorld Users
Date	:	2019
Prepared by	:	James Weber, Ph.D. Director of Software Development PowerWorld Corporation (217) 384-6330 ext. 13 weber@powerworld.com

2001 South First Street
Champaign, IL 61820
(217) 384-6330
www.powerworld.com

Table of Contents

1 DOCUMENT SUMMARY 3

2 BASIC FILE FORMAT RULES 3

2.1 SYNTAX RULES 3

2.1.1 Naming Conventions 3

2.1.2 Handling quotes inside of quoted strings 3

2.2 OBJECT TYPE STRINGS 4

2.2.1 Branch Objects (2-terminal AC devices) 4

2.3 SPECIFYING AN OBJECT USING A STRING 4

2.3.1 Primary Keys 5

2.3.2 Secondary Keys 5

2.3.3 Label Identifiers 6

2.3.4 AllLabels Field 6

2.3.5 Naming Collisions 7

2.4 SPECIAL TREATMENT OF BUSNUM INTEGER KEY FIELDS 7

3 AGGREGATION OBJECT NOTES 8

3.1 AREA, BALANCINGAUTHORITY, AND ZONE OBJECTS 8

3.1.1 Defining Area, BalancingAuthority, Zone objects 8

3.1.2 Using Area, BalancingAuthority, and Zone objects in Filtering 8

3.1.3 Using Area, BalancingAuthority, and Zone objects with regard to losses on Tie-lines 8

3.2 SUBSTATIONS 8

3.3 OWNER 8

3.4 DATAMAINTAINER 9

3.4.1 Specification of DataMaintainer within Software 9

3.4.2 Objects that don’t support DataMaintainer (NO/NO) 10

3.4.3 Objects to which DataMaintainer can be assigned, but Inheritance is not allowed 10

3.4.4 Objects that Can Inherit a DataMaintainer 10

3.4.5 Illustration of Objects that Can Inherit a DataMaintainer from Network Topology 13

3.4.6 DataMaintainer being assigned its own DataMaintainer 14

4 OBJECT DEFINITIONS 15

4.1 PWCASEINFORMATION 15

4.2 DATAMAINTAINER 15

4.3 OWNER 16

4.4 LOADMODELGROUP 17

4.5 SUBSTATION 17

4.6 XFCORRECTION 18

4.7 VOLTAGECONTROLGROUP 18

4.8 LIMIT_MONITORING_OPTIONS_VALUE 19

4.9 LimitSet ... 20
4.9.1 Branch and Interface relationship to LimitSet ... 21
4.9.2 Bus relationship to LimitSet ... 22
4.10 RatingSetNameBranch ... 22
4.11 RatingSetNameInterface ... 23
4.12 RatingSetNameBus ... 23
4.13 Bus ... 24
4.14 Gen ... 27
4.15 ReactiveCapability ... 31
4.16 Load ... 32
4.17 Branch (Not a Transformer) ... 34
4.18 Branch (Transformer) ... 36
4.19 3WXformer ... 40
4.20 MultiSectionLine ... 44
4.21 Shunt (ShuntMode <> Bus Shunt) ... 45
4.22 Shunt (ShuntMode = Bus Shunt) ... 49
4.23 LineShunt ... 51
4.24 DFACTSCorrection ... 52
4.25 DFACTS ... 53
4.26 DCTransmissionLine ... 54
4.27 VSCDCLine ... 56
4.28 MTDCRecord, MTDCBus, MTDCConverter, MTDCTransmissionLine ... 58
4.29 InjectionGroup, PartPoint ... 62
4.30 SuperArea ... 64
4.31 Area ... 64
4.32 StudyMWTransaction ... 66
4.33 BalancingAuthority ... 67
4.34 Zone ... 67
4.35 Interface, InterfaceElement ... 68
4.36 Sim_Solution_Options_Value ... 71
4.37 Nomogram ... 75
5 USING POWERWORLD SIMULATOR TO SAVE THIS INFORMATION ... 76
5.1 File, Save As Method ... 76
5.2 Create an Auxiliary File Export Format Description ... 77

1 Document Summary

This document will describe a database structure which can be used to store all information which is needed to replicate a power flow case in either a RAW, EPC or PowerWorld AUX file. This syntax is the same syntax used for the WECC Common Format for Contingency and RAS Definitions and would provide a template for sharing data among engineers in an open format managed by PowerWorld Corporation.

2 Basic File Format Rules

2.1 Syntax Rules

The following is a list of general syntax rules that apply throughout this file format:

- All strings are case *insensitive*. Thus "`Contingency`" is treated the same as "`CONTINGENCY`" or "`contingency`" or "`CoNtInGeNcY`". This will be true for names, fields, and any key words in the file syntax.
- Any line that starts with the two backslashes (//) will be treated as a comment and be ignored when parsing the file. Text appearing after two backslashes will also be treated as comments.
- Blank lines of text are ignored and skipped
- Many text lines are space delimited strings that use double quotes ("") as string unifiers. Note that these are straight quotes. Smart quotes such as " " are not supported by the format so be careful when copying and pasting from some text editors.
- Any TAB characters in the text file will be treated as a single space when read by a file parser
- Consecutive spaces in the position of a delimiter are treated as a one string delimiter

2.1.1 Naming Conventions

There are many objects in this file format that have a name including Model Conditions, Model Filters, Model Expressions, Contingencies, Remedial Action Schemes, and Injection Groups. There will be no restrictions placed on the length or content of the names in this format, except that they are limited to ASCII characters. Obviously the user should use discretion and not create names with 1,500 characters, but the file format will not specifically preclude this bad behavior.

2.1.2 Handling quotes inside of quoted strings

There are many places in the file format that require a string enclosed in either double quotes or single quotes. Sometimes there are even double quote strings that contain a space delimited string that uses single quotes inside of the double quotes.

For example, to specify a particular area, a space delimited string containing the string AREA followed by the name of the area enclosed in single quotes is used. The entire string is then enclosed in double quotes. However, if the name of the area contains either double or single quotes, this could cause trouble in the text parser. To accommodate this potential, the format specifies that such quotes or double quotes be repeated if contained inside of a string. Consider the examples in the following table:

Area Name	Object ID String
WECC's Office	"AREA 'WECC''s Office'"
"HIGH" Point’s	"AREA '""HIGH"" Point''s'"

In most situations such as object names, using quotes or double quotes is highly discouraged. However, it may be natural to include quotes in some of the fields in this document such as the Memo field for a contingency. The memo field is a free-form string in which the user includes notes about the contingency. Thus in general, this format will not enforce any requirement regarding quotes because in the end it is not necessary. The format will require that software parsers handle these situations.

2.2 Object Type Strings

There are many places in this file format where a particular object type must be referenced. All object type strings must not contain any spaces. The following is a list of some of the allowable object types:

```
Branch, Bus, Gen, Shunt, Load, Area, Zone, Substation, InjectionGroup, Interface,
3WXFormer, DCTransmissionLine, LineShunt, VSCDCLine, ModelExpression, Contingency,
ContingencyElement, TSContingency, TSContingencyElement, RemedialAction,
RemedialActionElement, CTG_Options_Value, Sim_Solution_Options_Value, LimitSet,
CustomMonitor, ModelFilter, ModelFilterCondition, ModelCondition,
ModelConditionCondition, Filter, Condition
```

2.2.1 Branch Objects (2-terminal AC devices)

An object type BRANCH signifies either an AC transmission line, 2-winding transformer, a series capacitor or reactor, or any AC device which connects two buses. Within a BRANCH there is then a field BranchDeviceType which can have the following entries: Line, Transformer, Series Cap, Breaker, Disconnect, ZBR, Fuse, Load Break Disconnect, or Ground Disconnect. This enumeration of device types comes from the Common Information Model (CIM) specification, except that a ZBR is called a Jumper in CIM. In general a user may toggle between these various device types, with the exception of a Transformer. Once an object is specified as a transformer it may not be turned back into a another branch device type.

2.3 Specifying an object using a string

There are many places in this file format where a particular object must be referenced. In these situations a string will be specified that is enclosed in double quotes. The object string will be space delimited with the first string representing the object type. Object type strings from Section 2.2 will never have spaces in them. Following the object type string there will be identification information for the object. This information allows for three potential formats that the software will need to parse: Primary Keys, Secondary Keys, or Labels. While each object can have labels, each different object type can have a different number of key fields. The key fields for the various object types are as follows:

Object Type	Primary Key Fields	Secondary Key Fields
Gen	BusNum ID	BusNameNomkV ID
Bus	Number	NameNomkV
Branch	BusNumFrom BusNumTo Circuit	BusNameNomkVFrom BusNameNomkVTo Circuit
Load	BusNum ID	BusNameNomkV ID
Shunt	BusNum ID	BusNameNomkV ID
Area	Number	Name
Zone	Number	Name
Substation	Number	Name
InjectionGroup	Name	*none available*
Interface	Name	Number
3WXformer	BusNumPri BusNumSec BusNumTer Circuit	BusNameNomkVPri BusNameNomkVSec BusNameNomkVTer Circuit
DCTransmissionLine	BusNumRect BusNumInv Circuit	BusNameNomkVRect BusNameNomkVInv Circuit
LineShunt	BusNumFrom BusNumTo BusNumLoc Circuit ID	BusNameNomkVFrom BusNameNomkVTo BusNameNomkVLoc Circuit ID
VSCDCLine	Name	*none available*
ModelExpression	Name	*none available*

2.3.1 Primary Keys

Primary keys for many objects are the bus numbers associated with the object and some string identifiers. The format of the object string using primary keys is then the object type and then a list of keys separated by spaces. If a key string has any spaces, a *single* quote must be used to enclose the key. Note: A single quote is used because throughout the format these entire object strings are enclosed in double quotes.

General Format	"Objecttype 'key1' 'key2' 'key3'"
Generator	"GEN 23 '12'"
Bus	"BUS 33"
Branch	"BRANCH 23 29 'AB'"
3WXformer	"3WXFORMER 23 29 66 'AB'"
Area	"AREA 51"
Zone	"ZONE 93"
Substation	"SUBSTATION 37"

2.3.2 Secondary Keys

Secondary keys for some objects are also available. These are often a combination of the bus name and nominal kV value of a bus, or for other objects they replace the numbers with names. Not all objects will have secondary key fields. For example, an injection group has only a name.

General Format	"Objecttype 'key1' 'key2' 'key3'"
Generator	"GEN 'Bus 23_138.00' '12'"
Bus	"BUS 'Bus 33_500.00'"
Branch	"BRANCH 'Bus 23_138.00' 'Bus 29_138.00' 'AB'"
Area	"AREA 'Fifty One'"
Zone	"ZONE 'Ninety Three'"
Substation	"SUBSTATION 'Thirty Seven'"

One special note regarding commonly used secondary keys that use the Name_NomkV convention. When reading this string from input, the string is processed by starting at the last character and searching backward until an underscore (_) character is found. The string is then split between the name and the Nominal voltage. When comparing whether two voltage are equal for the purposes of the nominal voltage we check to see if the string's value (call this *StringNomkV*) is within 0.1% of an existing bus' nominal kV (call this *BusNomkV*). Said mathematically, perform the following check

```
abs(BusNomkV - StringNomkV)/ BusNomkV < 0.001)
```

This ensures that nominal voltages are equal to at least 3 significant digits.

2.3.3 Label Identifiers

Label Identifiers are can also be specified for a particular object. Each particular object could potentially have multiple labels assigned to them, but within one object type, only one object can have a particular label. The format of the object string using a label is then simply the object type followed by the label enclosed in single quotes.

General Format	"Objecttype 'label'"
Generator	"GEN 'GrandCoule12'"
Bus	"BUS 'Coulee_N56'"
Branch	"BRANCH 'CaptJackGrizzly_56"

2.3.4 AllLabels Field

Many objects related to the topology of the model have a field associated with them called AllLabels. Objects can have more than one label associated with them but within one object type, only one object can have a particular label. The AllLabels field is a single comma-delimited string containing all the labels assigned to one particular object. Also remember that in the format each AllLabels field is enclosed in double quotes. To accommodate these conventions, if any label contains a comma, then this string for the label will use a single quote as the unifying character around a single label. In addition, for all labels any single quotes or double quotes must be repeated.

Consider the examples in the following table showing how AllLabels field is constructed for 3 labels.

Label #1	Label #2	Label #3	AllLabels Field writing to format
DEF	Bob"s,Home Requires unifying single quotes, repeated double quote	ABC 'Care' Requires repeated single quotes	"DEF,'Bob""s,Home',ABC ''Care''" Repeated quotes are highlighted in yellow Unifying single quotes highlighted in green
West	East	North	"West,East,North"

Using quotes, double quotes, or commas in a label is discouraged, but the format does not prevent this. Thus in general, this format will not enforce any requirement regarding quotes because in the end it is not necessary. The format will require that software parsers handle these situations.

2.3.5 Naming Collisions

It is possible in this format for a power system model to have secondary fields which do not create a unique identifier for the case. For example, the secondary key fields for buses are the concatenation of the Name and Nominal kV. These are *almost* always unique, but not *always*. For example, a recent WECC cases and there are 4 buses named CanyonGT at 13.8 kV (numbers 25211-25214). When reading from a file referencing the bus "CanyonGT_13.8", this format would just pick the one that a software vendors search routine finds first. There is no guarantee that this will be the one intended, so this is something the user must be careful with if using secondary key fields.

It is also possible for the label identifiers to collide with the secondary keys as well. Our experience in practice is that the labels are derived from unique identifiers in the EMS models which are longer than the secondary key strings and are a concatenation of the substation name(s), some unique delimiter like a $. Thus in practice this shouldn't happen, but it is possible. It is even possible with labels that there could be a conflict between the primary key and the label. We do not expect to see too many buses with a label of "1234", but a user could do something like that. Regardless of these hypothetical limitations, when parsing these strings, this format instructs that software parsers will always look first for the primary keys, then the secondary keys, and finally for any of the labels. Thus if there is a conflict between secondary keys and the labels, then the secondary key will have precedence.

2.4 Special Treatment of BusNum Integer Key Fields

Many objects use the bus number as one primary key field. This discussion does not apply to the Bus objecttype itself, but instead includes Gen, Load, Shunt, LineShunt, Branch, DCTransmissionLine, MTDCConverter, DFACTS, 3WXformer, and GenVarLim objects. Following the strict definition of the BusNum as an integer key field would require you to use this integer whenever you create one of these object types. However, the format makes special provisions to allow the use of the terminal bus' Name_NomkV (secondary key for bus) or one of the bus labels instead. In order to do this, when reading a record which uses the BusNum as a key, the software should interpret this as follows.

1. If the string read to is an integer, then attempt to lookup the bus using this integer otherwise the bus is not found.
2. If no bus was found in step 1, then attempt to split the string in the format of Name followed by an underscore and then followed by the nominal voltage. Then search for a bus which has this name and nominal voltage as described in Section 2.3.2.
3. If still no bus is found then attempt to search for a bus with a label that matches the string.

In this way, new objects can be optionally created without referring to integer identifiers but instead referring to either secondary or label identifiers of the bus.

3 Aggregation Object Notes

There are several aggregation definitions which can group together objects in a particular manner. The treatment in the data structure for these is defined here.

3.1 Area, BalancingAuthority, and Zone objects

Area, BalancingAuthority and Zone objects have the same treatment in the data structure. We will first introduce how to define (input) the area and zone definitions and then discuss how the areas and zones are used. How the Area and Zone definitions are interpreted will depend on where they are used.

3.1.1 Defining Area, BalancingAuthority, Zone objects

Each Bus in the model *must be* assigned to exactly one Area, exactly one BalancingAuthority and exactly one Zone. Note that there does not need to be any relationship at all between Area, BalancingAuthority and Zone objects. Frequently a zone is modeled as a sub-region of an area, but this does not need to be true. In most situations this is the end of the input specification, however, the objects Gen, Load, and Shunt may also be assigned to an Area, BalancingAuthority or Zone that is different than the terminal bus.

3.1.2 Using Area, BalancingAuthority, and Zone objects in Filtering

How the Area and Zone definitions are interpreted will depend on where they are used. For example, when filtering devices Gen, Load, or Shunt objects then they will obey any area/BA/zone assigned to the particular object, or if none is assigned they will use the area/BA/zone designation with the terminal bus.

For devices that have multiple terminals though, such as a Branch or DCTranmissionLine, then the device will be considered inside the area, BA, or zone if *any* of its terminals is inside. Interfaces will be inside an area, BA, or zone if *any* of the InterfaceElement devices are inside it.

3.1.3 Using Area, BalancingAuthority, and Zone objects with regard to losses on Tie-lines

When looking at which Area, BalancingAuthority, or Zone object is assigned the losses on a tie-line between two of them, then the device connecting them will have a "MeteredEnd" specified. The losses will then be assigned to the *non*-metered end.

3.2 Substations

Each Bus in the model can be assigned to one substation. This is not a requirement, but an option. For the purposes of filtering, an object is considered to meet the substation filter if *any* of the terminal buses are in the substation.

3.3 Owner

Owners are a slight modification to the aggregation concept because for some devices (Gen, Branch, Shunt), the object can be partially owned by several different Owner objects. When used with filtering the device will be considered "inside" the Owner if any part of it is owned. When showing totals such as total generation MW or Mvar, then a pro-rated portion of the generation output will be shown.

3.4 DataMaintainer

DataMaintainer objects represent entities responsible for maintaining the input data for objects in the model. The DataMaintainer object serves 2 purposes

1. If you suspect a piece of data in a case is in error you now have contact information in the form of an email or phone number of someone to contact to ask about this data
2. Software has features that allow a user to write out *only* the data that belongs to a particular DataMaintainer. This construct makes it easier to divide a case into chunks that represent the responsibility of maintaining particular data.

The following statements describe how DataMaintainers relate to other objects.

1. Some objects can be assigned a specified DataMaintainer.
2. Some objects may inherit their DataMaintainer from a related object.
3. Not all object types can even be considered to belong to a DataMaintainer.
4. Ultimately, an object can belong to only *one* DataMaintainer, and the assumption is that all fields associated with that object are the responsibility of the DataMaintainer.
5. Caveat to statement 4: a DataMaintainer object itself can belong to another DataMaintainer. In this way groups of DataMaintainers can be created.

Each ObjectType has 2 YES and NO questions based on the first two statements above. Can it be assigned a DataMaintainer? YES or No. Can it Inherit its DataMaintainer from a related object? YES or NO. This gives us 4 permutations to the Assign/Inherit questions: YES/YES, YES/NO, NO/YES and NO/NO.

3.4.1 Specification of DataMaintainer within Software

Within PowerWorld objects there are potentially 2 different columns associated with a DataMaintainer.

- **DataMaintainerAssign** column is the column that should be used to directly assign a DataMaintainer to a specific object. If an object does not have a DataMaintainer specified directly then this field is just blank. In addition, to clear the DataMaintainerAssign specification you can set this column to a blank string as well. Only objects that support having a DataMaintainer assigned will have this field.
- **DataMaintainer** column is the column that will show the active DataMaintainer for the object. This will show the inherited DataMaintainer if one is available. An object that either can have the DataMaintainer assigned or can inherited the DataMaintainer will include this field.
- [DataMaintainerInherit were added in Version 19, build on December 1, 2016]
  DataMaintainerInherit column is a YES/NO field. A value of YES indicates that inheritance is allowed for this objects
  - For objects listed below which do not allow Inheritance, this field is not enterable and will always show NO.
  - For objects below which AlwaysInherit, this field will also not be enterable and will always show YES.
  - For the small list of objects such as Bus, Gen, Load, Shunt, LineShunt, Branch, 3WXFormer, DFACTS, DCTransmissionLine, MTDCRecord and VSCDCLine which can have their own DataMaintainer assigned, but also can inherit the DataMaintainer, then this field will be enterable and the user may toggle the value between YES or NO. Setting the

value to NO will prevent these objects from inheriting a DataMaintainer. See the **purple** dots in Section 3.4.5 below for an indication of objects which can toggle this field.

- [DataMaintainerInheritBlock were added in Version 19, build on December 1, 2016] **DataMaintainerInheritBlock** column is a YES/NO field only available for a Bus and a Substation record. See the **pink** rounded rectangles in Section 3.4.5 below.
  - For a Bus, set to YES to block the inheritance of the Data Maintainer for other objects connected to this bus. For example, if this is YES, generators at this bus will not inherit the Data Maintainer from this bus.
  - For a Substation, set to YES to block the inheritance of the Data Maintainer for the buses in this substation.

3.4.2 Objects that don't support DataMaintainer (NO/NO)

We won't discuss all the ObjectTypes that represent the NO/NO permutation, but generally these are objects representing one of the following 4 types of ObjectTypes.

1. Solution options (CTG_Options, SimSolution_Options, etc.)
2. Environment options (RatingSetNameBus, RatingSetNameBranch, etc.)
3. Objects that are dynamically created and maintained by the software (Island, ZoneTieLine, AreaTieLine, SuperBus)
4. Results of a software calculation (ViolationCTG)

3.4.3 Objects to which DataMaintainer can be assigned, but Inheritance is not allowed

The following is a list of ObjectTypes that can be assigned a DataMaintainer but never Inherit their DataMaintainer from another object (YES/NO).

Area, BalancingAuthority, BGCalculatedField, Contingency, CTGElementBlock, CustomExpression, CustomExpressionStr, CustomMonitor, DataMaintainer, DFACTSCorrection, Direction, DistributionEquivalent, Fault, Filter, GlobalContingencyActionsElement, InjectionGroup, Interface, LimitSet, LoadModelGroup, ModelCondition, ModelExpression, ModelFilter, ModelStringExpression, MutualImpedance, Nomogram, Owner, PlayIn, PVPlot, RemedialAction, StudyMWTransactions, Substation, SuperArea, SupplementalData, TimeStepAction, TSContingency, TSLimitMonitor, TSPlot, VoltageControlGroup, XFCorrection, Zone

3.4.4 Objects that Can Inherit a DataMaintainer

Adding the ability for objects to inherit their DataMaintainer may seem to add complexity to the concept of a DataMaintainer. However, it is actually crucial to the use the DataMaintainer. This is because without it, it would require every single data record to be assigned a DataMaintainer independently, which would greatly increase the workload for those adding this new assignment. What we expect to occur instead is that each substation in the model will be assigned a DataMaintainer and that the vast majority of network objects will then simply inherit their DataMaintainer definition from the substation. Even if substations are not added to the models, the DataMaintainer could be assigned to a Bus with inheritance occurring from there.

That said we do expect that substations will be included in the model and that all busses will be assigned to a substation. We expect that to occur for the following reasons.

1. Geographic Information Systems tend to all be based on substation
2. The calculation of Geomagnetic Induced Current (GIC) require the definition of substations to accurately perform the calculation
3. Real time full topology models use the substation as their fundamental data record

With this explanation for why the inheritance of DataMaintainers is vital, we now cover the final 2 other permutations (YES/YES and NO/YES). This information can always be obtained from within PowerWorld Simulator by going to the Windows ribbon tab and choose **Export Case Object Fields > Send to Excel**. In this table in the row representing the ObjectType there are columns which show whether Data Maintainers are supported and whether Data Maintainer Inheritance is supported and how. The table below shows a summary of the objects that existed as of January 2016 in PowerWorld Simulator. The columns in this table are the ObjectType, whether it can be assigned a DataMaintainer, and finally a precedence of how it inherits its DataMaintainer from a related object.

