Record Specification and File Format for Specifying Contingency Definitions and Remedial Actions Schemes

Matt Davis
matt@powerworld.com
217 384 6330 Ext. 20

PowerWorld Corporation

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

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

Project Summary

- Goal: create format to enable exchange of all data needed for power flow and transient stability simulations
- This is a first step
  - Does not yet include everything needed for transient stability
  - We think it's better to have something to work with as a platform to build on

Key Fields

- How to identify an object in the model

Object Type	Primary Key Fields	Secondary Key Fields
Gen	BusNum ID	BusNameNomkV ID
Bus	Number	NameNomkV
Branch	BusNumFrom BusNumTo Circuit	BusNameNomkVFrom BusNameNomkVTo Circuit
Branch *(Special treatment for interacting with Branches using the EPC format)*	MSBusNumFrom MSBusNumTo Circuit Section	MSBusNameNomkVFrom MSBusNameNomkVTo Circuit Section
Load	BusNum ID	BusNameNomkV ID

Key Fields

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
LineShunt *(Special treatment for interacting with Branches using the EPC format)*	MSBusNumFrom MSBusNumTo MSBusNumLoc Circuit ID Section	MSBusNameNomkVFrom MSBusNameNomkVTo MSBusNumLoc Circuit ID Section
VSCDCLine	Name	*none available*
ModelExpression	Name	*none available*

Primary Keys

- These are the standard identifiers used in Simulator and other software
  - Bus number based keys

General Format	"Objecttype 'key1' 'key2' 'key3'"
Generator	"GEN 23 '12'"
Bus	"BUS 33"
Branch	"BRANCH 23 29 'AB'"
Branch	"BRANCH 23 29 'AB' 4" *(multi-section line branch)*
3WXformer	"3WXFORMER 23 29 66 'AB'"
Area	"AREA 51"
Zone	"ZONE 93"
Substation	"SUBSTATION 37"

Secondary Keys

- Secondary keys are based on the name and nominal kv at a bus level
- These are usually used when there is a problem linking to bus numbers

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'"
Branch	"BRANCH 'Bus 23_138.00' 'Bus 29_138.00' 'AB' 4"

	(multi-section line branch)
Area	"AREA 'Fifty One'"
Zone	"ZONE 'Ninety Three'"
Substation	"SUBSTATION 'Thirty Seven'"

Labels

- The label enclosed in single quotes following the object type. The entire identifier is contained in double quotes
- Labels with a full topology model are probably the best way to identify objects

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

Object Field Definitions

- Subset of PWAUX file format
- Describes the relevant objects an fields needed for defining RAS

Bus Fields (partial list)

Field	Description
Number	Number of the bus
Name	Name of the bus
NameNomkV	Combination of the Name and the kV of the bus separated by an underscore. For example "Johnson_34.5"
Status	Either DISCONNECTED or CONNECTED
Vpu	The per unit voltage magnitude of the bus
Vangle	The angle of the bus in degrees
kV	The voltage magnitude in kilovolts
NomkV	The nominal voltage of the bus in kilovolts

Filtering

- Filters are included as part of the RAS file format
- Filters are the basis for modeling RAS in PowerWorld Simulator

```
FILTER (ObjectType,Name,Logic,Enabled)
{
"Load" "Neg MW" "AND" "NO" "YES"
"Bus" "Neg Load or Generator" "OR" "NO" "YES"
}
CONDITION (ObjectType,Filter,CondNum,ObjectField,ConditionType,Value,OtherValue,Absolute)
{
"Load" "Neg MW" 1 "LoadMW" "<" "0" "" "NO"
"Bus" "Neg Load or Generator" 1 "BusGenMW" "notisblank" "" "" "NO"
"Bus" "Neg Load or Generator" 2 "_UseAnotherFilter" "meets" "<Load>Neg MW" "" "NO"
}
```

Device Filtering

- Device filtering allows you to use a power system object as a filter
  - Injection group can be used as a filter on generators.
    - Only generators in the injection group are returned by the filter
  - Substation can be used as a filter on buses
    - Only buses within a substation are returned as part of the filter
- Device filtering can be used to simplify more complicated filters
  - Instead of creating a filter with “AREA = PSE” as a condition use the PSE area as a filter

Script Sections to Set Defaults

- SetData can be used to set fields on all objects of a certain type

```
SCRIPT
{
SetData(Gen, [UseLineDrop], [“NO”], ALL);
}
```