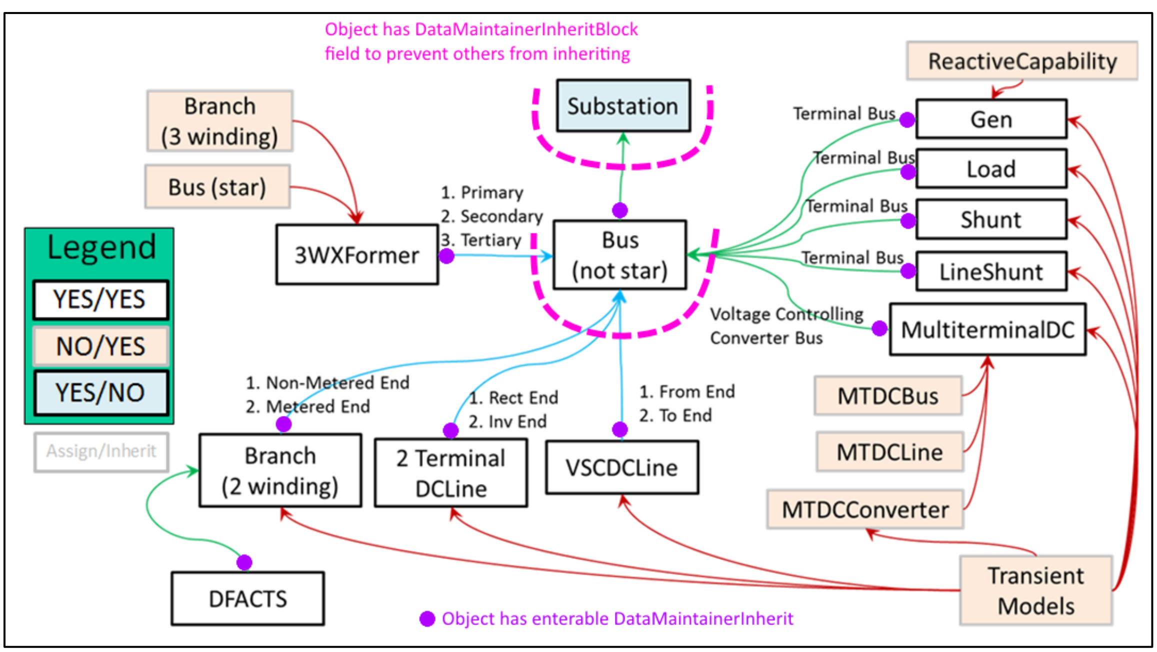
ObjectType	**Assign**	**Inheritance Precedence**
3WXFormer	YES	1. Primary Bus 2. Primary Bus' Substation 3. Secondary Bus 4. Secondary Bus' Substation 5. Tertiary Bus 6. Tertiary Bus' Substation
AreaContingencyReserveBid	NO	Area
AreaOperatingReserveBid	NO	Area
AreaRegulatingReserveBid	NO	Area
Branch	YES	1. NonMetered Bus 2. NonMetered Bus' Substation 3. Metered Bus 4. Metered Bus' Substation (except for windings of a 3WXFormer which inherit from the 3WXFormer)
Bus	YES	Substation (except for star buses of 3WXFormer which inherit from the 3WXFormer)
Condition	NO	Filter
ContingencyElement	NO	Contingency
ContingencyMonitoringException	NO	Contingency
CTGElementBlockElement	NO	CTGElementBlock
DCTransmissionLine	YES	1. Rectifier Bus 2. Rectifier Bus' Substation 3. Inverter Bus 4. Inverter Bus' Substation

DFACTS	YES	1. Branch 2. Branch's NonMetered Bus 3. Branch's NonMetered Bus' Substation 4. Branch's Metered Bus 5. Branch's Metered Bus' Substation
Gen	YES	1. Bus 2. Bus' Substation
GenBid	NO	1. Generator 2. Generator's Bus 3. Generator's Bus' Substation
InterfaceElement	NO	Interface
LineShunt	YES	1. Bus 2. Bus' Substation
Load	YES	1. Bus 2. Bus' Substation
LoadBid	NO	1. Load 2. Load's Bus 3. Load's Bus' Substation
ModelConditionCondition	NO	ModelCondition
ModelFilterCondition	NO	ModelFilter
MTDCBus	NO	1. MTDCRecord 2. MTDCRecord's VConv_Bus 3. MTDCRecord's VConv_Bus's Substation
MTDCConverter	NO	1. MTDCRecord 2. MTDCRecord's VConv_Bus 3. MTDCRecord's VConv_Bus's Substation
MTDCRecord	YES	1. Voltage Controlling Converter Bus 2. Voltage Controlling Converter Bus' Substation
MTDCTransmissionLine	NO	1. MTDCRecord 2. MTDCRecord's VConv_Bus 3. MTDCRecord's VConv_Bus's Substation
PartPoint	NO	3. InjectionGroup
PlayInInfo	NO	PlayIn
PlayInSignal	NO	PlayIn
PVPlotSeries	NO	PVPlot
PVPlotVertAxisGroup	NO	PVPlot
PVSubPlot	NO	PVPlot
ReactiveCapability	NO	1. Gen 2. Gen's Bus 3. Gen's Bus' Substation
RemedialActionElement	NO	RemedialAction
SGPlotSeries	NO	SGPlot
SGPlotVertAxisGroup	NO	SGPlot
SGSubPlot	NO	SGPlot
Shunt	YES	1. Bus

		2. Bus' Substation
StudyMWTransactionsBid	NO	3. StudyMWTransactions
SuperAreaContingencyReserveBid	NO	SuperArea
SuperAreaOperatingReserveBid	NO	SuperArea
SuperAreaRegulatingReserveBid	NO	SuperArea
SupplementalDataContainedObject	NO	MyObject
TSContingencyElement	NO	TSContingency
TSPlotSeries	NO	TSPlot
TSPlotVertAxisGroup	NO	TSPlot
TSSubPlot	NO	TSPlot
VSCDCLine	YES	1. From Bus 2. From Bus' Substation 3. To Bus 4. To Bus' Substation
ZoneContingencyReserveBid	NO	Zone
ZoneOperatingReserveBid	NO	Zone
ZoneRegulatingReserveBid	NO	Zone

3.4.5 Illustration of Objects that Can Inherit a DataMaintainer from Network Topology.

For the more complex relationships obtained from the network topology, the following picture illustrates how this inheritance is handled. Boxes which are shaded in light orange represent ObjectTypes which Always Inherit, while boxes that are not shaded represent ObjectTypes which can have a DataMaintainer assigned. ObjectTypes that have the field DataMaintainerInherit enterable have a purple dot next to their inheritance arrow. ObjectTypes that have the field DataMaintainerInheritBlock have thick dashed pink line around the incoming inheritance arrows.



3.4.6 DataMaintainer being assigned its own DataMaintainer

It is also possible for a DataMaintainer to be assigned to another DataMaintainer. In this way there can be a hierarchy of DataMaintianers. Consider the following made-up loosely based on the WECC system (this is purely hypothetical). For this example we assume that the user of the software tool has ensured that all objects which can belong to DataMaintainer have either been directly assigned a DataMaintianer, or at least an object in its inheritance path had been assigned one.

In this hypothetical, there would be a DataMaintainer at the top of the hierarchy called WECC to which all other DataMaintainers would ultimately belong. Then assume specification of DataMaintainers to other DataMaintainers as follows.

Hypothetical Structure	Representation in an Auxiliary File (Omitting Contact Info)
• WECC	DataMaintainer (Name, DataMaintainerAssign)
o Columbia Grid	{
▪ Avista	"WECC" ""
▪ BPA	"Columbia Grid" "WECC"
▪ Chelan	"Avista" "Columbia Grid"
▪ Grant	"BPA" "Columbia Grid"
▪ PSE	"Chelan" "Columbia Grid"
▪ Seattle	"Grant" "Columbia Grid"
▪ SnoPUD	"PSE" "Columbia Grid"
▪ Tacoma	"Seattle" "Columbia Grid"
o California	"SnoPUD" "Columbia Grid"
▪ SMUD	"Tacoma" "Columbia Grid"
▪ TID	"California" "WECC"
▪ LADWP	"SMUD" "California"
▪ ImperialCA	"TID" "California"
▪ SCE	"LADWP" "California"
▪ SANDIEGO	"ImperialCA" "California"
▪ MEXICO-CFE	"SCE" "California"
o PG&E	"SANDIEGO" "California"
o Arizona	"MEXICO-CFE" "California"
o El Paso	"PG&E" "California"
o New Mexico	"Arizona" "WECC"
o Nevada	"El Paso" "WECC"
	"New Mexico" "WECC"
	"Nevada" "WECC"
	}

If one chooses to “export data that belongs to WECC”, then they would get the entire system. If they choose to “export data that belongs to Chelan”, then they would only get data which has a DataMaintainer of Chelan. Finally, if they choose to “export data that belongs to Columbia Grid”, then they would get any data that has a DataMaintainer of Columbia Grid, Avista, BPA, Chelan, Grant, PSE, Seattle, SnoPUD, or Tacoma. Note: Columbia Grid could be directly assigned as the DataMaintainer of a particular object as well, though in this particular example I would not expect that to be done.

4 Object Definitions

The following sections describe the various AUX file sections and field necessary to replicate all the data necessary to describe power system data as used throughout WECC. Note that the order in which these data sections is read is also of importance. You must read them in this order as the relationships between the objects are ruled by various parent/child relationships. For example, an Area object can belong to a SuperArea, but in order for the portion of the AUX file defining the Area to make this assignment, you must define the SuperArea objects first.

4.1 PWCaseInformation

The PWCaseInformation object is a simple object mostly for providing output summary information about the power system. The information of interest to a power system model is contained in the SUBDATA section PWCaseHeader. This stores the free-form case description information about the model.

Field	Type	Description
Selected	String	Just a dummy field so we can read the case in

For more details on the SUBDATA sections, see the Auxiliary File format description on the PowerWorld website at
http://www.powerworld.com/WebHelp/#http://www.powerworld.com/files/Auxiliary-File-Format.pdf

A sample file section is shown as follows:

```
PWCaseInformation (Selected)
{
"YES"
   <SUBDATA PWCaseHeader>
     //Case Description
      WESTERN ELECTRICITY COORDINATING COUNCIL
      2015 HEAVY SUMMER OPERATING CASE
      DECEMBER 5, 2014
     [pre-title comments]
     # history file date Fri Dec 05 14:05:35 2014
     # present file date Mon May 11 10:44:43 2015
     # Version 18.1_02
     [comments]
      ALL COMMENTS FROM TSS AND OC REVIEW ARE INCLUDED
   </SUBDATA>
}
```

4.2 DataMaintainer

DataMaintainer fields in the base power flow case format are as follows

Field	Type	Description
Name	String	*KEY1* Name of the Data Maintainer. A string of any length is permitted
Contact	String	Contact person’s name

Phone	String	Phone number of contact
Email	String	Email of Contact
Company	String	Company at which contact is employed
Location	String	Location. Could be an address or just the name of a city
DataMaintainerAssign	String	Name of another DataMaintainer to which this DataMaintainer belongs. Normally this value will be blank indicating none, but it could be that the a group of DataMaintainer are contained inside another DataMaintainer. Columbia Grid would be an example of such a group.

A sample file section is shown as follows:

```
DataMaintainer (Name,Contact,Phone,Email,Company,Location,DataMaintainerAssign)
{
"WECC"   "Sally Sue"       "233" "ssue@wecc.com"       "" "" ""
"ColumbiaGrid" "John Olds" "123" "jolds@ColumbiaGrid.org" "" "" "WECC"
"BPA"    "Jackie Johnson"  "123" "jjohnson@bpa.gov"    "" "" "ColumbiaGrid"
"Avista" "Purvis Williams" "567" "purvis@avista.com"   "" "" "ColumbiaGrid"
"Chelan" "Le Xie"          "796" "lexie@chelanpud.org" "" "" "ColumbiaGrid"
}
```

4.3 Owner

Owner fields in the basic power flow case format are as follows. When reading a record that does not contain the primary key (`Number`), but does contain the secondary key (`Name`) with a string that does not describe an existing object, then the `Number` primary key will be automatically set to the maximum `Number` for this object type plus 1.

Field	Type	Description
Number	Integer	*KEY1* Number of the owner. This is the unique identifier for the owner and must be unique in the case. Any integer between 1 and 2,147,483,647 is permitted.
Name	String	*SECONDARY KEY1* Name of the owner. A string of any length is permitted.
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.

A sample file section is shown as follows:

```
Owner (Number, Name, DataMaintainerAssign)
{
 100 "Downtown" "WECC"
 110 "Home"     "WECC"
 140 "Zone 689" "WECC"
}
```

4.4 LoadModelGroup

LoadModelGroup fields in the basic power flow case format are as follows. LoadModelGroups are a container object that only has a name in the power flow format. Each Load object can then be assigned to a particular LoadModelGroup. This container is then used in transient stability input data to assign load models to.

Field	Type	Description
`Name`	String	*KEY1* Name
`DataMaintainerAssign`	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.

A sample file section is shown as follows:

```
LoadModelGroup (Name, DataMaintainerAssign)
{
"AGR" "WECC"
"NWI" "WECC"
"NCC" "WECC"
}
```

4.5 Substation

Substation fields in the basic power flow case format are as follows. When reading a record that does not contain the primary key (`Number`), but does contain the secondary key (`Name`) with a string that does not describe an existing object, then the `Number` primary key will be automatically set to the maximum `Number` for this object type plus 1.

Field	Type	Description
`Number`	Intege r	*KEY1* Number
`Name`	String	*SECONDARY KEY1* Name of the substation. A string of any length is permitted.
`IDExtra`	String	Another string that may contain identifying information. This string can be any length. It may for example be a longer more formal name of the substation.
`Latitude`	Real	Geographic Latitude in decimal degrees. Note: negative values represent the Southern hemisphere. Blank value is also permitted to indicate that the latitude is not known. This should be stored as a double precision floating point number which stores 15-16 significant digits.
`Longitude`	Real	Geographic Longitude in decimal degrees. Note: negative values represent the Western hemisphere. Blank value is also permitted to indicate that the longitude is not known. This should be stored as a double precision floating point number which stores 15-16 significant digits.

DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.
DataMaintainerInheritBlock	String	Default value of this field is NO. Set to YES to prevent all buses in the substation from inheriting their Data Maintainer from this substation. [Added in Version 19, December 1, 2016]

A sample file section is shown as follows:

```
Substation (Number, Name, IDExtra, Latitude, Longitude, DataMaintainerAssign)
{
    5468 "JOHNSVIL" "Johnsonville"  39.099727346589 -94.578567061357 "MISO"
 3469875 "MARZTOWN" "MarzinzikTown" 40.090122012345 -88.239244064897 "MISO"
12344657 "NICOLBRG" "Nicolberg"     45.574904034579 -122.57643304357 "ColumbiaGrid"
}
```

4.6 XFCorrection

XFCorrection fields in the basic power flow case format are as follows.

Field	Type	Description
Number	Integer	*KEY1* Number
Name	String	Name
Tap:0 Tap:1 Tap:2 ... Tap:49	Real	Tap or Phase for the factor lookup
Factor:0 Factor:1 Factor:2 ... Factor:49	Real	Factor multiplier for the factor lookup
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.

A sample file section is shown as follows:

```
XFCorrection (Number,Name,Tap,Factor,Tap:1,Factor:1,Tap:2,Factor:2,
              Tap:3,Factor:3,Tap:4,Factor:4, DataMaintainerAssign)
{
  1 "Tab" -10.00 20.00 -5.00 6.00  0.00 1.00 5.00  6.00 10.00 20.00 "WECC"
  2 "ABC" -45.00  2.07  0.00 1.00 45.00 2.07 ""    ""   ""    ""    "NWU"
  3 "XYZ"  0.875  1.12 1.000 1.00 1.025 0.98 1.175 0.89 ""    ""    "CAISO"
}
```

4.7 VoltageControlGroup

VoltageControlGroup fields in the basic power flow case format are as follows.

Groups of capacitor banks are defined by assigning switched shunt records to a “Voltage Control Group”. When performing capacitor switching during the power flow solution, each voltage control group processes its own list of switched shunts independently as follows.

1. Determine the switched shunt that has the largest deviation below Vlow (call this *LowShunt*)
2. Determine the switched shunt that has the largest deviation above Vhigh (call this *HighShunt*)
3. If *LowShunt* was found, then move that switch shunt UP by one step,
   Else if *HighShunt* was found move this shunt DOWN by one step

It is expected that folks will use this as part of a contingency solution with typically all switched shunt control is disabled in contingency analysis (that's a common practice elsewhere as well), however the Voltage Control Group algorithm will remain active (using the FORCEON Status below).

Field	**Type**	**Description**
`Name`	String	*KEY1* The name of the control group which will be used to refer to it from the switched shunt records.
`Status`	String	This field determines how switched shunt control will behave for this group. ON : Normal behavior where the Control Group acts as described above as long as the global options for moving shunts is enabled. OFF : Means that the control group is ignored and the individual shunts in the group revert back to their own individual control behavior FORCEON : Ignore the global option (or Area record option) to disable switched shunt control and always force control enabled for this group. The FORCEON status makes it easy for the user to disable switched shunt control globally in the contingency analysis tool and then override this disabling by setting the status to FORCEON for the *Voltage Control Group*.
`DataMaintainerAssign`	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.

A sample file section is shown as follows:

```
VoltageControlGroup (Name, Status, DataMaintainerAssign)
{
"EASTERN"  "FORCEON"  "SCE"
"SDG&E"    "FORCEON"  "SCE"
}
```

4.8 Limit_Monitoring_Options_Value

Limit_Monitoring_Options_Value fields in the basic power flow case format are as follows. There is only one option presently for this object and it is LMS_IgnoreRadial.

Field	**Type**	**Description**
`Option`	String	*KEY1* Name of the Option
`Value`	String	Value for the option

A sample file section is shown as follows:

```
Limit_Monitoring_Options_Value (Option,Value)
{
"LMS_IgnoreRadial" "NO"
}
```

4.9 LimitSet

LimitSet records store information about how buses, branches, and interfaces are monitored. Each Bus, Branch, and Interface is assigned to one LimitSet and inherits properties for how it is monitored from the LimitSet. The fields recognized by the LimitSet object type are shown in the following table.

Field	Type	Description
`Name`	String	*KEY1* Name of the LimitSet. This is the unique identifier for the LimitSet so there can only be one LimitSet with this Name.
`Disabled`	String	Set to `YES` to disable all monitoring of devices belonging to this LimitSet. Set to `NO` to enable monitoring
`DataMaintainerAssign`	String	Name of the DataMaintainer for this object. This can be blank as well which indicates no DataMaintainer has been assigned.
Following fields are related to monitoring branches and interfaces		
`AmpMVA`	String	Set to `Amp/MVA` to monitor based on Amp limits on transmission lines and MVA limit on transformers. Set to `MVA` to monitor based on MVA limits on all branches.
`MonitorEnd`	String	Set to either Higher or Lower with a default value of Higher. When monitoring branch flows in software, the percent flow at the From and To end will be different than one another. Higher means that reporting will be based on the higher of the 2 percentages. Lower means that reporting is based on the lower of the 2 percentages
`BranchPercent`	Real	Set to the percentage at which violations will be reported for branches
`InterfacePercent`	Real	Set to the percentage at which violations will be reported for interfaces
`BranchRateSet`	String	The rating set used to monitor branches during in the reference case. Set to either `A, B, C, D, E, F, G, H, I, J, K, L, M, N, O`. See Section 4.9.1.
`InterfaceRateSet`	String	The rating set used to monitor interfaces during in the reference case. Set to either `A, B, C, D, E, F, G, H, I, J, K, L, M, N, O`. See Section 4.9.1.
`BranchRateSetCTG`	String	The rating set used to monitor branches during a contingency solution. Set to either `A, B, C, D, E, F, G, H, I, J, K, L, M, N, O`. See Section 4.9.1.

`InterfaceRateSetCTG`	String	The rating set used to monitor interfaces during a contingency solution. Set to either `A, B, C, D, E, F, G, H, I, J, K, L, M, N, O`. See Section 4.9.1.
Following fields are related to monitoring high and low bus voltages.		
`HighVolt`	Real	Set to the per unit voltage above which is considered a high violation in the the reference case.
`LowVolt`	Real	Set to the per unit voltage below which is considered a low violation in the the reference case.
`HighVoltCTG`	Real	Set to the per unit voltage above which is considered a high violation in the the contingency solution.
`LowVoltCTG`	Real	Set to the per unit voltage below which is considered a low violation in the the contingency solution.
`HighVoltRateSet`	String	The rating set used for high bus voltages during in the reference case. Set to either `A, B, C, D`. These options are only used if a bus is configured to store its own voltage limits, which is not what is most common.
`LowVoltRateSet`	String	The rating set used for low bus voltages during in the reference case. Set to either `A, B, C, D`. These options are only used if a bus is configured to store its own voltage limits, which is not what is most common.
`HighVoltRateSetCTG`	String	The rating set used for high bus voltages during a contingency solution. Set to either `A, B, C, D`. These options are only used if a bus is configured to store its own voltage limits, which is not what is most common.
`LowVoltRateSetCTG`	String	The rating set used for low bus voltages during a contingency solution. Set to either `A, B, C, D`. These options are only used if a bus is configured to store its own voltage limits, which is not what is most common.

A sample file section is shown as follows:

```
LimitSet (Name,Disabled,DataMaintainerAssign,AmpMVA,MonitorEnd,BranchPercent,
   InterfacePercent,BranchRateSet,InterfaceRateSet,BranchRateSetCTG,
   InterfaceRateSetCTG,HighVolt,LowVolt,HighVoltCTG,LowVoltCTG,HighVoltRateSet,
   LowVoltRateSet,HighVoltRateSetCTG,LowVoltRateSetCTG)
{
"Default" "NO " "" "MVA" "Higher" 100.0 100.0 "A" "A" "A" "A" 1.10 0.90 1.10
0.90 "A" "A" "A" "A"
}
```

4.9.1 Branch and Interface relationship to LimitSet

Each branch in the underlying power system data records has 15 limits assigned to it. These are referred to as Limit A, B, C, D, E, F, G, H, I, J, K, L, M, N, and O. The fields associated with the limits are

LimitMVAA, LimitMVAB, LimitMVAC, LimitMVAD, LimitMVAE, LimitMVAF, LimitMVAG, LimitMVAH, …, LimitMVAN, and LimitMVAO.

Similarly, each interface has 8 limits (A-H) assigned to it. The fields associated with the limits are LimitMWA, LimitMWB, LimitMWC, LimitMWD, LimitMWE, LimitMWF, LimitMWG, LimitMWH, …, LimitMWN, and LimitMWO.

The LimitSet has various fields that specify which rating set to use when monitoring in the contingency reference case or during the post-contingency monitoring. These refer to these A – H limits.

4.9.2 Bus relationship to LimitSet

Traditionally, power flow data records have not stored voltage ratings with each individual bus record. As a result, the typical way that high and low bus limits are assigned to a bus is by assigning the bus to a LimitSet and then configuring the LimitSet fields HighVolt/LowVolt and HighVoltCTG/LowVoltCTG appropriately to assign the High/Low voltage limits during the reference case and post-contingency monitoring respectively.

It is also possible to assign 4 sets of high and low per unit voltage limits for each bus (call them High A, B, C, and D and Low A, B, C, and D). With each bus record, there is then a flag called "Use bus-specific limits". If this flag is set to YES, the bus limits will be determined by using the LimitSet fields that refer to a "RateSet" in much the same way as is done for Branch and Interface limits as described in Section 4.9.1. The fields associated with these bus-specific limits are as follows: UseSpecificLimits, LimitHighA, LimitHighB, LimitHighC, LimitHighD, LimitLowA, LimitLowB, LimitLowC, and LimitLowD.

4.10 RatingSetNameBranch

RatingSetNameBranch fields in the basic power flow case format are as follows. A special note about these objects is that they cannot be either created or deleted by the user. There are always 15 RatingSetNameBranch objects named A, B, C, D, E, F, G, H, I, J, K, L, M, N, and O.

Field	Type	Description
`Name`	String	*KEY1* Name of the rating set. Values are always A, B, C, D, E, F, G, H, I, J, K, L, M, N, O
`Header`	String	String indicating what is shown in the column header for a case information display showing the particular rating set
`Description`	String	String indicating what is shown in other locations or when showing a header hint on a case information display.

A sample file section is shown as follows:

```
RatingSetNameBranch (Name,Header,Description)
{
"A" "Summer Norm" "Summer Normal Rating"
"B" "Summer CTG" "Summer Contingency Rating"
"C" "Summer Emer" "Summer Emergency Rating"
"D" "Winter Norm" "Winter Normal Rating"
"E" "Winter CTG" "Winter Contingency Rating"
"F" "Winter Emer" "Winter Emergency Rating"
```

```
"G" "Fall Norm" "Fall Normal Rating"
"H" "Fall CTG" "Fall Contingency Rating"
"I" "Fall Emer" "Fall Emergency Rating"
"J" "Spring Norm" "Spring Normal Rating"
"K" "Spring CTG" "Spring Contingency Rating"
"L" "Spring Emer" "Spring Emergency Rating"
"M" "Norm" "Normal Rating"
"N" "CTG" "Contingency Rating"
"O" "Emer" "Emergency Rating"
}
```

4.11 RatingSetNameInterface

RatingSetNameInterface fields in the basic power flow case format are as follows. A special note about these objects is that they cannot be either created or deleted by the user. There are always 15 RatingSetNameInterface objects named A, B, C, D, E, F, G, H, I, J, K, L, M, N, and O.

Field	Type	Description
Name	String	*KEY1* Name of the rating set. Values are always A, B, C, D, E, F, G, H, I, J, K, L, M, N, O
Header	String	String indicating what is shown in the column header for a case information display showing the particular rating set
Description	String	String indicating what is shown in other locations or when showing a header hint on a case information display.

A sample file section is shown as follows:

```
RatingSetNameInterface (Name,Header,Description)
{
"A" "Summer Norm" "Summer Normal Rating"
"B" "Summer CTG"  "Summer Contingency Rating"
"C" "Summer Emer" "Summer Emergency Rating"
"D" "Winter Norm" "Winter Normal Rating"
"E" "Winter CTG"  "Winter Contingency Rating"
"F" "Winter Emer" "Winter Emergency Rating"
"G" "Fall Norm"   "Fall Normal Rating"
"H" "Fall CTG"    "Fall Contingency Rating"
"I" "Fall Emer"   "Fall Emergency Rating"
"J" "Spring Norm" "Spring Normal Rating"
"K" "Spring CTG"  "Spring Contingency Rating"
"L" "Spring Emer" "Spring Emergency Rating"
"M" "Norm"        "Normal Rating"
"N" "CTG"         "Contingency Rating"
"O" "Emer"        "Emergency Rating"
}
```

4.12 RatingSetNameBus

RatingSetNameBus fields in the basic power flow case format are as follows. A special note about these objects is that they cannot be either created or deleted by the user. There are always 4 RatingSetNameInterface objects named A, B, C, and D.

Field	Type	Description
Name	String	*KEY1* Name of the rating set. Values are always A, B, C, D

Header	String	String indicating what is shown in the column header for a case information display showing the particular rating set
Description	String	String indicating what is shown in other locations or when showing a header hint on a case information display.

A sample file section is shown as follows:

```
RatingSetNameBus (Name,Header,Description)
{
"A" "Normal"    "Normal Rating"
"B" "CTG"       "Contingency Rating"
"C" "Emergency" "Emergency Rating"
"D" "Other"     "Other Rating"
}
```

4.13 Bus

Bus fields in the basic power flow case format are as follows. When reading a record that does not contain the primary key (`Number`), but does contain the secondary key (`NameNomkV`) with a string that does not describe an existing object, then the `Number` primary key will be automatically set to the maximum `Number` for this object type plus 1.

Field	Type	Description
Number	Integer	*KEY1* Number
NameNomkV	String	*SECONDARY KEY1* This is a concatenation of the Name field, followed by the underscore (_) character, followed by the NomkV expressed to 4 significant digits. Thus you might see "MyName_69.60", "MyName_138.0", "MyName_0.2100". When reading using this field we follow the conventions described in Section 2.3.2.
Name	String	Name
NomkV	Real	The nominal kv voltage specified as part of the input file.
Slack	String	YES or NO. Set to YES to indicate that this bus should be the island slack bus.
NomG	Real	Nominal MW from extra shunt admittance at the bus (Mvar when operating at 1.0 per unit voltage). Positive values represent *load*. This is meant to represent fictitious injections such as created by an equivalencing routine or the state estimator mismatch as read from a state estimator solution.
NomB	Real	Nominal Mvar from extra shunt admittance at the bus (Mvar when operating at 1.0 per unit voltage). Positive values represent *generation*. This is meant to represent fictitious injections such as created by an equivalencing routine or the state estimator mismatch as read from a state estimator solution.
Vpu	Real	The per unit voltage magnitude. A value of 1.0 means

		the actual kV is equal to the nominal kV
`Vangle`	Real	Voltage: Angle (degrees)
`DCLossMultiplier`	Real	Only used when solving a DC power flow using the DC approximation solution option. This then specifies a multiplier at the bus used during the DC power flow solution. All loads at the bus will be artificially increased by this multiplier when calculating load MWs during the DC power flow.
`AreaNumber`	Integer	Number of the Area. Must be a positive integer value. Must be specified, so blank values are not permitted. When reading this field, if an area does not already exist with this number, then a new area will automatically be created.
`ZoneNumber`	Integer	Number of the Zone. Must be a positive integer value. Must be specified, so blank values are not permitted. When reading this field, if a zone does not already exist with this number, then a new zone will automatically be created.
`BANumber`	Integer	Number of the Balancing Authority. Must be a positive integer value. Must be specified, so blank values are not permitted. When reading this field, if a balancing authority does not already exist with this number, then a new balancing authority will automatically be created.
`OwnerNumber`	Integer	Number of the Owner to which the bus is assigned
`SubNumber`	Integer	Substation Number. Must be a positive integer value, however a blank value is permitted to indicate that the substation to which the bus belong is not known. When reading this field, if a substation does not already exist with this number, then a new zone will automatically be created.
`Monitor`	String	Set to `YES` to specify that this bus should be monitored. Set to `NO` to not monitor this bus.
`LimitSet`	String	Name of the Limit Set to which the bus belongs
`UseSpecificLimits`	String	Set to YES to specify specific limits for this bus in the format. When set to NO the limits will be obtained from the LimitSet objects instead
`LimitLowA`	Real	A low voltage limit of the bus in per unit
`LimitLowB`	Real	B low voltage limit of the bus in per unit
`LimitLowC`	Real	C low voltage limit of the bus in per unit
`LimitLowD`	Real	D low voltage limit of the bus in per unit
`LimitHighA`	Real	A high voltage limit of the bus in per unit
`LimitHighB`	Real	B high voltage limit of the bus in per unit
`LimitHighC`	Real	C high voltage limit of the bus in per unit
`LimitHighD`	Real	D high voltage limit of the bus in per unit

Latitude	Real	Geographic Latitude in decimal degrees. Note: negative values represent the southern hemisphere. Blank value is also permitted to indicate that the latitude is not known. This should be stored as a double precision floating point number which stores 15-16 significant digits.
Longitude	Real	Geographic Longitude in decimal degrees. Note: negative values represent the western hemisphere. Blank value is also permitted to indicate that the longitude is not known. This should be stored as a double precision floating point number which stores 15-16 significant digits.
TopologyBusType	String	Type of electrical connection point. Choices are BusBarSection, Junction, Internal_3WND, and Ground.
Priority	Integer	Integer priority used when choosing the primary node within a Superbus. Higher numbers have priority over lower numbers.
EMSType	String	Record type read from an EMS system
EMSID	String	String ID for node used in EMS systems
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank. *Note: For the internal bus of a three-winding transformer, this value cannot be specified and any entry here will be ignored. The DataMaintainer will always be inherited from the three-winding transformer record.*
DataMaintainerInherit	String	Default value of this field is YES. Set to NO to prevent the bus from inheriting its DataMaintainer from its substation [Added in Version 19, December 1, 2016]
DataMaintainerInheritBlock	String	Default value of this field is NO. Set to YES to prevent other objects from inheriting their Data Maintainer from this bus. For instance if this is YES, then any generator connected to this bus will not inherit it's DataMaintainer from the bus. [Added in Version 19, December 1, 2016]
AllLabels	String	A comma-delimited list of unique label identifiers for this object. The syntax for this field is described in detail in Section 2.3.4.

A sample file section is shown as follows:

```
Bus (Number,Name,NomkV,Slack,NomB,NomG,Vpu,Vangle,DCLossMultiplier,AreaNumber,
   ZoneNumber,BANumber,OwnerNumber,SubNumber,Monitor,LimitSet,UseSpecificLimits,
   LimitLowA,LimitLowB,LimitLowC,LimitLowD,LimitHighA,LimitHighB,LimitHighC,
   LimitHighD,Latitude,Longitude,TopologyBusType,Priority,EMSType,EMSID,
   DataMaintainerAssign,AllLabels)
{
```

```
10000 "MyBus0"  46.0000 "NO " 0.0 0.0 1.00024018 11.3919108 1.00 10 105 10 129 "" "YES"
"Default" "NO " "" "" "" "" "" "" "" "" "" "" "BusbarSection" 0 "" "" "" ""

10004 "MyBus4" 115.0000 "NO " 0.0 0.0 1.04261534  8.1979272 1.00 10 101 10 129 "" "YES"
"Default" "NO " "" "" "" "" "" "" "" "" "" "" "BusbarSection" 0 "" "" "" ""
}
```

4.14 Gen

Gen fields in the basic power flow case format are as follows.

Field	Type	Description
BusNum	Integer	*KEY1* Number of the bus. When reading record, see special note in Section 2.4.
ID	String[2]	*KEY2* 2 character generator identification field. Used to identify multiple generators at a single bus
Status	String	The status of the generator (Open or Closed)
VoltSet	Real	Desired per unit voltage setpoint at the regulated bus
RegBusNum	String	Number of regulated bus
RegFactor	Real	Remote regulation factor. When multiple buses have generation that control the voltage at a single bus, this determines the ratio in which the Mvar output is shared.
AGC	String	Set to YES or NO to specify whether or not generator is available for AGC
PartFact	Real	Generator's participation factor. Used during Area Interchange Control when set to AGC is set to Part AGC. Also used during post-contingency make-up power. Also used for sensitivity calculations when using Areas, Zones, or Super Areas.
MWSetPoint	Real	This is what the generator's MW output is if it is presently inservice. If the generator is inservice this is the same as the MW field, however if the generator is out of service then the MW field would return 0.0.
MWMax	Real	Generator's maximum MW limit
MWMin	Real	Generator's minimum MW limit
MWMaxEcon	Real	Generator's maximum MW limit
MWMinEcon	Real	Generator's minimum MW limit
EnforceMWLimit	String	Set to YES to specify whether or not generator's MW limits are enforced
AVR	String	Set to YES or NO to specify whether or not generator is available for AVR
MvarSetPoint	Real	This is what the generator's Mvar output is if it is presently inservice. If the generator is inservice this is the same as the Mvar field, however if the generator is out of service then the MW field would return 0.0.
MvarMax	Real	Generator's maximum Mvar limit
MvarMin	Real	Generator's minimum Mvar limit

UseCapCurve	String	Indicates whether or not the generator should use its Mvar capability curve if it has one defined.
WindContMode	String	Special Var limit modes of either "None", "Boundary Power Factor" or "Constant Power Factor". When not equal to None, the Var limit magnitudes are determined from the real power output and the Wind Control Mode Power Factor value. For Boundary mode, the maximum limit is positive and the minimum limit is negative. For Constant mode, minimum limit = maximum limit, a positive Wind Control Mode Power Factor means the limits have the same sign as the real power, and a negative Wind Control Mode Power Factor means the limits are the opposite sign as the real power.
WindContModePF	Real	This is the power factor value used with the Wind Control Mode. Magnitude of the value must be between 0.01 and 1.00. Negative values are important when the Wind Control Mode is "Constant Power Factor".
UseLineDrop	String	Field describing whether or not the generator uses line drop/reactive current compensation control
Rcomp	Real	Generator's Line Drop Compensation resistance in per unit on the system MVA Base
Xcomp	Real	Generator's Line Drop Compensation reactance in per unit on the system MVA Base
MVABase	Real	Generator's MVA base
GenR	Real	Machine Internal Resistance in per unit on Generator MVA Base
GenZ	Real	Machine Internal Reactance in per unit on Generator MVA Base
StepR	Real	Internal Step up: R (resistance)
StepX	Real	Internal Step up: X (reactance)
StepTap	Real	Internal Step up: Tap Ratio
GovRespLimit	String	Specifies how governors respond in transient stability simulation. The choices are `Normal`, `Down Only`, or `Fixed`
UnitTypeCode	String	Two character field describing what kind of machine the generator is. The choices are informed by the Energy Information Agency of US Department of Energy. There is an EIA Form 860 and 923 for the Annual Electric Generator Report. The choices are the first two characters of the following list. Also note that in square brackets are the integer code that will be written to an EPC file. UN (Unknown) [0] BA (Energy Storage, Battery) [42] BT (Turbines Used in a Binary Cycle, including those used for geothermal applications) [19]

		CA (Combined Cycle Steam Part) [2] CC (Combined Cycle Generic) [4] CE (Compressed Air Storage) [46] CP (Energy Storage, Concentrated Solar Power) [47] CS (Combined Cycle Single Shaft) [13] CT (Combined Cycle Combustion Turbine Part) [29] DC (represents DC ties) [40] ES (Energy Storage, Other) [44] FC (Fuel Cell) [99] FW (Energy Storage, Flywheel) [43] GT (Gas Turbine) [11] HA (Hydrokinetic, Axial Flow Turbine) [51] HB (Hydrokinetic, Wave Buoy) [52] HK (Hydrokinetic, Other) [53] HY (Hydro) [5] IC (Internal Combustion) [6] IT (Internal Combustion Turbo Charged) [7] JE (Jet Engine) [12] MP (Motor/Pump) [41] NB (ST - Boiling Water Nuclear Reactor) [1 ST] NG (ST - Graphite Nuclear Reactor) [1 ST] NH (ST - High Temperature Gas Nuclear Reactor)[1 ST] NP (ST - Pressurized Water Nuclear Reactor) [1 ST] OT (Other) [99] P3 (Photovoltaic,, Fixed) [32] P2 (Photovoltaic, DC Coupled No Storage) [34] P4 (Photovoltaic, Tracking) [33] PS (Hydro Pumped Storage) [54] PV (Photovoltaic) [31] SC (Synchronous Condenser) [14] ST (Steam Turbine) [1] SV (Static Var Compensator) [99 OT] W1 (Wind Turbine, Type 1) [21] W2 (Wind Turbine, Type 2) [22] W3 (Wind Turbine, Type 3) [23] W4 (Wind Turbine, Type 4) [24] WS (Wind Turbine, Offshore) [25] WT (Wind Turbine) [20] XC (Cross Compound Steam) [3]
FuelTypeCode	String	[Added in Version 21, December 11, 2019] Two or three character field describing what fuel type the generator uses. The choices are informed by the Energy Information Agency of US Department of Energy. There is an EIA Form 860 and 923 for the Annual Electric Generator Report. The choices are the first two characters of the following list. Also note that in square brackets are the integer code that will be written to an EPC file. UN (Unknown) [0] ANT (Anthracite Coal) [10] BIT (Bituminous Coal) [11] LIG (Lignite Coal) [12] SGC (Coal-Derived Synthesis Gas) [13] SUB (Subbituminous Coal) [14] WC (Waste/Other Coal) [15] RC (Refined Coal) [16] DFO (Distillate Fuel Oil) [20] JF (Jet Fuel) [21] KER (Kerosene) [22] PC (Petroleum Coke) [23] PG (Gaseous Propane) [24] RFO (Residual Fuel Oil) [25] SGP (Synthesis Gas from Petroleum Coke) [26] WO (Waste/Other Oil) [27] BFG (Blast Furnace Gas) [30]

		NG (Natural Gas) [31] OG (Other Gas) [32] AB (Agricultural By-Products) [40] MSW (Municipal Solid Waste) [41] OBS (Other Biomass Solids) [42] WDS (Wood/Wood Waste Solids) [43] OBL (Other Biomass Liquids) [50] SLW (Sludge Waste) [51] BLQ (Black Liquor) [52] WDL (Wood Waste Liquids excluding Black Liquor) [53] LFG (Landfill Gas) [60] OBG (Other Biomass Gas) [61] SUN (Solar) [70] WND (Wind) [71] GEO (Geothermal) [72] WAT (Water) [73] WPS (Water Pumped Storage) [74] NUC (Nuclear) [80] PUR (Purchased Steam) [81] WH (Waste Heat) [82] TDF (Tire Derived Fuels) [83] MWH (Electricity use for Energy Storage) [84] OTH (Other) [85]
AreaNumber	Integer	It is possible for the terminal bus to belong to a different area than the device belongs. This is the Area number of the Generator. When reading this field, if an area does not already exist with this number, then a new area will automatically be created.
ZoneNumber	Integer	It is possible for the terminal bus to belong to a different zone than the device belongs. This is the Zone number of the Generator. When reading this field, if a zone does not already exist with this number, then a new zone will automatically be created.
BANumber	Integer	It is possible for the terminal bus to belong to a different balancing authority than the device belongs. This is the Balancing Authority number of the Generator. When reading this field, if a balancing authority does not already exist with this number, then a new balancing authority will automatically be created.
OwnerNum1	Integer	Owner Number 1
OwnerPerc1	Real	Owner 1
OwnerNum2	Integer	Owner Number 2
OwnerPerc2	Real	Owner 2
OwnerNum3	Integer	Owner Number 3
OwnerPerc3	Real	Owner 3
OwnerNum4	Integer	Owner Number 4
OwnerPerc4	Real	Owner 4
OwnerNum5	Integer	Owner Number 5
OwnerPerc5	Real	Owner 5
OwnerNum6	Integer	Owner Number 6
OwnerPerc6	Real	Owner 6
OwnerNum7	Integer	Owner Number 7

OwnerPerc7	Real	Owner 7
OwnerNum8	Integer	Owner Number 8
OwnerPerc8	Real	Owner 8
EMSType	String	Record type read from an EMS system
EMSID	String	String ID for generator used in EMS systems
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.
DataMaintainerInherit	String	Default value of this field is YES. Set to NO to prevent the object from inheriting its DataMaintainer from another object. [Added in Version 19, December 1, 2016]
AllLabels	String	A comma-delimited list of unique label identifiers for this object. The syntax for this field is described in detail in Section 2.3.4.

A sample file section is shown as follows:

```
Gen (BusNum,ID,Status,VoltSet,RegBusNum,RegFactor,AGC,PartFact,MWSetPoint,
  MWMax,MWMin,EnforceMWLimit,AVR,Mvar,MvarMax,MvarMin,UseCapCurve,
  WindContMode,WndContModePF,UseLineDrop,Rcomp,Xcomp,MVABase,GenR,GenZ,
  StepR,StepX,StepTap,GovRespLimit,UnitTypeCode,AreaNumber,ZoneNumber,BANumber,
  OwnerNum1,OwnerPerc1,OwnerNum2,OwnerPerc2,OwnerNum3,OwnerPerc3,OwnerNum4,OwnerPerc4,
  OwnerNum5, OwnerPerc5,OwnerNum6,OwnerPerc6,OwnerNum7,OwnerPerc7,OwnerNum8,OwnerPerc8,
  EMSType,EMSID,DataMaintainerAssign,AllLabels)
{
10112 "PV" "Closed" 1.00000  10112 100.00000 "YES"    5.00000    3.30000    5.00000
0.00000 "YES" "NO "    0.00000    0.00000    0.00000 "NO " "None" 1.000000 "NO"
0.000000 0.000100   5.0000 0.000000 0.277000 0.000000 0.000000 1.000000 "Normal"    "UN"
10    100 10    130 100.000 "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""

 10131 "PV" "Closed" 1.00000  10131 100.00000 "YES"    5.00000    3.30000    5.00000
0.00000 "YES" "NO "    0.00000    0.00000    0.00000 "NO " "None" 1.000000 "NO"
0.000000 0.000100   5.0000 0.000000 0.277000 0.000000 0.000000 1.000000 "Normal"    "UN"
10    100 10    130 100.000 "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
}
```

4.15 ReactiveCapability

Bus fields in the basic power flow case format are as follows.

Field	Type	Description
Number	Integer	*KEY1* Number of the bus. When reading record, see special note in Section 2.4.
ID	String[2]	*KEY2* 2 character generator identification field. Used to identify multiple generators at a single bus
MW	Real	*KEY3* MW output at which MvarMin and MvarMax are specified (rounded to the nears 0.001 MW)
MvarMax	Real	Maximum Mvar output at respective MW output
MvarMin	Real	Minimum Mvar output at respective MW output

A sample file section is shown as follows:

```
ReactiveCapability (BusNum, ID, MW, MvarMin, MvarMax)
```

```
{
10246 "1 "  80.000 -79.670 107.960
10246 "1 "  84.000 -78.610 106.620
10246 "1 "  88.000 -77.480 105.210
10246 "1 "  92.000 -74.960 103.720
10246 "1 "  96.000 -74.950 102.170
10246 "1 " 100.000 -73.570 100.550
10246 "1 " 104.000 -72.100  98.860
10246 "1 " 108.000 -70.540  97.090
10246 "1 " 112.000 -68.900  95.260
10246 "1 " 116.000 -67.190  93.360
10246 "1 " 120.000 -65.380  91.310
10246 "1 " 124.000 -63.500  89.330
10246 "1 " 128.000 -61.530  87.220
10246 "1 " 132.000 -59.470  85.030
}
```

4.16 Load

Load fields in the basic power flow case format are as follows.

Field	Type	Description
`BusNum`	Integer	*KEY1* Number of the bus. When reading record, see special note in Section 2.4.
`ID`	String[2]	*KEY2* 2 character load identification field. Used to identify multiple loads at a single bus
`Status`	String	The status of the load (Open or Closed)
`AGC`	String	Set to YES to permit this load to be automatically controlled by various software tools
`SMW`	Real	Constant Real Power in MW
`SMvar`	Real	Constant Reactive Power in Mvar
`IMW`	Real	Constant Current Real Power in nominal MW (linearly dependent on per unit voltage)
`IMvar`	Real	Constant Current Reactive Power in nominal Mvar (linearly dependent on per unit voltage)
`ZMW`	Real	Constant Impedance Real Power in nominal MW (dependent on square of per unit voltage)
`ZMvar`	Real	Constant Impedance Reactive Power in nominal Mvar (linearly on square of per unit voltage)
`DistStatus`	String	Status of the Distributed Generation associated with the load record (OPEN or CLOSED)
`DistMWInput`	Real	Constant MW of the distributed generation associated with the load record
`DistMvarInput`	Real	Constant Mvar of the distributed generation associated with the load record
`Interruptible`	String	Either YES or NO. Presently this field is informational only.
`MWMax`	Real	When automatically dispatched this is the maximum MW demand of the load.
`MWMin`	Real	When automatically dispatched this is the minimum MW demand of the load.

DistMWMax	Real	The maximum DistMWInput expected for this load [Added in Version 21]
DistMWMin	Real	The minimum DistMWInput expected for this load (negative values indicate a storage device such as a battery) [Added in Version 21]
DistUnitTypeCode	String	Two-Character Field describing what kind of machine the generation is. See documentation of the generator field UnitTypeCode as the choices for this field are identical to that field. [Added in Version 21]
LoadModelGroup	String	Name of the LoadModelGroup to which the load belongs
AreaNumber	Integer	Number of Area to which the load is assigned. This can be different than the area of the terminal bus. When reading this field, if an area does not already exist with this number, then a new area will automatically be created.
ZoneNumber	Integer	Number of Zone to which the load is assigned. This can be different than the area of the terminal bus. When reading this field, if a zone does not already exist with this number, then a new zone will automatically be created.
BANumber	Integer	Number of Balancing Authority to which the load is assigned. This can be different than the area of the terminal bus. When reading this field, if a balancing authority does not already exist with this number, then a new balancing authority will automatically be created.
OwnerNumber	Integer	Number of the Owner to which the load is assigned
EMSType	String	Record type read from an EMS system
EMSID	String	String ID for load used in EMS systems
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.
DataMaintainerInherit	String	Default value of this field is YES. Set to NO to prevent the object from inheriting its DataMaintainer from another object. [Added in Version 19, December 1, 2016]
AllLabels	String	A comma-delimited list of unique label identifiers for this object. The syntax for this field is described in detail in Section 2.3.4.

A sample file section is shown as follows:

```
Load (BusNum,ID,Status,AGC,SMW,SMvar,IMW,IMvar,ZMW,ZMvar,DistStatus,DistMWInput,
  DistMvarInput,Interruptible,MWMax,MWMin,DistMWMax,LoadModelGroup,AreaNumber,ZoneNumber,
  BANumber,OwnerNumber,EMSType,EMSID,DataMaintainerAssign,AllLabels)
{
 10005 "1 " "Closed" "YES"   17.64932   -1.70899    0.00000    0.00000    0.00000
0.00000 "Closed"    0.00000    0.00000 "NO "   17.64932    0.00000   10.00000 "HID4"
10    101 10    129 "" "" "" ""

 10008 "TS" "Closed" "YES"    6.30000    1.80000    0.00000    0.00000    0.00000
0.00000 "Closed"    0.00000    0.00000 "NO "    6.30000    0.00000   10.00000 "HID3"
10    108 10    127 "" "" "" ""
```

```
 10013 "1 " "Closed" "YES"    9.17447   -1.33055    0.00000    0.00000    0.00000
0.00000 "Closed"    0.00000    0.00000 "NO "    9.17447    0.00000   10.00000 "HID4"
10    101 10    129 "" "" "" ""
}
```

4.17 Branch (Not a Transformer)

Branch fields for a non-transformer branch in the basic power flow case format are as follows.

Field	Type	Description
`BusNumFrom`	Integer	*KEY1* Number of the from bus. When reading record, see special note in Section 2.4.
`BusNumTo`	Integer	*KEY2* Number of the To bus. When reading record, see special note in Section 2.4.
`Circuit`	String[2]	*KEY3* Two character ID of the branch. This identifier must be unique regardless of whether the branch is a transformer or a non-transformer.
`BranchDeviceType`	String	Field can have the following entries: `Line`, `Transformer`, `Series Cap`, `Breaker`, `Disconnect`, `ZBR`, `Fuse`, `Load Break Disconnect`, or `Ground Disconnect`. This enumeration of device types comes from the Common Information Model (CIM) specification, except that a `ZBR` is called a Jumper in CIM. In general a user may toggle between these various device types, with the exception of a Transformer. Once an object is specified as a transformer it may not be turned back into another branch device type.
`ConsolidateAllow`	String	YES or NO. Set to YES to allow this branch to be consolidated in the integrated topology processing
`Status`	String	Status of the branch. Set to either OPEN or CLOSED
`StatusNormal`	String	Normal status of the branch. Set to either OPEN or CLOSED
`ByPass`	String	Set to YES to bypass the branch and treat it as a minimum series impedance (0.0000001 + j0.00001)
`MeteredEnd`	String	Specify either FROM or TO. Represents the end of the transmission line which is metered when used as a tie-line between areas, zones, or balancing authorities. The end of the line which is not metered will be responsible for the losses on the line.
`R`	Real	Series Resistance of the branch in per unit on the system MVABase and the nominal kV of the from bus
`X`	Real	Series Reactance of the branch in per unit on the system MVABase and the nominal kV of the from bus
`B`	Real	Shunt Susceptance of the branch in per unit on the system MVABase and the nominal kV of the from bus
`G`	Real	Shunt Conductance of the branch in per unit on the system MVABase and the nominal kV of the from bus

LineLength	Real	Length of the line. Number does not have unit associated with it.
Monitor	String	Set to YES to specify that this branch should be monitored. Set to NO to not monitor this branch.
LimitSet	String	Name of the Limit Set to which the branch belongs
LimitMVAA	Real	A Rating of the branch in MVA
LimitMVAB	Real	B Rating of the branch in MVA
LimitMVAC	Real	C Rating of the branch in MVA
LimitMVAD	Real	D Rating of the branch in MVA
LimitMVAE	Real	E Rating of the branch in MVA
LimitMVAF	Real	F Rating of the branch in MVA
LimitMVAG	Real	G Rating of the branch in MVA
LimitMVAH	Real	H Rating of the branch in MVA
LimitMVAI	Real	I Rating of the branch in MVA
LimitMVAJ	Real	J Rating of the branch in MVA
LimitMVAK	Real	K Rating of the branch in MVA
LimitMVAL	Real	L Rating of the branch in MVA
LimitMVAM	Real	M Rating of the branch in MVA
LimitMVAN	Real	N Rating of the branch in MVA
LimitMVAO	Real	O Rating of the branch in MVA
OwnerNum1	Integer	Owner Number 1. May also be listed as blank to indicate that the owner is the same as the owner of from bus.
OwnerPerc1	Real	Owner 1 Percent
OwnerNum2	Integer	Owner Number 2
OwnerPerc2	Real	Owner 2
OwnerNum3	Integer	Owner Number 3
OwnerPerc3	Real	Owner 3
OwnerNum4	Integer	Owner Number 4
OwnerPerc4	Real	Owner 4
OwnerNum5	Integer	Owner Number 5
OwnerPerc5	Real	Owner 5
OwnerNum6	Integer	Owner Number 6
OwnerPerc6	Real	Owner 6
OwnerNum7	Integer	Owner Number 7
OwnerPerc7	Real	Owner 7
OwnerNum8	Integer	Owner Number 8
OwnerPerc8	Real	Owner 8
EMSType	String	Record type read from an EMS system
EMSID	String	String ID for branch used in EMS systems
EMSLineID	String	String ID for group container in EMS system

EMSCBTyp	String	String ID for switch type in EMS System
EMSID2From	String	String ID for the from bus side measurement object in EMS System
EMSID2To	String	String ID for the to bus side measurement object in EMS System
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.
DataMaintainerInherit	String	Default value of this field is YES. Set to NO to prevent the object from inheriting its DataMaintainer from another object. [Added in Version 19, December 1, 2016]
AllLabels	String	A comma-delimited list of unique label identifiers for this object. The syntax for this field is described in detail in Section 2.3.4.

A sample file section is shown as follows:

```
Branch (BusNumFrom,BusNumTo,Circuit,BranchDeviceType,ConsolidateAllow,Status,
 StatusNormal,ByPass,MeteredEnd,R,X,B,G,LineLength,Monitor,LimitSet,LimitMVAA,
 LimitMVAB,LimitMVAC,LimitMVAD,LimitMVAE,LimitMVAF,LimitMVAG,LimitMVAH,
 LimitMVAI,LimitMVAJ,LimitMVAK,LimitMVAL,LimitMVAM,LimitMVAN,LimitMVAO,
 OwnerNum1,OwnerPerc1,OwnerNum2,OwnerPerc2,OwnerNum3,OwnerPerc3,OwnerNum4,
 OwnerPerc4,OwnerNum5,OwnerPerc5,OwnerNum6,OwnerPerc6,OwnerNum7,OwnerPerc7,
 OwnerNum8,OwnerPerc8,EMSType,EMSID,EMSLineID,EMSCBTyp,EMSID2From,EMSID2To,
 DataMaintainerAssign,AllLabels,CustomFloat:0)
{
 10026 10 "1" "Line" "" "Closed" "Closed" "NO" "To" 0.02160 0.03160 0.000120
0.0 1.00 "YES" "Default" 27.0 27.0 27.0 27.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 129 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
"" "" "" 0.0

 10356 10 "1" "Line" "" "Closed" "Closed" "NO" "To" 0.02390 0.03750 0.000120
0.0 1.10 "YES" "Default" 27.0 27.0 27.0 27.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 129 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
"" "" "" 0.0
}
```

4.18 Branch (Transformer)

Branch fields for a two-winding transformer branch in the basic power flow case format are as follows.

Field	Type	Description
BusNumFrom	Integer	*KEY1* Number of the from bus
BusNumTo	Integer	*KEY2* Number of the To bus
Circuit	String[2]	*KEY3* Two character ID of the branch. This identifier must be unique regardless of whether the branch is a transformer or a non-transformer.
BranchDeviceType	String	Will always be Transformer
Status	String	Status of the branch. Set to either OPEN or CLOSED
StatusNormal	String	Normal status of the branch. Set to either OPEN or CLOSED

`ByPass`	String	Set to YES to bypass the branch and treat it as a minimum series impedance (0.0000001 + j0.00001). Note this should not be used with a Transformer Branch, but in some circumstances while using software it may be convenient to have this field.
`MeteredEnd`	String	End of the transmission line which is metered when used as a tie-line between areas, zones, or balancing authorities. The end of the line which is not metered will be responsible for the losses on the line.
`ControlType`	String	Control Type of the transformer. Choices are Fixed, LTC, Mvar, Phase. LTC means that a voltage in controlled by moving the transformer tap. Mvar means that the Mvar flow on the branch is controlled by moving the tap ratio. Phase means that the MW flow on the branch is controlled by changing the phase shift angle.
`AutoControl`	String	Set to either YES or NO to indicate whether automatic transformer control is available for this branch. If the ControlType = Phase, then the choice OPF is also available to indicate that the phase angle can be an OPF control variable.
`RegBusNum`	String	Regulated Bus Number. Only used for the ControlType = LTC
`UseLineDrop`	String	Set to either `YES` or `NO` `NO` : Use normal voltage control based on the regulated bus and voltage setpoint. `YES` : Use line drop compensation voltage control always, including in the power flow solution. In order to use line drop compensation, the regulated bus must be one of the terminals of the branch. The line drop is then calculated looking out from that branch into the rest of the system.
`Rcomp`	Real	Line Drop Compensation resistance value used during a contingency power flow solution. Value will be expressed in per unit on the system MVA base.
`Xcomp`	Real	Line Drop Compensation reactance value used during a contingency power flow solution. Value will be expressed in per unit on the system MVA base.
`RegMax`	Real	Maximum desired regulated value for control
`RegMin`	Real	Minimum desired regulated value for control
`RegTargetType`	String	Target Type for the control when going outside of the RegMax and RegMin values. Choices are 1. Middle 2. Max/Min
`XFMVABase`	Real	MVA Base on which the transformer impedances (`Rxfbase`, `Xxfbase`, `Gxfbase`, `Bxfbase`, `Gmagxfbase`, `Bma`

		gxfbase) are given.
XFNomkVbaseFrom	Real	Transformer's Nominal Voltage base for the FROM bus
XFNomkVbaseTo	Real	Transformer's Nominal Voltage base for the TO bus
Rxfbase	Real	Resistance given on the transformer base
Xxfbase	Real	Reactance given on the transformer base
Gxfbase	Real	Shunt Conductance given on the transformer base
Bxfbase	Real	Shunt Susceptance given on the transformer base
Gmagxfbase	Real	Magnetizing Conductance given on the transformer base
Bmagxfbase	Real	Magnetizing Susceptance given on the transformer base
TapFixedFrom	Real	Fixed tap ratio on the FROM side on the transformer base
TapFixedTo	Real	Fixed tap ratio on the TO side on the transformer base
TapMaxxfbase	Real	Maximum tap ratio on the transformer base
TapMinxfbase	Real	Minimum tap ratio on the transformer base
TapStepSizexfbase	Real	Tap ratio step size on the transformer base
Tapxfbase	Real	Present tap ratio on the transformer base
Phase	Real	Phase Shift Angle
ImpCorrTable	Integer	Impedance correction table used. Specify 0 if none used.
LineLength	Real	Length of the branch. This field really doesn't make a lot of sense for a transformer, but is included to be helpful as non-transformer branches all have a LineLength.
Monitor	String	Set to YES to specify that this branch should be monitored. Set to NO to not monitor this branch.
LimitSet	String	Name of the Limit Set to which the branch belongs
LimitMVAA	Real	A Rating of the branch in MVA
LimitMVAB	Real	B Rating of the branch in MVA
LimitMVAC	Real	C Rating of the branch in MVA
LimitMVAD	Real	D Rating of the branch in MVA
LimitMVAE	Real	E Rating of the branch in MVA
LimitMVAF	Real	F Rating of the branch in MVA
LimitMVAG	Real	G Rating of the branch in MVA
LimitMVAH	Real	H Rating of the branch in MVA
LimitMVAI	Real	I Rating of the branch in MVA
LimitMVAJ	Real	J Rating of the branch in MVA
LimitMVAK	Real	K Rating of the branch in MVA
LimitMVAL	Real	L Rating of the branch in MVA
LimitMVAM	Real	M Rating of the branch in MVA
LimitMVAN	Real	N Rating of the branch in MVA
LimitMVAO	Real	O Rating of the branch in MVA
OwnerNum1	Integer	Owner Number 1. May also be listed as blank to indicate that the owner is the same as the owner of

		from bus.
OwnerPerc1	Real	Owner 1 Percent
OwnerNum2	Integer	Owner Number 2
OwnerPerc2	Real	Owner 2
OwnerNum3	Integer	Owner Number 3
OwnerPerc3	Real	Owner 3
OwnerNum4	Integer	Owner Number 4
OwnerPerc4	Real	Owner 4
OwnerNum5	Integer	Owner Number 5
OwnerPerc5	Real	Owner 5
OwnerNum6	Integer	Owner Number 6
OwnerPerc6	Real	Owner 6
OwnerNum7	Integer	Owner Number 7
OwnerPerc7	Real	Owner 7
OwnerNum8	Integer	Owner Number 8
OwnerPerc8	Real	Owner 8
EMSType	String	Record type read from an EMS system
EMSID	String	String ID for branch used in EMS systems
EMSLineID	String	String ID for group container in EMS system
EMSCBTyp	String	String ID for switch type in EMS System
EMSID2From	String	String ID for the from bus side measurement object in EMS System
EMSID2To	String	String ID for the to bus side measurement object in EMS System
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank. *Note: For the windings a three-winding transformer, this value cannot be specified and any entry here will be ignored. The DataMaitainer will always be inherited from the three-winding transformer record.*
DataMaintainerInherit	String	Default value of this field is YES. Set to NO to prevent the object from inheriting its DataMaintainer from another object. [Added in Version 19, December 1, 2016]
AllLabels	String	A comma-delimited list of unique label identifiers for this object. The syntax for this field is described in detail in Section 2.3.4.

A sample file section is shown as follows:

```
Branch (BusNumFrom,BusNumTo,Circuit,BranchDeviceType,Status,StatusNormal,ByPass,
 MeteredEnd,ControlType,AutoControl,RegBusNum,UseLineDrop,Rcomp,Xcomp,RegMax,
 RegMin,RegTargetType,XFMVABase,XFNomkVbaseFrom,XFNomkVbaseTo,Rxfbase,Xxfbase,
 Gxfbase,Bxfbase,Gmagxfbase,Bmagxfbase,TapFixedFrom,TapFixedTo,TapMaxxfbase,
 TapMinxfbase,TapStepSizexfbase,Tapxfbase,Phase,ImpCorrTable,LineLength,
```

```
 Monitor,LimitSet,LimitMVAA,LimitMVAB,LimitMVAC,LimitMVAD,LimitMVAE,LimitMVAF,
 LimitMVAG,LimitMVAH,LimitMVAI,LimitMVAJ,LimitMVAK,LimitMVAL,LimitMVAM,
 LimitMVAN,LimitMVAO,OwnerNum1,OwnerPerc1,OwnerNum2,OwnerPerc2,OwnerNum3,
 OwnerPerc3,OwnerNum4,OwnerPerc4,OwnerNum5,OwnerPerc5,OwnerNum6,OwnerPerc6,
 OwnerNum7,OwnerPerc7,OwnerNum8,OwnerPerc8,EMSType,EMSID,EMSLineID,EMSCBTyp,
 EMSID2From,EMSID2To,DataMaintainerAssign,AllLabels,CustomFloat:0)
{
 10004 10005 "1" "Transformer" "Closed" "Closed" "NO" "From" "LTC" "YES" 10005 "NO " 0.0
0.00010 1.0210 0.9790 "Middle" 100.0 115.0 12.50 0.0     0.6808 0.0 0.0 0.0 0.0 1.0 1.0
1.10 0.90 0.006250 1.03750 0.0 0 1.0 "YES" "Default" 21.8 24.0 21.8 24.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 129 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
"" "" "" "" "" ""
 10006 10408 "1" "Transformer" "Closed" "Closed" "NO" "From" "LTC" "YES" 10408 "NO " 0.0
0.00010 1.0210 0.9790 "Middle" 100.0  46.0 12.50 0.04316 1.2330 0.0 0.0 0.0 0.0 1.0 1.0
1.10 0.90 0.006250 1.04375 0.0 0 1.0 "YES" "Default"  7.0  7.7  7.0  7.7 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 129 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
"" "" "" "" "" ""
 10007 10008 "1" "Transformer" "Closed" "Closed" "NO" "From" "LTC" "YES" 10008 "NO " 0.0
0.00010 1.0210 0.9790 "Middle" 100.0 115.0 13.80 0.0     0.8061 0.0 0.0 0.0 0.0 1.0 1.0
1.10 0.90 0.006250 1.00625 0.0 0 1.0 "YES" "Default" 12.5 13.8 12.5 13.8 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 129 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
"" "" "" "" "" ""
 10007 10535 "1" "Transformer" "Closed" "Closed" "NO" "From" "LTC" "YES" 10535 "NO " 0.0
0.00010 1.0210 0.9790 "Middle" 100.0 115.0 13.80 0.0     0.8061 0.0 0.0 0.0 0.0 1.0 1.0
1.10 0.90 0.006250 1.01250 0.0 0 1.0 "YES" "Default" 12.5 13.8 12.5 13.8 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 129 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
"" "" "" "" "" ""
}
```

4.19 3WXformer

3WXformer fields for a three-winding transformer in the basic power flow case format are as follows.

Field	Type	Description
BusIDPri	Integer	*KEY1* Number of the primary bus. When reading record, see special note in Section 2.4.
BusIDSec	Integer	*KEY2* Number of the secondary bus. When reading record, see special note in Section 2.4.
BusIDTer	Integer	*KEY3* Number of the tertiary bus. When reading record, see special note in Section 2.4.
Circuit	String[2]	*KEY4* Two character ID of the branch
BusIDStar	Integer	Number of the star bus.
BusNameStar	String	Name of the star bus
VpuStar	Real	Voltage in per unit at the star bus
VangleStar	Real	Angle in degrees at the star bus
StatusPri StatusSec StatusTer	String	Status of the primary, secondary and tertiary winding. Either OPEN or CLOSED
MeteredEndPri MeteredEndSec MeteredEndTer	String	Specify either FROM or TO. FROM means the terminal of the primary, secondary, or tertiary of the 3WXformer, while TO means the internal star bus. This indicates the end of the branch which is metered when used as a tie-line between areas, zones, or balancing authorities.
MVABasePriSec	Real	MVA BasePrimary-Secondary (RbasePriSec and

		XbasePriSec are given on this MVA Base). If a zero or negative value is specified for this field, then the value is set equal to the System MVA Base
MVABaseSecTer	Real	MVA Base Secondary-Tertiary (RbaseSecTer and XbaseSecTer are given on this MVA Base). If a zero or negative value is specified, then the value is set equal to MVABasePriSec
MVABaseTerPri	Real	MVA Base Tertiary-Primary (RbaseTerPri and XbaseTerPri are given on this MVA Base). If a zero or negative value is specified, then the value is set equal to MVABasePriSec
RbasePriSec	Real	Per unit resistance Primary-Secondary on MVABasePriSec
XbasePriSec	Real	Per unit reactance Primary-Secondary on MVABasePriSec
RbaseSecTer	Real	Per unit resistance Secondary-Tertiary on MVABaseSecTer
XbaseSecTer	Real	Per unit reactance Secondary-Tertiary on MVABaseSecTer
RbaseTerPri	Real	Per unit resistance Tertiary-Primary on MVABaseTerPri
XbaseTerPri	Real	Per unit reactance Tertiary-Primary on MVABaseTerPri
Gmagbase	Real	Per unit magnetizing conductance (G) on MVABasePriSec
Bmagbase	Real	Per unit magnetizing susceptance (B) on MVABasePriSec
ImpCorrTablePri ImpCorrTableSec ImpCorrTableTer	Integer	Impedance correction table used for respective winding. Specify 0 if none used.
ControlTypePri ControlTypeSec ControlTypeTer	String	Control Type of the respective winding of the transformer. Choices are Fixed, LTC, Mvar, Phase. LTC means that a voltage in controlled by moving the transformer tap. Mvar means that the Mvar flow on the branch is controlled by moving the tap ratio. Phase means that the MW flow on the branch is controlled by changing the phase shift angle.
AutoControlPri AutoControlSec AutoControlTer	String	Set to either YES or NO to indicate whether automatic transformer control is available for the respective winding. If the ControlType of winding is Phase, then the choice OPF is also available to indicate that the phase angle can be an OPF control variable.
RegBusNumPri RegBusNumSec RegBusNumTer	Integer	Regulated Bus Number for respective winding's tap
UseLineDropPri UseLineDropSec UseLineDropTer	String	Set to either YES or NO NO : Use normal voltage control based on the regulated bus and voltage setpoint. YES : Use line drop compensation voltage

		control on the respective winding In order to use line drop compensation on the respective winding, the regulated bus must be one of the terminals of the branch. The line drop is then calculated looking out from that branch into the rest of the system.
RcompPri RcompSec RcompTer	Real	Line Drop Compensation resistance value for respective winding used during a contingency power flow solution. Value will be expressed in per unit on the system MVA base.
XcompPri XcompSec XcompTer	Real	Line Drop Compensation reactance value used during a contingency power flow solution. Value will be expressed in per unit on the system MVA base.
RegMaxPri RegMaxSec RegMaxTer	Real	Maximum desired regulated value for control using the respective winding
RegMinPri RegMinSec RegMinTer	Real	Minimum desired regulated value for control using the respective winding
RegTargetTypePri RegTargetTypeSec RegTargetTypeTer	String	Target Type for the control when going outside of the RegMax and RegMin values. Choices are 1. Middle 2. Max/Min
NomkVPri NomkVSec NomkVTer	Real	Transformer Nominal kV Voltage of the respective winding
TapFixedPri TapFixedSec TapFixedTer	Real	Fixed Tap on transformer voltage kV base for respective winding
TapMaxPri TapMaxSec TapMaxTer	Real	Maximum total tap on transformer voltage kV base for primary
TapMinPri TapMinSec TapMinTer	Real	Minimum total tap on transformer voltage kV base for respective winding
TapStepSizePri TapStepSizeSec TapStepSizeTer	Real	Tap step size on transformer voltage kV base for respective winding
TapPri TapSec TapTer	Real	Total Tap on transformer voltage kV base for respective winding
PhasePri PhaseSec PhaseTer	Real	Phase shift for respective winding
OwnerNum1	Integer	Owner Number 1. May also be listed as blank to indicate that the owner is the same as the owner of from bus.
OwnerPerc1	Real	Owner 1 Percent
OwnerNum2	Integer	Owner Number 2
OwnerPerc2	Real	Owner 2
OwnerNum3	Integer	Owner Number 3
OwnerPerc3	Real	Owner 3

OwnerNum4	Integer	Owner Number 4
OwnerPerc4	Real	Owner 4
OwnerNum5	Integer	Owner Number 5
OwnerPerc5	Real	Owner 5
OwnerNum6	Integer	Owner Number 6
OwnerPerc6	Real	Owner 6
OwnerNum7	Integer	Owner Number 7
OwnerPerc7	Real	Owner 7
OwnerNum8	Integer	Owner Number 8
OwnerPerc8	Real	Owner 8
MonitorPri MonitorSec MonitorTer	String	Set to YES to specify that this respective winding should be monitored. Set to NO to not monitor this respective winding.
LimitSetPri LimitSetSec LimitSetTer	String	Name of the Limit Set to which the respective winding belongs
LimitMVAAPri .. LimitMVAOPri	Real	15 limits A..H for the primary winding
LimitMVAASec .. LimitMVAOSec	Real	15 limits A..H for the secondary winding
LimitMVAATer .. LimitMVAOTer	Real	15 limits A..H for the secondary winding
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.
DataMaintainerInherit	String	Default value of this field is YES. Set to NO to prevent the object from inheriting its DataMaintainer from another object. [Added in Version 19, December 1, 2016]
AllLabels	String	A comma-delimited list of unique label identifiers for this object. The syntax for this field is described in detail in Section 2.3.4.

A sample file section is shown as follows:

```
3WXFormer (BusIDPri,BusIDSec,BusIDTer,BusIDStar,Circuit,StatusPri,StatusSec,StatusTer,
 MeteredEndPri,MeteredEndSec,MeteredEndTer,MVABasePriSec,MVABaseSecTer,
 MVABaseTerPri,RbasePriSec,XbasePriSec,RbaseSecTer,XbaseSecTer,RbaseTerPri,
 XbaseTerPri,Gmagbase,Bmagbase,ImpCorrTablePri,ControlTypePri,AutoControlPri,
 RegBusNumPri,UseLineDropPri,RcompPri,XcompPri,RegMaxPri,RegMinPri,
 RegTargetTypePri,NomkVPri,TapFixedPri,TapMaxPri,TapMinPri,TapStepSizePri,
 TapPri,PhasePri,ImpCorrTableSec,ControlTypeSec,AutoControlSec,RegBusNumSec,
 UseLineDropSec,RcompSec,XcompSec,RegMaxSec,RegMinSec,RegTargetTypeSec,
 NomkVSec,TapFixedSec,TapMaxSec,TapMinSec,TapStepSizeSec,TapSec,PhaseSec,
 ImpCorrTableTer,ControlTypeTer,AutoControlTer,RegBusNumTer,UseLineDropTer,
 RcompTer,XcompTer,RegMaxTer,RegMinTer,RegTargetTypeTer,NomkVTer,TapFixedTer,
 TapMaxTer,TapMinTer,TapStepSizeTer,TapTer,PhaseTer,OwnerNum1,OwnerPerc1,
 OwnerNum2,OwnerPerc2,OwnerNum3,OwnerPerc3,OwnerNum4,OwnerPerc4,OwnerNum5,
 OwnerPerc5,OwnerNum6,OwnerPerc6,OwnerNum7,OwnerPerc7,OwnerNum8,OwnerPerc8,
 MonitorPri,LimitSetPri,MonitorSec,LimitSetSec,MonitorTer,LimitSetTer,
 LimitMVAAPri,LimitMVABPri,LimitMVACPri,LimitMVADPri,LimitMVAEPri,LimitMVAFPri,
 LimitMVAGPri,LimitMVAHPri,LimitMVAIPri,LimitMVAJPri,LimitMVAKPri,LimitMVALPri,
 LimitMVAMPri,LimitMVANPri,LimitMVAOPri,LimitMVAASec,LimitMVABSec,LimitMVACSec,
 LimitMVADSec,LimitMVAESec,LimitMVAFSec,LimitMVAGSec,LimitMVAHSec,LimitMVAISec,
```

```
 LimitMVAJSec,LimitMVAKSec,LimitMVALSec,LimitMVAMSec,LimitMVANSec,LimitMVAOSec,
 LimitMVAATer,LimitMVABTer,LimitMVACTer,LimitMVADTer,LimitMVAETer,LimitMVAFTer,
 LimitMVAGTer,LimitMVAHTer,LimitMVAITer,LimitMVAJTer,LimitMVAKTer,LimitMVALTer,
 LimitMVAMTer,LimitMVANTer,LimitMVAOTer,DataMaintainerAssign,AllLabels)
{
" 11008" " 11010" " 11238" " 11237" " 1" "Closed" "Closed" "Open"   "To"   "From" "From"
100.0 100.0 100.0 0.00080 0.063992 0.0      0.16210 0.0      0.0910 0.000690 -0.000690 0
"Fixed" "NO" 11010 "NO " 0.0 0.00010 0.0 0.0 "Middle" 115.0 1.0     0.0  0.0 0.0 1.00
0.0 0 "Fixed" "NO" 0 "NO " 0.0 0.00010 1.50 0.510 "Middle" 345.0 1.0 1.50 0.510 0.006250
1.0 0.0 0 "Fixed" "NO" 0 "NO " 0.0 0.00010 1.50 0.510 "Middle" 13.20 1.0 1.50 0.510
0.006250 1.0 -30.0 125 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" "" "YES" "Default"
"YES" "Default" "YES" "Default" 290.0 333.0 290.0 333.0 290.0 333.0 290.0 333.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 60.0 69.0
69.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 "" ""
" 11060" " 11061" " 11201" " 11202" " 1" "Closed" "Closed" "Closed" "To"   "From" "From"
100.0 100.0 100.0 0.00113 0.051490 0.005870 0.24190 0.005660 0.1781 0.000630 -0.000230 0
"Fixed" "NO"     0 "NO " 0.0 0.00010 0.0 0.0 "Middle" 115.5 1.0     0.0  0.0 0.0 0.99995
0.0 0 "Fixed" "NO" 0 "NO " 0.0 0.00010 1.50 0.510 "Middle" 345.0 1.0 1.50 0.510 0.006250
1.0 0.0 0 "Fixed" "NO" 0 "NO " 0.0 0.00010 1.50 0.510 "Middle" 13.80 1.0 1.50 0.510
0.006250 1.0 -30.0 125 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" "" "YES" "Default"
"YES" "Default" "YES" "Default" 224.0 258.0 224.0 258.0 224.0 258.0 224.0 258.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0   0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 "" ""
" 11111" " 11110" " 11203" " 11204" " 1" "Closed" "Closed" "Open"   "From" "From" "From"
100.0 100.0 100.0 0.00101 0.051440 0.001940 0.05630 0.002190 0.1185 0.000810 -0.001030 0
"Fixed" "NO"     0 "NO " 0.0 0.00010 0.0 0.0 "Middle" 345.0 1.0250  0.0  0.0 0.0 1.00
0.0 0 "Fixed" "NO" 0 "NO " 0.0 0.00010 1.50 0.510 "Middle" 115.0 1.0 1.50 0.510 0.006250
1.0 0.0 0 "Fixed" "NO" 0 "NO " 0.0 0.00010 1.50 0.510 "Middle" 13.80 1.0 1.50 0.510
0.006250 1.0 -30.0 125 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" "" "YES" "Default"
"YES" "Default" "YES" "Default" 230.0 265.0 230.0 265.0 230.0 265.0 230.0 265.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0   0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 "" ""
}
```

4.20 MultiSectionLine

MultiSectionLine fields in the basic power flow case format are as follows. Note that the series of branch objects that make up a MultiSectionLine must be directly in series with each other. Also the intermediate buses in this aggregation can only have single terminal objects connected to them such as generators, loads, and shunt objects.

Field	Type	Description
BusNumFrom	Integer	*KEY1* Number of the from bus
BusNumTo	Integer	*KEY2* Number of the To bus
Circuit	String[2]	*KEY3* Two character ID of the MultiSectionLine. This identifier must be unique regardless of whether the branch is a transformer or a non-transformer.
AllowMixedStatus	String	Set to YES to allow the branches inside the MultiSectionLine to have different statuses. Normally this is NO indicating that when you change the status of one of the Branch in the MultiSectionLine then all other branches change to match this status.
BusInt:0 .. BusInt:20	String	Identifiers for the Intermediate Buses of the MultiSectionLine

A sample file section is shown as follows:

```
MultiSectionLine (BusNumFrom,BusNumTo,Circuit,AllowMixedStatus,
                  BusInt:0,BusInt:1,BusInt:2,BusInt:3,BusInt:4,
```

```
                  BusInt:5,BusInt:6,BusInt:7,BusInt:8,BusInt:9,DataMaintainerAssign)
{
 10048  10014 "1 " "NO " "10051" "" "" "" "" "" "" "" "" "" ""
 10099  10098 "1 " "NO " "10109" "10110" "10146" "10148" "" "" "" "" "" "" ""
 10107  10057 "1 " "NO " "10175" "" "" "" "" "" "" "" "" "" ""
10159  10073 "1 " "NO " "10189" "10190" "" "" "" "" "" "" "" "" ""
 40687  30005 "2 " "NO " "40696" "40697" "40698" "" "" "" "" "" "" "" ""
 40489  40687 "2 " "NO " "40700" "40702" "40704" "40706" "40708" "40710" "" "" "" "" ""
 40381  40693 "1 " "NO " "40711" "40712" "" "" "" "" "" "" "" "" ""
 40051  40699 "1 " "NO " "40714" "" "" "" "" "" "" "" "" "" ""
 45035  30020 "1 " "NO " "45036" "45038" "45039" "" "" "" "" "" "" "" ""
 45035  40834 "1 " "NO " "45040" "45041" "45042" "45043" "45044" "45046" "" "" "" "" ""
 45047  40635 "1 " "NO " "45048" "" "" "" "" "" "" "" "" "" ""
}
```

4.21 Shunt (ShuntMode <> Bus Shunt)

Shunt fields in the basic power flow case format are as follows.

Field	Type	Description
`BusNum`	Integer	*KEY* Number of the Bus. When reading record, see special note in Section 2.4.
`ID`	String[2]	*KEY2* 2 character identification field. Used to identify multiple shunts at a single bus. Note these identifiers must be unique across all shunt objects regardless of ShuntMode.
`Status`	String	Status of the shunt. Set to either OPEN or CLOSED
`StatusBranch`	String	Object string referencing a branch in the system using the format of Section 2.3. The shunt will be considered out of service whenever the linked branch is out of service. This is a method of defining a line shunt if your model does not presently have breakers explicitly defined. This option will be available starting in Version 20 of Simulator.
`ShuntMode`	String	Specify the type of control mode for this shunt. The choices are `Fixed`, `Discrete`, `Continuous`, `Bus Shunt` or `SVC`. Note that in this table it is expected that none of the entries will be `Bus Shunt` as those are stored in another table.
`AutoControl`	String	Set to either YES, NO, or FORCE to indicate whether automatic control is available for this shunt. NO : means it is not controlled YES : means it is available for control if the Area field `AutoControlShunt` = YES and the global option to enable shunts to move is enabled FORCE : means it is available for control and it ignores the Area field `AutoControlShunt` and the global option regarding shunt control. The FORCE option will be available starting in Version 20 of Simulator.

VoltageControlGroup	String	Name of the voltage control group to which the shunt belongs.
MWNom	Real	Nominal MW value of the shunt at 1.0 per unit voltage. The shunt is modeled as an impedance, therefore the actual MW will then be this value multiplied by the square of the per unit voltage.
MvarNom	Real	Nominal Mvar value of the shunt at 1.0 per unit voltage. The shunt is modeled as an impedance, therefore the actual Mvar will then be this value multiplied by the square of the per unit voltage.
RegBusNum	Integer	Bus number of the regulated bus
RegHigh	Real	Shunt will try to keep regulated value below this value
RegLow	Real	Shunt will try to keep regulated value above this value
RegTarget	Real	When the regulated value goes outside of the Low-High desired range, the control logic will attempt to bring it back to this target value
RegTargetHighUse	String	Set to YES or NO. Default value is NO which means that the RegTarget value is used for both low and high excursions. If set to YES, then low excursions will use RegTarget, while high excursions will use RegTargetHigh.
RegTargetHigh	Real	When the regulated value goes above of the RegHigh value, the control logic will attempt to bring it back to this target value
RegFactor	Real	Amount of Mvar support that this switched shunt will provide if the bus that it is regulating is being regulated by more than one switched shunt.
RegulationType	String	Choices are either `Volt`, `Gen Mvar`, `Wind Mvar`, or `Custom Control`
CustomControlModelExpressionName	String	When using Custom Control RegulationType, this is the name of the Model Expression which described the desired Mvar output of the shunt.
InnerPowerFlow	String	Set to YES to allow the switched shunt to operate during the solution of the power flow equations. Default value is NO meaning that the shunt will be moved during the voltage control loop.
FullCapacitySwitch	String	Default Value is NO. Set to YES to signify that this this shunt may operate only at highest Nominal Mvar possible or at lowest Nominal Mvar possible. This is only done when (ShuntMode = `Discrete`) or when both (ShuntMode = `SVC`) AND (SVCType = `SVSMO2`)
ContinuousUse	String	Set to YES to use the ContinuousMvarNomMax and ContinuousMvarNomMin when the ShuntMode is

		set to Discrete or Continuous
ContinousMvarNomMax	Real	Minimum Nominal Mvar of the continuous element. This values is used with Discrete or Continuous ShuntMode when ContinuousUse = YES. It is also used when ShuntMode = SVC if the SVCType = SVSM01 or SVSM03
ContinousMvarNomMin	Real	Minimum Nominal Mvar of the continuous element. This values is used with Discrete or Continuous ShuntMode when ContinuousUse = YES. It is also used when ShuntMode = SVC if the SVCType = SVSM01 or SVSM03
BlockNumberStep1	Integer	Number of equal nominal Mvar steps for block 1
BlockMvarPerStep1	Real	Nominal Mvar per step for block 1
BlockNumberStep2	Integer	Number of equal nominal Mvar steps for block 2
BlockMvarPerStep2	Real	Nominal Mvar per step for block 2
BlockNumberStep3	Integer	Number of equal nominal Mvar steps for block 3
BlockMvarPerStep3	Real	Nominal Mvar per step for block 3
BlockNumberStep4	Integer	Number of equal nominal Mvar steps for block 4
BlockMvarPerStep4	Real	Nominal Mvar per step for block 4
BlockNumberStep5	Integer	Number of equal nominal Mvar steps for block 5
BlockMvarPerStep5	Real	Nominal Mvar per step for block 5
BlockNumberStep6	Integer	Number of equal nominal Mvar steps for block 6
BlockMvarPerStep6	Real	Nominal Mvar per step for block 6
BlockNumberStep7	Integer	Number of equal nominal Mvar steps for block 7
BlockMvarPerStep7	Real	Nominal Mvar per step for block 7
BlockNumberStep8	Integer	Number of equal nominal Mvar steps for block 8
BlockMvarPerStep8	Real	Nominal Mvar per step for block 8
BlockNumberStep9	Integer	Number of equal nominal Mvar steps for block 9
BlockMvarPerStep9	Real	Nominal Mvar per step for block 9
BlockNumberStep10	Integer	Number of equal nominal Mvar steps for block 10
BlockMvarPerStep10	Real	Nominal Mvar per step for block 10
SVCType	String	When ShuntMode = SVC, then this specifies the type of the SVC. Choices are None, SVSM01, SVSM02, and SVSM03
SVCXcomp	Real	SVC control compensating reactance. This works very similarly to line drop compensation for both generator and transformer control.
SVCMvarNomMaxSH	Real	Maximum of Nominal Mvar range in which remote shunts are not switched. Value is expressed in nominal Mvar which represent what the Mvar would be at 1.0 per unit voltage.
SVCMvarNomMinSH	Real	Minimum of Nominal Mvar range in which remote shunts are not switched. Value is expressed in

		nominal Mvar which represent what the Mvar would be at 1.0 per unit voltage.
SVCstsb	String	YES/NO status. For SVCType = SVSM01 and SVSM02, set this to YES to enable the Slow B Control. For SVCType = SVSM03, set this to YES to enable the Ireset or deadband control
SVCMvarNomMaxSB	Real	Maximum of Nominal Mvar range for SVCType = SVSM01 and SVSM02 for Slow B Control. Not used with SVCType = SVSM03
SVCMvarNomMinSB	Real	Minimum of Nominal Mvar range for SVCType = SVSM01 and SVSM02 for Slow B Control. Not used with SVCType = SVSM03
SVCVrefmax	Real	Voltage Range Maximum for the Slow B Control used with SVCType = SVSM01 and SVSM02.
SVCVrefmin	Real	Voltage Range Minimum for the Slow B Control used with SVCType = SVSM01 and SVSM02.
SVCdvdb	Real	Voltage Sensitivity for a change in Injection. Units are Per unit Voltage / Per unit B
AreaNumber	Integer	It is possible for the terminal bus to belong to a different area than the device belongs. This is the Area number of the Shunt. When reading this field, if an area does not already exist with this number, then a new area will automatically be created.
ZoneNumber	Integer	It is possible for the terminal bus to belong to a different zone than the device belongs. This is the Zone number of the Shunt. When reading this field, if a zone does not already exist with this number, then a new zone will automatically be created.
BANumber	Integer	It is possible for the terminal bus to belong to a different balancing authority than the device belongs. This is the Balancing Authority number of the Shunt. When reading this field, if a balancing authority does not already exist with this number, then a new balancing authority will automatically be created.
OwnerNum1	Integer	Owner Number 1
OwnerPerc1	Real	Owner 1
OwnerNum2	Integer	Owner Number 2
OwnerPerc2	Real	Owner 2
OwnerNum3	Integer	Owner Number 3
OwnerPerc3	Real	Owner 3
OwnerNum4	Integer	Owner Number 4
OwnerPerc4	Real	Owner 4
EMSType	String	Record type read from an EMS system
EMSID	String	String ID for shunt used in EMS systems

DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.
DataMaintainerInherit	String	Default value of this field is YES. Set to NO to prevent the object from inheriting its DataMaintainer from another object. [Added in Version 19, December 1, 2016]
AllLabels	String	A comma-delimited list of unique label identifiers for this object. The syntax for this field is described in detail in Section 2.3.4.

A sample file section is shown as follows:

```
Shunt (BusNum,ID,Status,ShuntMode,VoltageControlGroup,MWNom,MvarNom,RegBusNum,RegHigh,
 RegLow,RegTarget,RegTargetHighUse,RegTargetHigh,RegFactor,RegulationType,
 CustomControlModelExpressionName,InnerPowerFlow,FullCapacitySwitch,
 ContinuousUse,ContinuousMvarNomMax,ContinuousMvarNomMin,BlockNumberStep1,
 BlockMvarPerStep1,BlockNumberStep2,BlockMvarPerStep2,BlockNumberStep3,
 BlockMvarPerStep3,BlockNumberStep4,BlockMvarPerStep4,BlockNumberStep5,
 BlockMvarPerStep5,BlockNumberStep6,BlockMvarPerStep6,BlockNumberStep7,
 BlockMvarPerStep7,BlockNumberStep8,BlockMvarPerStep8,BlockNumberStep9,
 BlockMvarPerStep9,BlockNumberStep10,BlockMvarPerStep10,SVCType,SVCXcomp,
 SVCMvarNomMaxSH,SVCMvarNomMinSH,SVCstsb,SVCMvarNomMaxSB,SVCMvarNomMinSB,
 SVCVrefmax,SVCVrefmin,SVCdvdb,AreaNumber,ZoneNumber,BANumber,OwnerNum1,
 OwnerPerc1,OwnerNum2,OwnerPerc2,OwnerNum3,OwnerPerc3,OwnerNum4,OwnerPerc4,
 EMSType,EMSID,DataMaintainerAssign,AllLabels)
{
 10931 "1 " "Closed" "Fixed"    "" 0.0  7.50    10931 1.0220 1.0220 1.0220 "NO " 1.0220
1.0 "Volt" "None" "NO " "NO " "YES"  0.0   0.0 5 7.50 "" "" "" "" "" "" "" "" "" "" "" ""
"" "" "" "" "" "" "None" 0.0 25.0 -50.0 "YES" 25.0 -50.0 0.0 0.0 0.0550 10 10 10 2407
100.0 "" "" "" "" "" "" "" "" "" ""
 11010 "v1" "Closed" "SVC"      "" 0.0  5.07818 11010 1.0320 1.0320 1.0320 "NO " 1.0320
1.0 "Volt" "None" "NO " "NO " "NO " 25.0 -50.0 1 5.07818 "" "" "" "" "" "" "" "" "" "" ""
"" "" "" "" "" "" "" "SVSMO1" 0.0 25.0 -50.0 "YES" 25.0 -50.0 1.050 1.020 0.0550 11 110
11 125 100.0 "" "" "" "" "" "" "" "" "" ""
 12011 "v " "Closed" "Discrete" "" 0.0 15.0     12011 1.0500 1.0200 1.0350 "NO " 1.0350
1.0 "Volt" "None" "NO " "NO " "NO "  0.0   0.0 1 15.0 1 15.0 "" "" "" "" "" "" "" "" ""
"" "" "" "" "" "" "" "None" 0.0 25.0 -50.0 "YES" 25.0 -50.0 0.0 0.0 0.0550 10 121 10 127
100.0 "" "" "" "" "" "" "" "" "" ""
}
```

4.22 Shunt (ShuntMode = Bus Shunt)

Shunt fields for shunts which are a simple Bus Shunt in the basic power flow case format are as follows.

Field	Type	Description
BusNum	Integer	*KEY1* Number of the Bus. When reading record, see special note in Section 2.4.
ID	String	*KEY2* 2 character identification field. Used to identify multiple shunts at a single bus. Note these identifiers must be unique across all shunt objects regardless of ShuntMode.
Status	String	Status of the shunt. Set to either OPEN or CLOSED
StatusBranch	String	Object string referencing a branch in the system using the format of Section 2.3. The shunt will be considered out of service whenever the linked branch is out of service. This

		is a method of defining a line shunt if your model does not presently have breakers explicitly defined.
`ShuntMode`	String	Will always be `BusShunt`
`AutoControl`	String	Set to either YES, NO, or FORCE to indicate whether automatic control is available for this shunt. NO : means it is not controlled YES : means it is available for control if the Area field `AutoControlShunt` = YES and the global option to enable shunts to move is enabled FORCE : means it is available for control and it ignores the Area field `AutoControlShunt` and the global option regarding shunt control.
`MWNom`	Real	Nominal MW value of the shunt at 1.0 per unit voltage. The shunt is modeled as an impedance, therefore the actual MW will then be this value multiplied by the square of the per unit voltage.
`MvarNom`	Real	Nominal Mvar value of the shunt at 1.0 per unit voltage. The shunt is modeled as an impedance, therefore the actual Mvar will then be this value multiplied by the square of the per unit voltage.
`SVCstsv`	String	YES or NO. Specifies whether the shunt can be controlled by the another shunt which is on the ShuntMode = SVC
`SVCControlling`	String	Object String for the shunt which is controlling this Shunt. Format is string format described in Section 2.2.
`AreaNumber`	Integer	It is possible for the terminal bus to belong to a different area than the device belongs. This is the Area number of the Shunt. When reading this field, if an area does not already exist with this number, then a new area will automatically be created.
`ZoneNumber`	Integer	It is possible for the terminal bus to belong to a different zone than the device belongs. This is the Zone number of the Shunt. When reading this field, if a zone does not already exist with this number, then a new zone will automatically be created.
`BANumber`	Integer	It is possible for the terminal bus to belong to a different balancing authority than the device belongs. This is the Balancing Authority number of the Shunt. When reading this field, if a balancing authority does not already exist with this number, then a new balancing authority will automatically be created.
`OwnerNum1`	Integer	Owner Number 1
`OwnerPerc1`	Real	Owner 1
`OwnerNum2`	Integer	Owner Number 2
`OwnerPerc2`	Real	Owner 2
`OwnerNum3`	Integer	Owner Number 3

OwnerPerc3	Real	Owner 3
OwnerNum4	Integer	Owner Number 4
OwnerPerc4	Real	Owner 4
EMSType	String	Record type read from an EMS system
EMSID	String	String ID for shunt used in EMS systems
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.
DataMaintainerInherit	String	Default value of this field is YES. Set to NO to prevent the object from inheriting its DataMaintainer from another object. [Added in Version 19, December 1, 2016]
AllLabels	String	A comma-delimited list of unique label identifiers for this object. The syntax for this field is described in detail in Section 2.3.4.

A sample file section is shown as follows:

```
Shunt (BusNum,ID,Status,ShuntMode,MWNom,MvarNom,SVCstsv,SVCControlling,AreaNumber,
 ZoneNumber,BANumber,OwnerNum1,OwnerPerc1,OwnerNum2,OwnerPerc2,OwnerNum3,
 OwnerPerc3,OwnerNum4,OwnerPerc4,EMSType,EMSID,DataMaintainerAssign,AllLabels)
{
 10010 "b1" "Open"   "Bus Shunt" 0.0  27.0 "NO " "" 10 107 10 130 100.0 "" "" "" "" "" ""
"" "" "" ""
 10010 "b2" "Open"   "Bus Shunt" 0.0  27.0 "NO " "" 10 107 10 130 100.0 "" "" "" "" "" ""
"" "" "" ""
 10116 "b1" "Closed" "Bus Shunt" 0.0 -65.0 "NO " "" 10  10 10 130 100.0 "" "" "" "" "" ""
"" "" "" ""
}
```

4.23 LineShunt

We discourage the use of this separate LineShunt object and instead a Shunt object should be created with an appropriate StatusBranch field assigned that references the branch whose Status being open means that shunt is also out of service. However, to support existing shunt technologies in the short run, the LineShunt object will remain. LineShunt fields in the basic power flow case format are as follows.

Field	Type	Description
BusNumFrom	Integer	*KEY1* Number of the FROM Bus. When reading record, see special note in Section 2.4.
BusNumTo	Integer	*KEY2* Number of the TO Bus. When reading record, see special note in Section 2.4.
BusNumLoc	Integer	*KEY3* Number of the Bus at which device is connected. When reading record, see special note in Section 2.4.
Circuit	String[2]	*KEY4* Two character ID of the branch to which the shunt is associated.
ID	String[2]	*KEY5* Two character ID of the shunt itself. This must be unique for all shunts on the same BusNomLoc of a particular branch.

AllLabels	String	A comma-delimited list of unique label identifiers for this object. The syntax for this field is described in detail in Section 2.3.4.
Status	String	Status of the shunt. Set to either OPEN or CLOSED
MWNom	Real	Nominal MW value of the shunt at 1.0 per unit voltage. The shunt is modeled as an impedance, therefore the actual MW will then be this value multiplied by the square of the per unit voltage.
MvarNom	Real	Nominal Mvar value of the shunt at 1.0 per unit voltage. The shunt is modeled as an impedance, therefore the actual Mvar will then be this value multiplied by the square of the per unit voltage.
OwnerNum1	Integer	Owner Number 1
OwnerPerc1	Real	Owner 1
OwnerNum2	Integer	Owner Number 2
OwnerPerc2	Real	Owner 2
OwnerNum3	Integer	Owner Number 3
OwnerPerc3	Real	Owner 3
OwnerNum4	Integer	Owner Number 4
OwnerPerc4	Real	Owner 4
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.
DataMaintainerInherit	String	Default value of this field is YES. Set to NO to prevent the object from inheriting its DataMaintainer from another object. [Added in Version 19, December 1, 2016]

A sample file section is shown as follows:

```
LineShunt (BusNumFrom,BusNumTo,Circuit,ID,BusNumLoc,Status,MWNom,MvarNom,OwnerNum1,
 OwnerPerc1,OwnerNum2,OwnerPerc2,OwnerNum3,OwnerPerc3,OwnerNum4,OwnerPerc4,
 DataMaintainerAssign)
{
 10374 10025 "1" "f " 10025 "Closed" 0.0 -65.0 130 100.0 "" "" "" "" "" "" ""
10842 10232 "1" "f1" 10232 "Closed" 0.0 -65.4 130 100.0 "" "" "" "" "" "" ""
}
```

4.24 DFACTSCorrection

XFCorrection fields in the basic power flow case format are as follows.

Field	Type	Description
Number	Integer	*KEY1* Number
Name	String	Name
Amps:0 Amps:1 Amps:2 ... Amps:49	Real	Amps for factor lookup

XC:0 XC:1 XC:2 ... XC:49	Real	XCModule (per unit XC per phase) as a function of Amps
XL:0 XL:1 XL:2 ... XC:49	Real	XLModule (per unit XL per phase) as a function of Amps
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.

A sample file section is shown as follows:

```
DFACTSCorrection (Number,Name,Amps:0,XC:0,XL:0,Amps:1,XC:1,XL:1,
   Amps:2,XC:2,XL:2,Amps:3,XC:3,XL:3
{
}
```

4.25 DFACTS

DFACTS fields in the basic power flow case format are as follows.

Field	Type	Description
BusNumFrom	Integer	*KEY1* Number of the From bus. When reading record, see special note in Section 2.4.
BusNumTo	Integer	*KEY2* Number of the To bus. When reading record, see special note in Section 2.4.
Circuit	String[2]	*KEY3*
ModulesAvail	Integer	
XLModule	Real	
XCModule	Real	
CorrTable	Integer	
ModulesActive	Integer	
Mode	String	
I0Amp	Real	
IlimAmp	Real	
AutoModulesAvail	String	
AutoModulesAvailPerc	Real	
AutoI0	String	
AutoI0Perc	Real	
AutoIlim	String	
AutoIlimPerc	Real	
RegMin	Real	
RegMax	Real	
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit

		their DataMaintainer this will still be blank.
DataMaintainerInherit	String	Default value of this field is YES. Set to NO to prevent the object from inheriting its DataMaintainer from another object. [Added in Version 19, December 1, 2016]

A sample file section is shown as follows:

```
DFACTS (BusNumFrom,BusNumTo,Circuit,ModulesAvail,XLModule,XCModule,CorrTable,
 ModulesActive,Mode,I0pu,Ilimpu,AutoModulesAvail,AutoModulesAvailPerc,AutoI0,
 AutoI0Perc,AutoIlim,AutoIlimPerc,RegMax,RegMin,DataMaintainerAssign)
{
 10026 10 "1" 11 0.000837 0.0 "" 0 "Limit" 82.3473 91.4976 "YES" 30.0 "YES" 90.0 "YES"
100.0 0.0 0.0 ""
}
```

4.26 DCTransmissionLine

DCTransmissionLine fields in the basic power flow case format are as follows.

Field	Type	Description
BusNumRect	Integer	*KEY1* AC Number at Rectifier Bus. When reading record, see special note in Section 2.4.
BusNumInv	Integer	*KEY2* AC Number at Inverter Bus. When reading record, see special note in Section 2.4.
Circuit	Integer	*KEY3* Num ID. Used to identify between multiple DC lines between the same two buses
Name	String	Name of the DC Transmission Line.
Mode	String	Control Mode of DC Line. Choices are • Current (Amps) • Power (MW) • Blocked
R	Real	Resistance (Ohms) of DC line
SetpointMag	Real	Setpoint magnitude of the DC Line. Note that units are determined by the control mode and are either MW or Amps
SetpointEnd	String	Set to either Rectifier or Inverter. Specifies whether the setpoint MW or Amp is meant to be at the rectifier or inverter
SetpointVolt	Real	This is the scheduled DC voltage in kV at which the DC line is to operate
ModeSwitchVoltage	Real	DC mode switch voltage. At a DC voltage below this, the DC line should switch to current mode if presently in power mode. This is not presently handled for two-terminal DC lines, but is for multi-terminal DC lines
RComp	Real	Compounding resistance is used to specify the point along the DC line at which the voltage is controlled. A value of zero means voltage control is at the inverter. A value equal to one-half the resistance of the line means voltage control is half way along the DC line.

Margin	Real	When the DC voltage schedule can not be achieved, the power or current order will not be reduced by more than this amount multiplied by the setpoint . Beyond this point, the voltage schedule will be reduced instead
MeteredEnd	String	Metered end of the DC Line. This is important for area interchange control because all losses are assigned to the area of the bus at the non-metered end.
NumBridgesRect	Integer	Number of Bridges at Rectifier Bus
Alpha	Real	Alpha firing angle at the rectifier
AlphaMax	Real	Alpha maximum firing angle at the rectifier
AlphaMin	Real	Alpha minimum firing angle at the rectifier
RCommRect	Real	Commutating transformer resistance at rectifier bus
XCommRect	Real	Commutating transformer reactance at rectifier bus
VacBaseRect	Real	Primary AC Base Voltage at Rectifier Bus
XFRatioRect	Real	Rectifier Tap Ratio
TapRect	Real	Rectifier Tap Value
TapMaxRect	Real	Rectifier Tap Max
TapMinRect	Real	Rectifier Tap Min
TapStepRect	Real	Tap Step Size at Rectifier Bus
VdiodeRect	Real	Diode voltage drop in kV across each rectifier bridge
NumBridgesInv	Integer	Number of Bridges at Inverter Bus
Gamma	Real	Gamma firing angle at the inverter
GammaMax	Real	Gamma maximum firing angle at the inverter
GammaMin	Real	Gamma minimum firing angle at the inverter
RCommInv	Real	Commutating transformer resistance at inverter bus
XCommInv	Real	Commutating transformer reactance at inverter bus
VacBaseInv	Real	Primary AC Base Voltage at Inverter Bus
XFRatioInv	Real	Inverter Tap Ratio
TapInv	Real	Inverter Tap Value
TapMaxInv	Real	Inverter Tap Max
TapMinInv	Real	Inverter Tap Min
TapStepInv	Real	Tap Step Size at Inverter Bus
VdiodeInv	Real	Diode voltage drop in kV across each inverter bridge
OwnerNum1	Integer	Owner Number 1
OwnerPerc1	Real	Owner 1
OwnerNum2	Integer	Owner Number 2
OwnerPerc2	Real	Owner 2
OwnerNum3	Integer	Owner Number 3
OwnerPerc3	Real	Owner 3
OwnerNum4	Integer	Owner Number 4
OwnerPerc4	Real	Owner 4

OwnerNum5	Integer	Owner Number 5
OwnerPerc5	Real	Owner 5
OwnerNum6	Integer	Owner Number 6
OwnerPerc6	Real	Owner 6
OwnerNum7	Integer	Owner Number 7
OwnerPerc7	Real	Owner 7
OwnerNum8	Integer	Owner Number 8
OwnerPerc8	Real	Owner 8
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.
DataMaintainerInherit	String	Default value of this field is YES. Set to NO to prevent the object from inheriting its DataMaintainer from another object. [Added in Version 19, December 1, 2016]

A sample file section is shown as follows:

```
DCTransmissionLine (BusNumRect,BusNumInv,Circuit,Name,Mode,R,
 SetpointMag,SetpointEnd,SetpointVolt,
 ModeSwitchVoltage,RComp,Margin,MeteredEnd,NumBridgesRect,Alpha,AlphaMax,
 AlphaMin,RCommRect,XCommRect,VacBaseRect,XFRatioRect,TapRect,TapMaxRect,
 TapMinRect,TapStepRect,VdiodeRect,NumBridgesInv,Gamma,GammaMax,GammaMin,
 RCommInv,XCommInv,VacBaseInv,XFRatioInv,TapInv,TapMaxInv,TapMinInv,TapStepInv,
 VdiodeInv,OwnerNum1,OwnerPerc1,OwnerNum2,OwnerPerc2,OwnerNum3,OwnerPerc3,
 OwnerNum4,OwnerPerc4,OwnerNum5,OwnerPerc5,OwnerNum6,OwnerPerc6,OwnerNum7,
 OwnerPerc7,OwnerNum8,OwnerPerc8,DataMaintainerAssign)
{
 55615 55618 1 "1" "Power (MW)" 7.3650  200.0 "Rectifier" 500.0 0.0 0.0 0.10 "I" 1
14.5675 17.30 12.70 0.0 21.320 500.0 0.8320 1.11250 1.20 0.9750 0.01250 0.0 1 19.4087
20.50 18.0 0.0 21.320 240.0 1.73250 1.11610 1.20640 0.89680 0.01290 0.0 76 100.0 "" "" ""
"" "" "" "" "" "" "" "" "" "" "" ""

 54623 54485 1 "2" "Power (MW)" 5.1250  400.0 "Rectifier" 500.0 0.0 0.0 0.10 "I" 1
14.5758 17.30 12.70 0.0 21.320 500.0 0.8320 1.10 1.20 0.9750 0.01250 0.0 1 18.0598 20.50
17.90 0.0 21.320 240.0 1.73250 1.09030 1.20640 0.89680 0.01290 0.0 76 100.0 "" "" "" ""
"" "" "" "" "" "" "" "" "" "" "" 

 41313 26097 1 "3" "Power (MW)" 20.0   1000.0 "Rectifier" 500.0 0.0 20.0 0.0 "R" 4
18.3315 24.0  17.0  0.0 7.5130 230.0 0.4630 1.01090 1.1610 1.0010 0.010 0.0 2 18.7570
24.0 17.0 0.0 8.7160 230.0 0.9230 1.06050 1.2940 0.9060 0.0130 0.0 14 100.0 "" "" "" ""
"" "" "" "" "" "" "" "" "" "" "" 

 41314 26099 1 "4" "Power (MW)" 20.0   1000.0 "Rectifier" 500.0 0.0 20.0 0.0 "R" 4
18.1008 24.0  17.0  0.0 7.5130 230.0 0.4630 1.01090 1.1610 1.0010 0.010 0.0 2 17.2696
24.0 17.0 0.0 8.7160 230.0 0.9230 1.1040 1.3370 0.950 0.0130 0.0 14 100.0 "" "" "" "" ""
"" "" "" "" "" "" "" "" "" "" 
}
```

4.27 VSCDCLine

VSCDCLine fields in the basic power flow case format are as follows.

Field	Type	Description
Name	String	*KEY1*
Status	String	Either Open or Closed

`MWSetSide`	String	Set to either *AC Side* or *DC Side*. Specifies whether the DC MW Setpoint refers to the MW of the DC system or the AC system. There is a difference when there are converter losses modeled.
`R`	Real	Resistance (Ohms) of DC line
`BusNumFrom` `BusNumTo`	Integer	Number of the from and to bus When reading record, see special note in Section 2.4.
`DCModeFrom` `DCModeTo`	String	Mode of the DC control on the respective side . Choices are either `Out-Of-Service`, `Voltage` or `Power`
`ACModeFrom` `ACModeTo`	String	Mode of the AC control on the respective side. Choices are either `Voltage` or `Power Factor`
`DCSetFrom` `DCSetTo`	Real	DC control set point on the respective. For DC Mode of Voltage, the units are per unit. For DC Mode of Power the unit are MW.
`ACSetFrom` `ACSetTo`	Real	AC control setpoint on the respective side. Units are per unit for voltage or the factor for power factor control.
`AlossFrom` `AlossTo`	Real	The converter loss is modeled using the equation Loss = Aloss + Bloss*Idc Where Idc is the dc current in amps. In addition, a minimum converter loss can be specified.
`BlossFrom` `BlossTo`	Real	See Aloss explanation
`MinlossFrom` `MinlossTo`	Real	See Aloss explanation
`SmaxFrom` `SmaxTo`	Real	Maximum AC apparent power
`ImaxFrom` `ImaxTo`	Real	Maximum AC current in AC amps
`PWFFrom` `PWFTo`	Real	Power Weighting factor is a value between 0 and 1. It is used to reduce real and reactive power when limits are hit.
`MvarMaxFrom` `MvarMaxTo`	Real	Maximum Mvar injection into the network when AC mode is Voltage.
`MvarMinFrom` `MvarMinTo`	Real	Minimum Mvar injection into the network when AC mode is Voltage.
`RegBusNumFrom` `RegBusNumTo`	Integer	When AC model is Voltage, this is the regulated bus number on the respective side. If not specified then the terminal is regulated.
`RegFactorFrom` `RegFactorTo`	Real	The regulation factor used when the respective side shares var regulation with nearby generation.
`OwnerNum1`	Integer	Owner Number 1
`OwnerPerc1`	Real	Owner 1
`OwnerNum2`	Integer	Owner Number 2
`OwnerPerc2`	Real	Owner 2
`OwnerNum3`	Integer	Owner Number 3
`OwnerPerc3`	Real	Owner 3
`OwnerNum4`	Integer	Owner Number 4

OwnerPerc4	Real	Owner 4
OwnerNum5	Integer	Owner Number 5
OwnerPerc5	Real	Owner 5
OwnerNum6	Integer	Owner Number 6
OwnerPerc6	Real	Owner 6
OwnerNum7	Integer	Owner Number 7
OwnerPerc7	Real	Owner 7
OwnerNum8	Integer	Owner Number 8
OwnerPerc8	Real	Owner 8
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.
DataMaintainerInherit	String	Default value of this field is YES. Set to NO to prevent the object from inheriting its DataMaintainer from another object. [Added in Version 19, December 1, 2016]

A sample file section is shown as follows:

```
VSCDCLine (Name,Status,MWSetSide,R,BusNumFrom,DCModeFrom,ACModeFrom,DCSetFrom,ACSetFrom,
 AlossFrom,BlossFrom,MinlossFrom,SmaxFrom,ImaxFrom,PWFFrom,MvarMaxFrom,
 MvarMinFrom,RegBusNumFrom,RegFactorFrom,BusNumTo,DCModeTo,ACModeTo,DCSetTo,
 ACSetTo,AlossTo,BlossTo,MinlossTo,SmaxTo,ImaxTo,PWFTo,MvarMaxTo,MvarMinTo,
 RegBusNumTo,RegFactorTo,OwnerNum1,OwnerPerc1,OwnerNum2,OwnerPerc2,OwnerNum3,
 OwnerPerc3,OwnerNum4,OwnerPerc4,OwnerNum5,OwnerPerc5,OwnerNum6,OwnerPerc6,
 OwnerNum7,OwnerPerc7,OwnerNum8,OwnerPerc8,DataMaintainerAssign)
{
"MyVSC1" "Closed" "DC Side" 0.0 "" "Out-of-Service" "Voltage" 200.0 1.0 0.0 0.0 0.0 0.0
0.0 1.0 9999.0 -9999.0 "" 100.0 "" "Out-of-Service" "Voltage" 200.0 1.0 0.0 0.0 0.0 0.0
0.0 1.0 9999.0 -9999.0 "" 100.0 129 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
}
```

4.28 MTDCRecord, MTDCBus, MTDCConverter, MTDCTransmissionLine

A MTDCRecord an entire DC Transmission network. This consists of sub-record MTDCBus, MTDCConverter and MTDCTranmissionLine. The MTDCRecord fields in the basic power flow case format are as follows.

Field	Type	Description
Number	Integer	*KEY1* Record Number
Name	String	Record Name
VControlBus	String	Voltage Controlling AC Bus. When writing out this field, the option that is used to specify which key field to use in SUBDATA sections is used to identify the bus by either primary, secondary, or label identifiers. When reading from an AUX file any of these identifiers can be used to identify the bus.
Mode	String	Control Mode: Choices are • Current (Amps) • Power (MW)

		• Blocked
Status	String	Status. Either OPEN or CLOSED.
ModeSwitchVoltage	Real	Minimum DC Voltage for Power Control. Note, this same setting is also with each MTDCConverter. The MTDCConverter will use the maximum of its own value and the value with the MTDCRecord.
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.
DataMaintainerInherit	String	Default value of this field is YES. Set to NO to prevent the object from inheriting its DataMaintainer from another object. [Added in Version 19, December 1, 2016]

The MTDCBus fields in the basic power flow case format are as follows.

Field	Type	Description
Number	Integer	*KEY1* DC Bus Number
MTDCNumber	Integer	*KEY2* MTCD Record Num
Name	String	DC Bus Name
AreaNumber	Integer	DC Bus Area Num
BANumber	Integer	DC Bus Balancing Authority Num
OwnerNum1	Integer	Owner Number 1
ZoneNumber	Integer	DC Bus Zone Num

The MTDCConverter fields in the basic power flow case format are as follows.

Field	Type	Description
BusNumAC	Integer	*KEY1* AC Bus Number. When reading record, see special note in Section 2.4.
BusNumDC	Integer	*KEY2* DC Bus Number
MTDCNumber	Integer	*KEY3* MTDC Record Num
Status	String	Status
NumBridges	Integer	Number of bridges in the converter
FiringAngle	Real	Firing Angle
FiringAngleMax	Real	Firing Angle Max
FiringAngleMin	Real	Firing Angle Min
RComm	Real	Commutating Resistance
XComm	Real	Commutating Reactance
VacBase	Real	Transformer AC Winding Base Voltage
XFRatio	Real	Transformer Ratio
Tap	Real	Tap
TapMax	Real	Tap Max

TapMin	Real	Tap Min
TapStep	Real	Tap Step Size
FixedACTap	Real	Fixed AC Tap
Vdiode	Real	Diode voltage drop in kV across each bridge
Type	String	Type of Converter. Either Rect or Inv
Setpoint	Real	Setpoint Value of the converter. Units depend on the Set mode. Can be either MW, Amps, or voltage in kV
PartFact	Real	When multiple rectifiers or inverters exist, if one has to reduce its power or current order, the others of the same type pick up this reduction according to the DC particpation factor
DCMargin	Real	When the DC voltage schedule can not be achieved, the power or current order will not be reduced by more than this amount multiplied by the setpoint . Beyond this point, the voltage schedule will be reduced instead
ModeSwitchVoltage	Real	At a DC voltage below this, the converter will switch to current mode if presently in power mode.
LimitAmpEnforce	String	Set to YES to use and enforce the Current Rating field as the maximum allowed current. If enforcing the limit, setpoints will be reduced if the current exceeds the limit.
LimitAmp	Real	Current Rating
OwnerNum1	Integer	Owner Number 1
OwnerPerc1	Real	Owner 1
OwnerNum2	Integer	Owner Number 2
OwnerPerc2	Real	Owner 2
OwnerNum3	Integer	Owner Number 3
OwnerPerc3	Real	Owner 3
OwnerNum4	Integer	Owner Number 4
OwnerPerc4	Real	Owner 4
OwnerNum5	Integer	Owner Number 5
OwnerPerc5	Real	Owner 5
OwnerNum6	Integer	Owner Number 6
OwnerPerc6	Real	Owner 6
OwnerNum7	Integer	Owner Number 7
OwnerPerc7	Real	Owner 7
OwnerNum8	Integer	Owner Number 8
OwnerPerc8	Real	Owner 8

The MTDCTransmissionLine fields in the basic power flow case format are as follows.

Field	Type	Description
BusNumFrom	Integer	*KEY1* DC From Bus Number

BusNumTo	Integer	*KEY2* DC To Bus Number
Circuit	String	*KEY3* Circuit ID
MTDCNumber	Integer	*KEY4* MTDC Record Num
Status	String	Status
Aloss	Real	aLoss factor is used when the the is a tie-line. Set to 1.0 to assign all losses to the from side (equivalent to using the to side as the meter). Set to 0.0 to assign all losses to the side (equivalent to using the from side as the meter). Set to 0.5 to split the losses between the from and to side.
R	Real	DC Line Resistance
L	Real	DC Line Inductance
OwnerNum1	Integer	Owner Number 1
OwnerPerc1	Real	Owner 1
OwnerNum2	Integer	Owner Number 2
OwnerPerc2	Real	Owner 2
OwnerNum3	Integer	Owner Number 3
OwnerPerc3	Real	Owner 3
OwnerNum4	Integer	Owner Number 4
OwnerPerc4	Real	Owner 4
OwnerNum5	Integer	Owner Number 5
OwnerPerc5	Real	Owner 5
OwnerNum6	Integer	Owner Number 6
OwnerPerc6	Real	Owner 6
OwnerNum7	Integer	Owner Number 7
OwnerPerc7	Real	Owner 7
OwnerNum8	Integer	Owner Number 8
OwnerPerc8	Real	Owner 8

A sample file section is shown as follows:

```
MTDCRecord (Number,Name,VControlBus,Mode,Status,ModeSwitchVoltage,DataMaintainerAssign)
{
 1 "MTDC Line 1" "12311" "Power (MW)" "Closed" 0.0 ""
 2 "MTDC Line 2" "12312" "Power (MW)" "Closed" 0.0 ""
}
MTDCBus (Number,MTDCNumber,Name,AreaNumber,BANumber,OwnerNum1,ZoneNumber)
{
   1 1 "MyHill1X" 26 26   7 260
   2 1 "MyVall1X" 26 26 123 260
 101 1 "MyHill1R" 26 26  81 260
 102 1 "MyVall1I" 26 26  81 260
  21 2 "MyHill2X" 26 26   7 260
  22 2 "MyVall2X" 26 26 123 260
 121 2 "MyHill2R" 26 26  81 260
 122 2 "MyVall2I" 26 26  81 260
}
MTDCConverter (BusNumAC,BusNumDC,MTDCNumber,Status,NumBridges,FiringAngle,FiringAngleMax,
 FiringAngleMin,RComm,XComm,VacBase,XFRatio,Tap,TapMax,TapMin,TapStep,
 FixedACTap,Vdiode,Type,Setpoint,PartFact,DCMargin,ModeSwitchVoltage,
 LimitAmpEnforce,LimitAmp,OwnerNum1,OwnerPerc1,OwnerNum2,OwnerPerc2,OwnerNum3,
 OwnerPerc3,OwnerNum4,OwnerPerc4,OwnerNum5,OwnerPerc5,OwnerNum6,OwnerPerc6,
```

```
 OwnerNum7,OwnerPerc7,OwnerNum8,OwnerPerc8)
{
 12314 101 1 "Closed" 2 16.0229 17.50 15.0 0.0 11.8510 345.0 0.597101 1.080648 1.15010
0.92490 0.0 1.0 0.60 "Rect" 844.250 1600.0 0.064 0.0 "NO " 1600.0 123 100.0 "" "" "" ""
"" "" "" "" "" "" "" "" "" ""
 12311 102 1 "Closed" 2 16.9999 90.0 17.0 0.0 11.2720 500.0 0.40480 1.098098 1.20 0.9250
0.0 1.0 0.570 "Inv" 466.50 1600.0 "" 0.0 "NO " 1600.0 123 100.0 "" "" "" "" "" "" "" ""
"" "" "" "" "" ""
 12316 121 2 "Closed" 2 16.0229 17.50 15.0 0.0 11.8510 345.0 0.597101 1.080648 1.15010
0.92490 0.0 1.0 0.60 "Rect" 844.250 1600.0 0.064 0.0 "NO " 1600.0 123 100.0 "" "" "" ""
"" "" "" "" "" "" "" "" "" ""
 12312 122 2 "Closed" 2 16.9999 90.0 17.0 0.0 11.2720 500.0 0.40480 1.098098 1.20 0.9250
0.0 1.0 0.570 "Inv" 466.50 1600.0 "" 0.0 "NO " 1600.0 123 100.0 "" "" "" "" "" "" "" ""
"" "" "" "" "" ""
}
MTDCTransmissionLine (BusNumFrom,BusNumTo,Circuit,MTDCNumber,Status,Aloss,R,L,
 OwnerNum1,OwnerPerc1,OwnerNum2,OwnerPerc2,OwnerNum3,OwnerPerc3,OwnerNum4,OwnerPerc4,
 OwnerNum5,OwnerPerc5,OwnerNum6,OwnerPerc6,OwnerNum7,OwnerPerc7,OwnerNum8,OwnerPerc8)
{
   1  2 "1" 1 "Closed" 1.0 7.82 0.694 123 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" ""
 101  1 "1" 1 "Closed" 1.0 0.02 500.0 123 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" ""
 102  2 "1" 1 "Closed" 1.0 0.02 500.0 123 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" ""
  21 22 "1" 2 "Closed" 1.0 7.82 0.694 123 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" ""
 121 21 "1" 2 "Closed" 1.0 0.02 500.0 123 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" ""
 122 22 "1" 2 "Closed" 1.0 0.02 500.0 123 100.0 "" "" "" "" "" "" "" "" "" "" "" "" "" ""
}
```

4.29 InjectionGroup, PartPoint

An InjectionGroup represents a collection of generators, loads, switched shunts, or other injection groups. A primary purpose when modeling contingencies is to use an injection group to model generator drop schemes. The fields recognized by the InjectionGroup object type are shown in the following table.

Field	**Type**	**Description**
`Name`	String	*KEY1* Name of the Injection Group. There can be only one Injection Group with this name.
`DataMaintainerAssign`	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.

Injection Groups are made up of any number of PartPoint objects. The fields recognized by the PartPoint object type are shown in the following table:

Field	Description and Rules for Field
`GroupName`	*KEY2* Name of the Injection Group to which this PartPoint belongs
`Object`	*KEY1* Identifies the object for this injection point using the object string in format specified in Section **2.3**. The object type must be either Gen, Load, Shunt or InjectionGroup.
`AutoCalcMethod`	Specifies how the participation factor of this object is determined. The choices depend on the object type as shown below. GEN *SPECIFIED* : Set to the numerical value in field PartFact *MAX GEN MW* : Set equal to the Maximum MW Output *MAX GEN INC* : Set equal to the (Max MW – Present MW) *MAX GEN DEC* : Set equal to the (Present MW – Min MW)

	Field Name : Set equal to the value of this generator field. Model Expression Name : Set equal to the value of this Model Expression LOAD *SPECIFIED* : Set to the numerical value in field PartFact *LOAD MW* : Set Equal to the Load MW Field Name : Set equal to the value of this load field. Model Expression Name : Set equal to the value of this Model Expression. SHUNT *SPECIFIED* : Set to the numerical value in field PartFact *MAX SHUNT MVAR* : Set equal to the Maximum Mvar *MAX SHUNT INC* : Set equal to the (Max Mvar – Present Mvar) *MAX SHUNT DEC* : Set equal to the (Present Mvar – Min Mvar) Field Name : Set equal to the value of this shunt field. Model Expression Name : Set equal to the value of this Model Expression. INJECTIONGROUP *SPECIFIED* : Set to the numerical value in field PartFact Field Name : Set equal to the value of this injection group field. Model Expression Name : Set equal to the value of this Model Expression.
`PartFact`	Present numeric value for the participation factor. It may be automatically recalculated based on the AutoCalcMethod and AutoCalc fields.
`AutoCalc`	Set to `YES` to specify that the participation factor value should be automatically recalculated before every use of the injection group based on the AutoCalcMethod. If set to `NO`, the PartFact should be specified.

A sample file section is shown as follows:

```
InjectionGroup (Name, DataMaintainerAssign)
{
"Boundary Generators" "DM1"
"Bridger Generators"  "DM1"
}



PartPoint (GroupName, Object,        AutoCalcMethod, PartFact, AutoCalc)
{
"Boundary Generators" "GEN 46464 51" "MAX GEN MW"     50       "YES"
"Boundary Generators" "GEN 46466 52" "MAX GEN MW"     50       "YES"
"Boundary Generators" "GEN 46468 53" "MAX GEN MW"     50       "YES"
"Bridger Generators"  "GEN 65386 1"  "SPECIFIED"     557.00    "NO"
"Bridger Generators"  "GEN 65387 1"  "SPECIFIED"     557.00    "NO"
"Bridger Generators"  "GEN 65388 1"  "SPECIFIED"     557.00    "NO"
"Bridger Generators"  "GEN 65389 1"  "SPECIFIED"     557.00    "NO"
}
```

4.30 SuperArea

SuperArea fields in the basic power flow case format are as follows.

Field	Type	Description
`Name`	String	*KEY1* Name of the super area. A string of any length is permitted
`AGC`	String	String indicating what type of automatic control the area uses when performing solutions. Options are Off AGC, Part. AGC, ED, and OPF
`AGCTolerance`	Real	Tolerance in MW used when performing Super Area control
`DataMaintainerAssign`	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.

A sample file section is shown as follows:

```
SuperArea (Name, AGC, AGCTolerance, DataMaintainerAssign)
{
  "CALIFORNIA" "Part. AGC"  2.0 "WECC"
  "SOUTHWEST"  "Part. AGC"  2.0 "WECC"
}
```

4.31 Area

Area fields in the basic power flow case format are as follows. When reading a record that does not contain the primary key (`Number`), but does contain the secondary key (`Name`) with a string that does not describe an existing object, then the `Number` primary key will be automatically set to the maximum `Number` for this object type plus 1.

Field	Type	Description
`Number`	Integer	*KEY1* Number of the area. This is the unique identifier for the Area and must be unique in the case. Any integer between 1 and 2,147,483,647 is permitted.
`Name`	String	*SECONDARY KEY1* Name of the area. A string of any length is permitted.
`SuperArea`	String	Name of the Super Area to which the Area Belongs
`MonitorLimits`	String	Set to `YES` to specify that objects belonging to this area be monitored for violations. Set to `NO` to not monitor objects in the area. Devices such as a branch that ties two areas together may be monitored if either one of its terminal areas are monitored.
`MonitorMaxkV`	Real	Specify the maximum nominal kV level at which monitoring is done. For branches that are tie-lines, the branch may be monitored if either terminal bus meets the voltage range.

`MonitorMinkV`	Real	Specify the minimum nominal kV level at which monitoring is done.
`AutoControlShunt`	String	Either YES or NO. Set to YES to allow switched shunt control for shunts in this area.
`AutoControlXF`	String	Either YES or NO. Set to YES to allow transformer tap control for transformers in this area.
`AGC`	String	String indicating what type of automatic control the area uses when performing solutions. Options are Off AGC Part. AGC ED OPF Area Slack IG Slack
`ExportMWUnspecified`	Read	The amount of MW export which is not specified as part of the StudyMWTransaction table. The total net export for an area is this value added to the sum of the transactions in the StudyMWTransaction Table.
`SlackBus`	String	String indicating the bus which acts as the slack when area uses *Area Slack* control. String may be either the number of the bus, the name of the bus, the name_kV of the bus, or the label.
`AGCTolerance`	Real	Tolerance in MW used when performing Area control
`EnforceGenMWLimits`	String	Either YES or NO. Set to YES to enforce generator MW Limits when performing automatic dispatch routines in Simulator.
`SlackInjectionGroup`	String	Name of the Injection Group which is used to redispatch the area when the AGC = IG Slack.
`SlackIGUseConstantPowerFactor`	String	Either YES or NO. Set to YES to always scale loads using a constant power factor. Otherwise the chan
`SlackIGPowerFactor`	Real	Power factor of the change in load. The change in Mvar will be calculated from the change in MW using this power factor.
`SlackIGPowerFactorQMult`	Real	When using the `SlackIGPowerFactor` value, multiply the change in Mvar by this value.
`SlackIGEnforcePosLoad`	String	Either YES or NO. Set to YES to prevent loads from going negative when being dispatched as part of the injection group.
`SlackIGEnforceMWLimits`	String	Either YES or NO. If Set to YES, then generators in the injection group will enforce their MW

		limits regardless of the global case and area options for generator MW limit enforcement
SlackIGEnforceAGC	String	Either YES or NO. If set to YES, only generators and loads in the injection group that are designated as available for AGC control will be allowed to participate in injection changes.
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.

A sample file section is shown as follows:

```
Area (Number,Name,SuperArea,MonitorLimits,MonitorMaxkV,MonitorMinkV,AutoControlShunt,
 AutoControlXF,AGC,ExportMWUnspecified,SlackBus,AGCTolerance,
 EnforceGenMWLimits,SlackInjectionGroup,SlackIGUseConstantPowerFactor,
 SlackIGPowerFactor,SlackIGPowerFactorQMult,SlackIGEnforcePosLoad,
 SlackIGEnforceMWLimits,SlackIGEnforceAGC,DataMaintainerAssign)
{
 10 "Name A" "" "YES" 9999.0 0.0 "YES" "YES" "Area Slack" 0.0 "10321" 1.0 "YES" "" "YES"
1.0 1.0 "NO " "NO " "NO " ""
 11 "Name B" "" "YES" 9999.0 0.0 "YES" "YES" "Area Slack" 0.0 "11135" 1.0 "YES" "" "YES"
1.0 1.0 "NO " "NO " "NO " ""
 14 "Name C" "" "YES" 9999.0 0.0 "YES" "YES" "Area Slack" -0.001 "15926" 1.0 "YES" ""
"YES" 1.0 1.0 "NO " "NO " "NO " ""
}
```

4.32 StudyMWTransaction

StudyMWTransaction fields in the basic power flow case format are as follows.

Field	Type	Description
NumberExport	Integer	*KEY1* Number of the area which is the exporter
NumberImport	Integer	*KEY2* Number of the area which is the importer
ID	String	*KEY3* ID string for the StudyMWTransaction (no limit on string length)
MW	String	Scheduled Export in MW being sent from the Exporter to the Importer
Enabled	String	Set to YES to indicate the Transaction should be used in the calculation of the Area Control Error (ACE) used when performing automatic dispatch in the software. Set to NO to ignore the transaction
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.

A sample file section is shown as follows:

```
StudyMWTransactions(NumberExport,NumberImport,ID,MW,Enabled,DataMaintainerAssign)
{
 10  14 "A"    59.3 "YES" "WECC"
 10  11 "B"   313.1 "YES" "WECC"
 14  70 "C"    34.2 "YES" "WECC"
}
```

4.33 BalancingAuthority

BalancingAuthority fields in the basic power flow case format are as follows. When reading a record that does not contain the primary key (`Number`), but does contain the secondary key (`Name`) with a string that does not describe an existing object, then the `Number` primary key will be automatically set to the maximum `Number` for this object type plus 1.

Field	Type	Description
`Number`	Integer	*KEY1* Number of the BalancingAuthority. This is the unique identifier for the BalancingAuthority and must be unique in the case. Any integer between 1 and 2,147,483,647 is permitted.
`Name`	String	*SECONDARY KEY1* Name of the BalancingAuthority. A string of any length is permitted
`AGC`	String	String indicating what type of automatic control the area uses when performing solutions. Options are Off AGC Part. AGC BA Slack
`ExportMWUnspecified`	Read	The total net MW export for the balancing authority. Note that unlike Areas, the Balancing authorities do not have any specified transactions.
`SlackBus`	String	String indicating the bus which acts as the slack when area uses *Area Slack* control. String may be either the number of the bus, the name of the bus, or the name_kV of the bus
`AGCTolerance`	Real	Tolerance in MW used when performing Area control
`DataMaintainerAssign`	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.

A sample file section is shown as follows:

```
BalancingAuthority (Number,Name,AGC,ExportMWUnspecified,SlackBus,AGCTolerance,
   DataMaintainerAssign)
{
10 "My Name"    "BA Slack"   180.9 "10321" 1.0 ""
11 "Your Name"  "BA Slack" -1023.0 "11135" 1.0 ""
14 "Other Name" "BA Slack"  5033.3 "15926" 1.0 ""
}
```

4.34 Zone

Zone fields in the basic power flow case format are as follows. When reading a record that does not contain the primary key (`Number`), but does contain the secondary key (`Name`) with a string that does not describe an existing object, then the `Number` primary key will be automatically set to the maximum `Number` for this object type plus 1.

Field	Type	Description
`Number`	Integer	*KEY1* Number of the zone. This is the unique identifier for the zone and must be unique in the case. Any integer

		between 1 and 2,147,483,647 is permitted.
Name	String	*SECONDARY KEY1* Name of the area. A string of any length is permitted.
MonitorLimits	String	Set to YES to specify that objects belonging to this zone be monitored for violations. Set to NO to not monitor objects in the zone. Devices such as a branch that ties two zone together may be monitored if either one of its terminal areas are monitored.
MonitorMaxkV	Real	Specify the maximum nominal kV level at which monitoring is done. For branches that are tie-lines, the branch may be monitored if either terminal bus meets the voltage range.
MonitorMinkV	Real	Specify the minimum nominal kV level at which monitoring is done.
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.

A sample file section is shown as follows:

```
Zone (Number,Name,MonitorLimits,MonitorMaxkV,MonitorMinkV,DataMaintainerAssign)
{
  0 "ZONE0"  "YES" 9999.0 0.0 ""
  1 "ZONE1"  "YES" 9999.0 0.0 ""
 11 "ZONE11" "YES" 9999.0 0.0 ""
 10 "ZonePN" "YES" 9999.0 0.0 ""
}
```

4.35 Interface, InterfaceElement

An Interface represents a collection of Branch, DCTransmissionLine, InjectionGroup, Gen, Load, MultiSectionLine, and Interface object, as well as potentially some BranchOpen or BranchClose entries which represent the contingent elements. The fields recognized by the Interface object type are shown in the following table.

Field	Type	Description
Name	String	*KEY1* Name of the Interface
Number	Integer	Number of the Interface
MonDirection	String	Direction in which the interface is monitored: Either FROM -> TO, TO -> FROM
MonBoth	String	Set to YES to monitor both directions
MWOffSet	Real	Number specifying a shift applied to the value reported as
Monitor	String	Set to YES to specify that this interface should be monitored. Set to NO to not monitor this interface.
LimitSet	String	Name of the Limit Set to which the interface belongs
LimitMWA...LimitMWO	Real	15 MW limits associated with the interface
DataMaintainerAssign	String	Name of the DataMaintainer specifically assigned to this object. This can be blank as well. For objects which inherit their DataMaintainer this will still be blank.

Interfaces are made up of any number of InterfaceElement objects. The fields recognized by the InterfaceElement object type are shown in the following table:

Field	Type	Description
`InterfaceName`	String	*KEY1* Name of the Interface to which the InterfaceElement Belongs

`Element`	Integer	*KEY2* Description of the value measured for this InterfaceElement
		There are eleven kinds of elements allowed in an interface. Please note that the direction specified in the monitoring elements is important.
		• `"BRANCH num1 num2 ckt"` : Monitor the MW flow on the branch starting from bus num1 going to bus num2 with circuit ckt. (order of bus numbers defines the direction)
		• `"AREA num1 num2"` : Monitor the sum of the AC branches that connect area1 and area2.
		• `"ZONE num1 num2"` : Monitor the sum of the AC branches that connect zone1 and zone2.
		• `"BRANCHOPEN num1 num2 ckt"` : When monitoring the elements in this interface, monitor them under the contingency of opening this branch.
		• `"BRANCHCLOSE num1 num2 ckt"` : When monitoring the elements in this interface, monitor them under the contingency of closing this branch.
		• `"DCLINE num1 num2 ckt"` : Monitor the flow on a DC line.
		• `"INJECTIONGROUP 'name'"` : Monitor the net injection from an injection group (generation contributes as a positive injection, loads as negative).
		• `"GEN num1 id"` : Monitor the net injection from a generator (output is positive injection)
		• `"LOAD num1 id"` : Monitor the net injection from a load (output is negative injection).
		• `"MSLINE num1 num2 ckt"` : Monitor the MW flow on the multi-section line starting from bus num1 going to bus num2 with circuit ckt.
		• `"INTERFACE 'name'"` : Monitor the MW flow on the interface given by name.
		
		Note: bus# values may be replaced by a string enclosed in single quotes where the string is the name of the bus followed by an underscore character and then the nominal voltage of the bus. Labels may also be used as follows.
		    • bus# values for all element types may be replaced by a string enclosed in single quotes where the string is the label of the bus.
		    • for GEN or LOAD elements, the section *num1 id* may be

		replaced by the device's label. • For MSLINE, DCLINE, or BRANCH elements, the *num1 num2 ckt* section may be replaced by the device's label.
`MeterFar`	String	Set to YES to meter at the far end of the Element. Otherwise set to NO. Note that the ***direction*** of the element is determined by the Element field. This only determines which end to meter and not the directionality of the measurement.
`Weight`	Real	A number by which the measured flow will be multiplied.

A sample file section is shown as follows:

```
INTERFACE (Number,Name,MonDirection,MWOffset,MonBoth,LimitMWA,LimitMWB,
           LimitMWC,LimitMWD,LimitMWE,LimitMWF,LimitMWG,LimitMWH,LimitMWI,
           LimitMWJ,LimitMWK,LimitMWL,LimitMWM,LimitMWN,LimitMWO,
           DataMaintainerAssign)
{
  1 "ALBERTA - BRITISH COLUMBIA " "FROM -> TO" 0.0000 "NO " 1000.0 -1200.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 "DM1"
  2 "ALBERTA - SASKATCHEWAN "     "FROM -> TO" 0.0000 "NO "  150.0  -150.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 "DM1"
 76 "ALTURAS PROJECT"             "FROM -> TO" 0.0000 "NO "  300.0  -300.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 "DM1"
}

INTERFACEELEMENT (InterfaceName,Element,MeterFar,Weight)
{
"ALBERTA - BRITISH COLUMBIA" "BRANCH 54456 50791 1"  "NO " 1.000000
"ALBERTA - BRITISH COLUMBIA" "BRANCH 54232 50776 1"  "YES" 1.000000
"ALBERTA - BRITISH COLUMBIA" "BRANCH 54329 50830 1"  "NO " 1.000000
"ALBERTA - SASKATCHEWAN"     "BRANCH 54674 55473 30" "YES" 1.000000
"ALTURAS PROJECT"            "BRANCH 40537 64058 1"  "NO " 1.000000
}
```

4.36 Sim_Solution_Options_Value

The Sim_Solution_Options_Value records are a list of the various solution options used in the power flow solution algorithm. Each power flow solution option (Sim_Solution_Options_Value) has a unique string identifier and then a particular set of potential values associated with them. The field names for the object are simply *Option* and *Value*. The following table shows a list of the string identifiers for each option and a description of what the value should contain.

Option	Value
AGCTolerance	Island AGC Convergence Tolerance Note: In the AUX file, this value is written in per unit. Thus if SBase=100, then a 5 MW tolerance should be written as 0.05
AGCToleranceMVA	Island AGC Convergence Tolerance in MVA. Same value as AGCTolerance but units are in MVA.
ChkDFACTS	UNCHECK box in the GUI dialog and set to YES in auxiliary file or case information display to allow automatic DFACTS control in the voltage control loop.
ChkMWAGC	UNCHECK box in the GUI dialog and set to YES in auxiliary file or case information display to perform the MW Control Loop to balance load and generation. Normally this is done by Area and/or SuperArea, but Island-Based AGC is also possible.
ChkPhaseShifters	UNCHECK box in the GUI dialog and set to YES in auxiliary file or case information display to allow automatic phase shifter control in the voltage control loop.
ChkShunts	UNCHECK box in the GUI dialog and set to YES in auxiliary file or case information display to allow automatic switched shunt control in the voltage control loop.
ChkSVCs	UNCHECK box in the GUI dialog and set to YES in auxiliary file or case information display to allow automatic SVC (static var compensator) control in the voltage control loop.
ChkTaps	UNCHECK box in the GUI dialog and set to YES in auxiliary file or case information display to allow automatic transformer tap artio control in the voltage control loop.
ChkVarBackoffImmediately	Set to YES to check whether to backoff generator Mvar limits in the inner power flow loop. This means that hitting a generator Mvar limit will not be evaluated after each inner loop iteration (but will still be handled in the voltage control loop).
ChkVarImmediately	Set to YES to check generator Mvar limits in the inner power flow loop. This means that both backing off limits and hitting a generator Mvar limit will be evaluated after each inner loop iteration.
CloseCBToEnergizeShunts	Set to YES to allow switched shunts that are presently not energized to participate in automatic switched shunt control if they can be energized by closing breakers (or load break disconnects).
ConsolidationUse	Set to YES to automatically perform topology processing at the beginning of solution activities to consolidate branches marked for consolidation.

CTGInterfaceEnforcement	Contingent Interface Inclusion (Never = never enforce flows; PowerFlow = only enforce in power flow and OPF; 2 : CTG = also enforce in Contingency and SCOPF)
DCApprox	Set to YES to assume a DC approximation (the DC power flow)
DCLossComp	Use User-Defined Loss Sens for DC OPF and ED
DCModelType	Set to RIgnore to assume the series resistance (r) is zero. Set to GIgnore to assume the series conductange (g) is zero
DCMvarComp	Set to YES to Use Constant Voltages to Adjust Branch Limits in DC Power Flow Solution
DCPhaseIgnore	Set to YES to ignore the Phase Shifter Term that could be introduced in the DC approximation. Normally this should be ignored.
DCXFCorrIgnore	Set to YES to ignore the transformer impedance correction in the DC approximation. Normally this should be ignored.
DisableAngleRotation	Set to YES to disable voltage angle rotation. Normally at the end of a power flow solution angles are brought inside of +/- 160 degrees if possible.
DisableAngleSmoothing	Set to YES to disable the angle smoothing that is done as a preprocess to the power flow. Angle smoothing attempts to reduce large angle differences across branches that have been been closed in. Normally this option should NOT be disabled.
DisableGenMVRCheck	Set to YES to ignore generator Mvar limits.
DisableOptMult	Set to YES to disable the optimal multiplier in the inner power flow loop iterations. Normally this should NOT be done.
DisableRestoreSolution	Disable Restore last successful solution
DisableRestoreState	Disable Restore State before failed solution attempt
DisableXFTapWrongSign	Set this to YES to disable transformer control when a transformer is regulating one of its own terminal buses and the tap sensitivity is the wrong sign. If regulating the from bus the correct sign is positive and when regulating the to bus the correct sign is negative. This should normally be set to YES.
DoOneIteration	Do only one inner power flow loop iteration
DynAssignSlack	Set to YES to allow Simulator to determine automatically add or remove a slack buses as system topology changes. Preference is given to the buses chosen by the user to be slack buses while in Edit Mode. Otherwise the generation with the largest maximum MW is given preference.
EconDispEnforceConvex	Enforce Convex Cost Curves in the economic dispatch. If a generator ends up in a part of its cost curve that is not convex then the generator AGC status will be set to NO
EconDispPenaltyFactors	Include Penalty Factors in the economic dispatch to account for losses
EnforceGenMWLimits	Set to YES to enforce the generator MW limits. Note that for economic modeling such as in the ED or OPF, generators MW limits are always enforced regardless of this option.
FlatStartInit	Initialize the system to a flat start at the beginning of each power

	flow solution
LogID	Message Log Bus Identifier
LogLTCColor	Color LTC Messages
LogLTCSuppress	Suppress LTC Messages
LogMvarColor	Color Gen MVAR Messages
LogMvarSuppress	Suppress Gen MVAR Messages
LogMWColor	Color AGC Messages
LogMWSuppress	Suppress AGC Messages
LogNomkV	Message Log Include Nominal Voltage
LogOPFLPColor	Color LP Variable Messages
LogOPFLPSuppress	Suppress LP Variable Messages
LogPhaseColor	Color Phase Shifter Messages
LogPhaseSuppress	Suppress Phase Shifter Messages
LogShuntColor	Color Switched Shunt Messages
LogShuntSuppress	Suppress Switched Shunt Messages
LossSenseFunc	Loss Sensitivity Function
LTCTapBalance	UNCHECK box in the GUI dialog and set to YES in auxiliary file or case information display so that the power flow solution will always try to maintain a balance of tap ratios for transformers that are in parallel with each other.
MaxItr	Maximum number of iterations in the inner power flow loop
MaxItrVoltLoop	Maximum number of Voltage Control Loop Iterations
MinLTCSense	Transformer Tap ratios with voltage to tap sensitivities smaller than this value will not attempt to control the regulated bus voltage. If the transformer sensitivity later improves the transformer will automatically regain control.
MinVoltILoad	Minimum Per Unit Voltage for Constant Current Loads. If the voltage at the terminal bus falls below this value the load will decrease using a cosine function towards a value of zero load at zero voltage.
MinVoltSLoad	Minimum Per Unit Voltage for Constant Power Loads. If the voltage at the terminal bus falls below this value the load will decrease using a cosine function towards a value of zero load at zero voltage.
ModelPSDiscrete	Set to YES to force phase-shifters to have angles at the discrete steps defined. For most modeling, it is recommended the this be set to NO.
MVABase	System MVABase
MVAConvergenceTol	Convergence Tolerance in MVA
MvarSharingAllocation	Determines how generators regulating the same bus share the Mvars needed to maintain the voltage. Options are RegPerc, MinMaxRange, and SumRegPerc.
MWAGCAreaIsland	Island Based AGC Type (Choices are U, G, A or I meaning Use Area/SuperArea, Gen, Area, or Injection Group)

MWAGCIGEnforceGenMWLimits	Island-based AGC Injection Group Enforce Generator Limits
MWAGCIGName	Island-based AGC Injection Group
MWAGCIGOnlyAGCAble	Island-based AGC Injection Group Enforce Generator AGC
MWAGCIGPositiveLoad	Island-based AGC Injection Group Enforce Positive Load
MWAGCIGPowerFact	Power Factor used When Scaling Load for Island-Based AGC
MWAGCIGQMult	Multiplier for Mvar When Scaling Load for Island-Based AGC
MWAGCIGUseContantPF	Use Constant Power Factor When Scaling Load for Island-Based AGC
PreventOscillations	Set to YES to prevent Generator Mvar limit, Transformer Tap Ratio, Phase Shifters, and Switched Shunt controller oscillations. If one of these devices begins to oscillate the control will be turned off.
PUConvergenceTol	Convergence Tolerance. Note: In the AUX file, this value is written in per unit. Thus if SBase=100, then a 0.1 MVA tolerance should be written as 0.001
ShuntInner	UNCHECK box in the GUI dialog and set to YES in auxiliary file or case information display to allow continuous switched shunts to be treated as PV buses in the inner power flow loop.
TransformerStepping	If value is "Coordinated", then transformers switching control will be coordinated between all transformers. If the value is "Self", then each transformer only looks at its own control.
ZBRThreshold	This is the per unit impedance threshold below which Simulator will automatically determine groupings of buses which are connected by very low impedance branches. This effects the treatment of voltage regulation for devices which regulate a bus in this grouping.

A sample file section is shown as follows:

```
Sim_Solution_Options_Value (Option,Value)
{
"AGCToleranceMVA"                  "5"
"ChkDFACTS"                        "NO"
"ChkMWAGC"                         "NO"
"ChkPhaseShifters"                 "NO"
"ChkShunts"                        "NO"
"ChkSVCs"                          "NO"
"ChkTaps"                          "NO"
"ChkVarBackoffImmediately"         "NO"
"ChkVarImmediately"                "NO"
"CloseCBToEnergizeShunts"          "NO"
"ConsolidationUse"                 "NO"
"CTGInterfaceEnforcement"          "PowerFlow"
"DCApprox"                         "NO"
"DCLossComp"                       "NO"
"DCModelType"                      "RIgnore"
"DCMvarComp"                       "NO"
"DCPhaseIgnore"                    "YES"
"DCXFCorrIgnore"                   "YES"
"DisableAngleRotation"             "NO"
"DisableAngleSmoothing"            "NO"
"DisableGenMVRCheck"               "NO"
"DisableOptMult"                   "NO"
"DisableRestoreSolution"           "NO"
"DisableRestoreState"              "NO"
"DisableXFTapWrongSign"            "YES"
```

```
"DoOneIteration"                  "NO"
"DynAssignSlack"                  "YES"
"EconDispEnforceConvex"           "NO"
"EconDispPenaltyFactors"          "YES"
"EnforceGenMWLimits"              "YES"
"FlatStartInit"                   "NO"
"LogID"                           "Numbers"
"LogLTCColor"                     "32768"
"LogLTCSuppress"                  "NO"
"LogMvarColor"                    "8388736"
"LogMvarSuppress"                 "NO"
"LogMWColor"                      "16711680"
"LogMWSuppress"                   "NO"
"LogNomkV"                        "NO"
"LogOPFLPColor"                   "16711680"
"LogOPFLPSuppress"                "YES"
"LogPhaseColor"                   "8388863"
"LogPhaseSuppress"                "NO"
"LogShuntColor"                   "33023"
"LogShuntSuppress"                "NO"
"LossSenseFunc"                   "None"
"LTCTapBalance"                   "YES"
"MaxItr"                          "50"
"MaxItrVoltLoop"                  "20"
"MinLTCSense"                     "0.01"
"MinVoltILoad"                    "0.5"
"MinVoltSLoad"                    "0.7"
"ModelPSDiscrete"                 "NO"
"MVABase"                         "100"
"MVAConvergenceTol"               "0.1"
"MvarSharingAllocation"           "SumRegPerc"
"MWAGCAreaIsland"                 "Use Area/SuperArea ACE"
"MWAGCIGEnforceGenMWLimits"       "NO"
"MWAGCIGName"                     ""
"MWAGCIGOnlyAGCAble"              "NO"
"MWAGCIGPositiveLoad"             "NO"
"MWAGCIGPowerFact"                "1"
"MWAGCIGQMult"                    "1"
"MWAGCIGUseContantPF"             "NO"
"PreventOscillations"             "YES"
"ShuntInner"                      "YES"
"TransformerStepping"             "Coordinated"
"ZBRThreshold"                    "0.00029"
}
```

4.37 Nomogram

A nomogram represents a 2-dimensional limit boundary. Each dimension is defined by a separate Interface specification. These are described using a SUBDATA section named InterfaceAElement and InterfaceBElement. The 2-dimensional boundary is then defined in a third SUBDATA section named NomogramBreakPoint. Besides these 3 SUBDATA sections, the fields recognized by the Nomogram object type are shown in the following table.

Field	Type	Description
`Name`	String	*KEY1* Name of the Interface
`Monitor`	String	Set to `YES` to specify that this interface should be monitored. Set to `NO` to not monitor this interface.

`LimitSet`	String	Name of the Limit Set to which the interface belongs

For more details on the SUBDATA sections, see the Auxiliary File format description on the PowerWorld website at
http://www.powerworld.com/WebHelp/#http://www.powerworld.com/files/Auxiliary-File-Format.pdf

A sample file section is shown as follows:

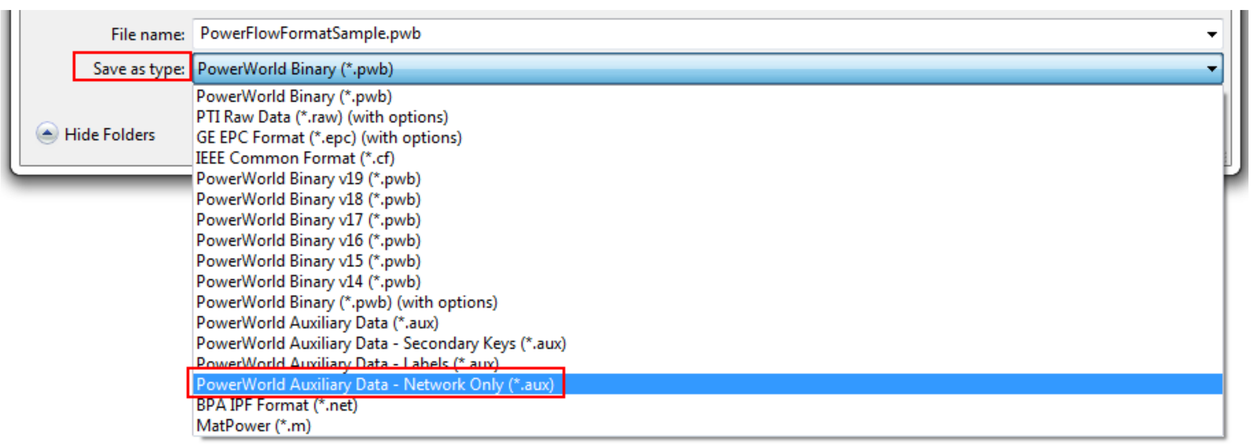
```
Nomogram (Nomogram, Monitor, LimitSet)
{
"My Nomo Name" "YES" "MyLimitSet"
    <SUBDATA InterfaceAElement
      "BRANCH  8   9 1" NO
      "BRANCH 12  33 1" YES
    </SUBDATA>
    <SUBDATA InterfaceBElement
      "BRANCH  4   5 1" NO
      "BRANCH 66  77 1" YES
    </SUBDATA>
    <SUBDATA NomogramBreakPoint>
     //  LimA   LimB
         -100    -20
         -100    100
           80     50
           60    -10
    </SUBDATA>
}
```

5 Using PowerWorld Simulator to Save this Information

Within PowerWorld Simulator there are 2 methods for saving a model to a text file format that has all the information described in this document.

5.1 File, Save As Method

The first most direct method is to simply choose File, Save As and then choose a new special type called “PowerWorld Auxiliary Data – Network Only (*.aux)” as shown in the following figure.

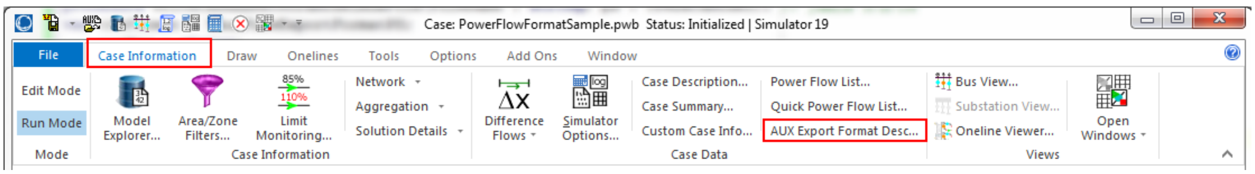


After choosing this option then Simulator will write out this information. Not also that using this option will always write out an auxiliary file using the option **Use Concise Variable Names and Auxiliary File Headers** as described in Simulator's online help documentation at the following links.

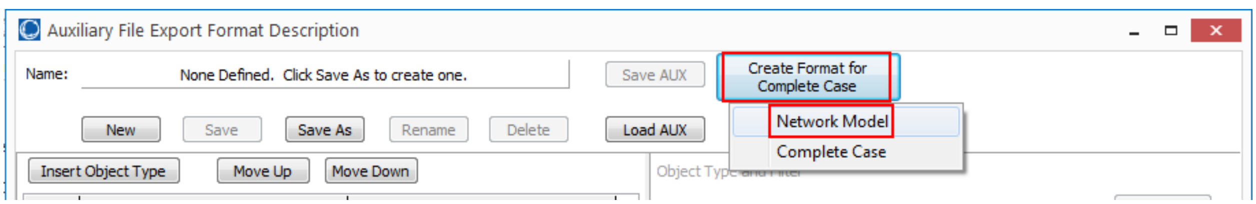
- http://www.powerworld.com/WebHelp/#MainDocumentation_HTML/Auxiliary_Files.htm
- http://www.powerworld.com/WebHelp/#MainDocumentation_HTML/Case_Information_Display_Options.htm

5.2 Create an Auxiliary File Export Format Description

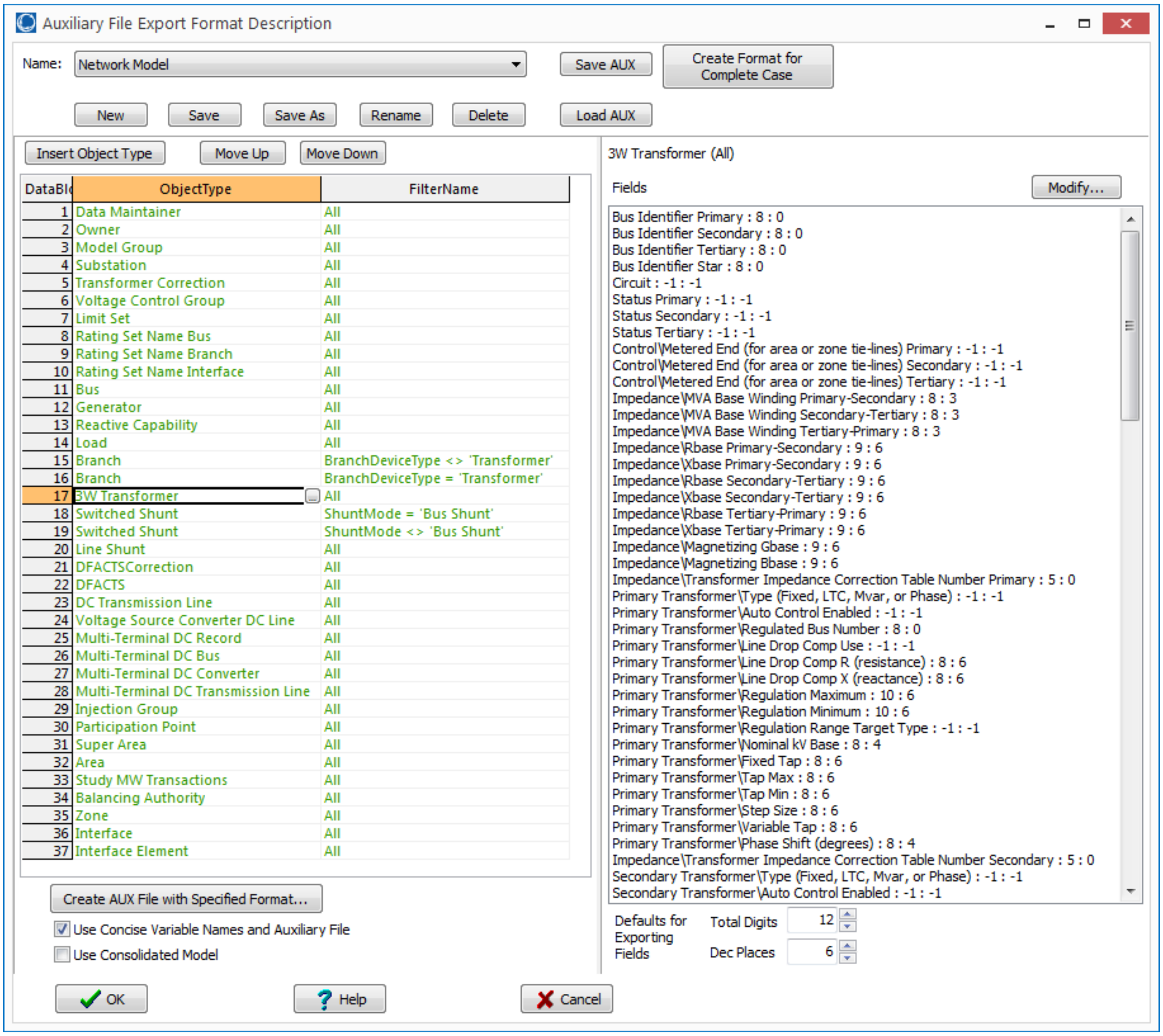
To create an Auxiliary File Export Format Description, go to the Case Information ribbon tab of Simulator and choose AUX Export Format Desc as shown in the following figure.



This will open a dialog for creating an object that describes what to write to an Auxiliary File. On the dialog that appears you can then choose the button Create Format for Complete Case>Network Model as shown in the following image.

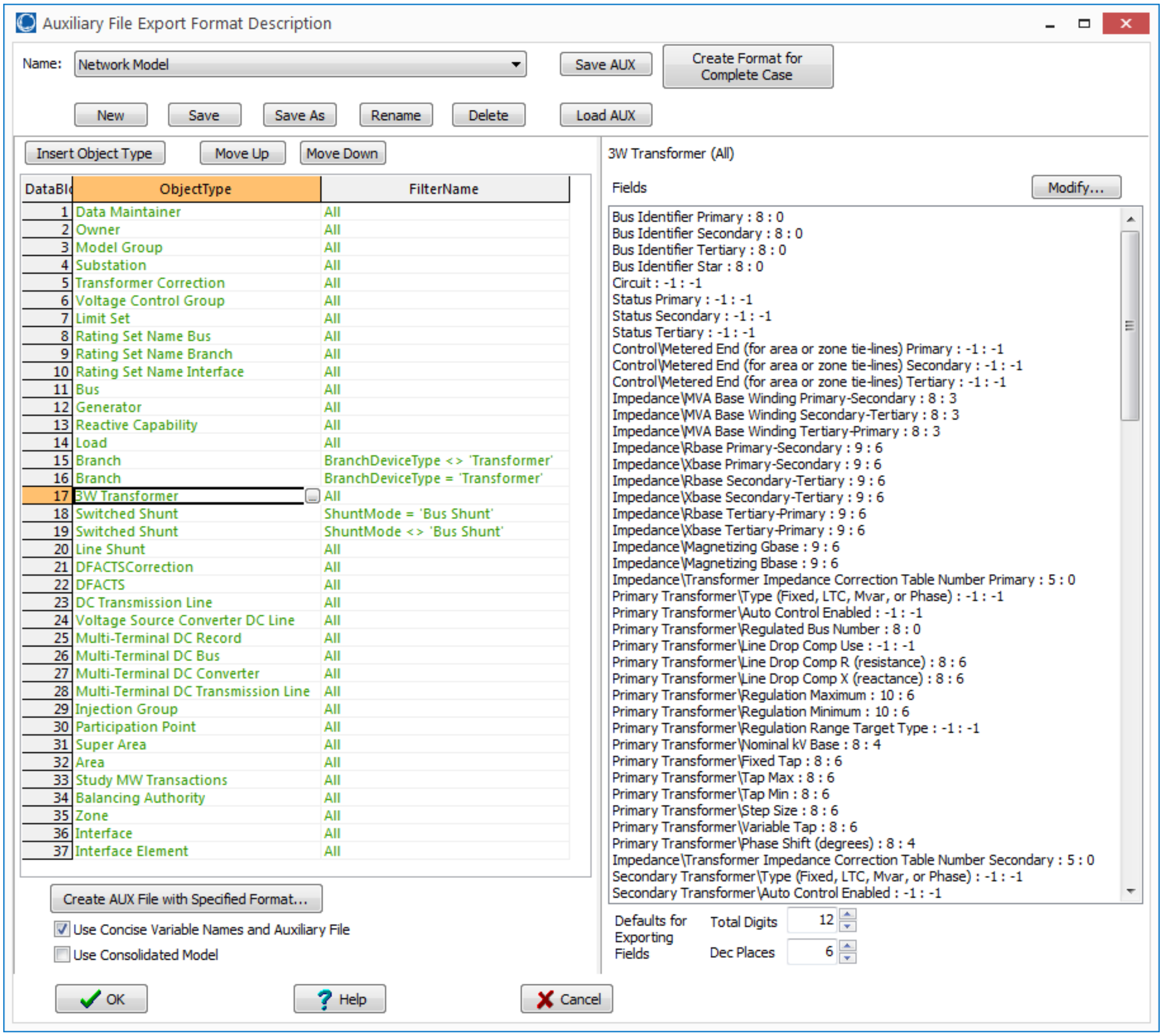


After doing this you will see the following dialog listing all the object types described in this document.



For more help on Auxiliary File Export Format Descriptions, see PowerWorld's help documentation at http://www.powerworld.com/WebHelp/#MainDocumentation_HTML/Auxiliary_ExportFormatDescription.htm

From within this dialog if you check the checkbox **Use Concise Variable Names and Auxiliary File Headers**, you may then click the **Create AUX File with Specified Format button** to write out an auxiliary file that matches the information in this document.


Auxiliary File Export Format Description
Name: Network Model
Save AUX
Create Format for Complete Case
New
Save
Save As
Rename
Delete
Load AUX
Insert Object Type
Move Up
Move Down
DataBl
ObjectType
FilterName
1 Data Maintainer All
2 Owner All
3 Model Group All
4 Substation All
5 Transformer Correction All
6 Voltage Control Group All
7 Limit Set All
8 Rating Set Name Bus All
9 Rating Set Name Branch All
10 Rating Set Name Interface All
11 Bus All
12 Generator All
13 Reactive Capability All
14 Load All
15 Branch BranchDeviceType <> 'Transformer'
16 Branch BranchDeviceType = 'Transformer'
17 3W Transformer All
18 Switched Shunt ShuntMode = 'Bus Shunt'
19 Switched Shunt ShuntMode <> 'Bus Shunt'
20 Line Shunt All
21 DFACTSCorrection All
22 DFACTS All
23 DC Transmission Line All
24 Voltage Source Converter DC Line All
25 Multi-Terminal DC Record All
26 Multi-Terminal DC Bus All
27 Multi-Terminal DC Converter All
28 Multi-Terminal DC Transmission Line All
29 Injection Group All
30 Participation Point All
31 Super Area All
32 Area All
33 Study MW Transactions All
34 Balancing Authority All
35 Zone All
36 Interface All
37 Interface Element All
Create AUX File with Specified Format...
Use Concise Variable Names and Auxiliary File
Use Consolidated Model
3W Transformer (All)
Fields
Modify...
Bus Identifier Primary : 8 : 0
Bus Identifier Secondary : 8 : 0
Bus Identifier Tertiary : 8 : 0
Bus Identifier Star : 8 : 0
Circuit : -1 : -1
Status Primary : -1 : -1
Status Secondary : -1 : -1
Status Tertiary : -1 : -1
Control\Metered End (for area or zone tie-lines) Primary : -1 : -1
Control\Metered End (for area or zone tie-lines) Secondary : -1 : -1
Control\Metered End (for area or zone tie-lines) Tertiary : -1 : -1
Impedance\MVA Base Winding Primary-Secondary : 8 : 3
Impedance\MVA Base Winding Secondary-Tertiary : 8 : 3
Impedance\MVA Base Winding Tertiary-Primary : 8 : 3
Impedance\Rbase Primary-Secondary : 9 : 6
Impedance\Xbase Primary-Secondary : 9 : 6
Impedance\Rbase Secondary-Tertiary : 9 : 6
Impedance\Xbase Secondary-Tertiary : 9 : 6
Impedance\Rbase Tertiary-Primary : 9 : 6
Impedance\Xbase Tertiary-Primary : 9 : 6
Impedance\Magnetizing Gbase : 9 : 6
Impedance\Magnetizing Bbase : 9 : 6
Impedance\Transformer Impedance Correction Table Number Primary : 5 : 0
Primary Transformer\Type (Fixed, LTC, Mvar, or Phase) : -1 : -1
Primary Transformer\Auto Control Enabled : -1 : -1
Primary Transformer\Regulated Bus Number : 8 : 0
Primary Transformer\Line Drop Comp Use : -1 : -1
Primary Transformer\Line Drop Comp R (resistance) : 8 : 6
Primary Transformer\Line Drop Comp X (reactance) : 8 : 6
Primary Transformer\Regulation Maximum : 10 : 6
Primary Transformer\Regulation Minimum : 10 : 6
Primary Transformer\Regulation Range Target Type : -1 : -1
Primary Transformer\Nominal kV Base : 8 : 4
Primary Transformer\Fixed Tap : 8 : 6
Primary Transformer\Tap Max : 8 : 6
Primary Transformer\Tap Min : 8 : 6
Primary Transformer\Step Size : 8 : 6
Primary Transformer\Variable Tap : 8 : 6
Primary Transformer\Phase Shift (degrees) : 8 : 4
Impedance\Transformer Impedance Correction Table Number Secondary : 5 : 0
Secondary Transformer\Type (Fixed, LTC, Mvar, or Phase) : -1 : -1
Secondary Transformer\Auto Control Enabled : -1 : -1
Defaults for Exporting Fields
Total Digits 12
Dec Places 6
OK
Help
Cancel