- Example turns off line drop compensation for all generators in the case

Data Record Structures

- Multiple DATA sections are allowed in a single file (just like a PW AUX file)
- Aggregations might be needed for setting options need for contingency analysis, especially areas and injection groups

```
Area (ObjectID, CTGMakeupGen)
{
  "AREA 40"          0.80
  "AREA 50"          0.15
  "AREA 'PG AND E'"  0.35
  "AREA 'ALBERTA'"   0.05
}
```

Special settings for area make up power (defined in 5.1)

Solution Options

- There are objects/fields to specify the solution options for the power flow as well as the contingencies

RAS Modeling: Supported Objects

- Model Expressions
  - Mathematical Expressions evaluated from one or more system parameters
- Model Condition
  - Used to specify the conditions when a RAS scheme should be used or armed
- Model Filter / Model Filter Condition
  - Used to specify the conditions when a RAS scheme should be used or armed
  - Combination of Model Conditions and other model filters

Model Condition

- Defined using a table of model conditions and model condition conditions (sorry)

```
ModelCondition (Name, Object, FilterObjectType, FilterLogic, EvaluateInRef,
                DisableIfTrueInRef, Memo)
{
"2-11 Line (Open)"     "Branch 2 11 1"        "Branch" "AND" "YES" "YES" "check 2-11"
"3-44 Line (Open)"     "Branch 3 44 1"        "Branch" "AND" "YES" "YES" "check 3-44"
"Interface Bobby High" "Interface 'Bobby'" "Interface" "AND" "NO"  "NO"  "checkbobby"
"Bus Jackie Low"       "Bus 'Jackie_500.00'"     "Bus" "AND" "NO"  "NO"  "checkJackie"
"Sample Field and Exp" "Gen 'Jackie_34.50' '1'"  "Gen" "AND" "NO"  "NO"  "checkJackie"
}
ModelConditionCondition (ModelCondition, CondNum, ObjectField, ConditionType,
                         Value, OtherValue, Absolute)
{
"2-11 Line (Open)"     "1" "DerivedOnline" "=" "Closed" "" "NO"
"3-44 Line (Open)"     "2" "DerivedOnline" "=" "Closed" "" "NO"
"Interface Bobby High" "3" "MW"            ">" 500      "" "NO"
"Interface Bobby High" "4" "Mvar"          ">" 100      "" "NO"
"Bus Jackie Low"       "5" "V"             "<" 0.95     "" "NO"
"Sample Field and Exp" "6" "MW"            ">" "MWMax" "" "NO"
"Sample Field and Exp" "7" "Mvar"          ">" "Expression 'Mvar Lim'" "" "NO"
}
```

Model Filter

- Defines filter with it's name and logic
- The filter conditions are in a separate table of ModelFilterCondition Objects that reference the model filter objects by name

```
ModelFilter (Name,                  Logic,  Memo)
{
"One is Out"                         "OR"  "Either 3-44 or 2-11 go out"
"Bobby Up and Jackie Down" "AND" "test on Bobby and Jackie"
}
ModelFilterCondition (ModelFilter, CondNum, Criteria,                Logic)
{
"One is Out"                          1        "3-44 Line (Open)"     ""
"One is Out"                          2        "2-11 Line (Open)"     ""
"Bobby Up and Jackie Down"            1        "Interface Bobby High" ""
"Bobby Up and Jackie Down"            2        "Bus Jackie Low"       "NOT"
}
```

Contingency Definitions

```
Contingency (Name, Category, Skip, Memo)
{
"L-2_Roughrider-Raven 2&3" "Double" "NO" "My Memo A"
"L-2_Roughrider-Raven 1&2" "Double" "NO" "My Memo A"
"L_Falcon-PatriotC1"       "Single" "NO" "My Memo A"
"T_Falcon-TitanC1"         "Single" "NO" "My Memo A"
}

ContingencyElement (Contingency,Object,Action,Criteria,CriteriaStatus,Comment)
{
"L-2_Roughrider-Raven 2&3" "BRANCH 15 54 2" "OPEN" "" "CHECK" ""
"L-2_Roughrider-Raven 2&3" "BRANCH 15 54 3" "OPEN" "" "CHECK" ""
"L-2_Roughrider-Raven 1&2" "BRANCH 15 54 1" "OPEN" "" "CHECK" ""
"L-2_Roughrider-Raven 1&2" "BRANCH 15 54 2" "OPEN" "" "CHECK" ""
"L_Falcon-PatriotC1"       "BRANCH 10 13 1" "OPEN" "" "CHECK" ""
"T_Falcon-TitanC1"         "BRANCH 10 39 1" "OPEN" "" "CHECK" ""
}

TSContingency (Name, Category, Skip, Memo)
{
"Double Outage" "ReallyBad" "NO" "My Memo W"
"Fault Short"   "NotSoBad"  "NO" "My Memo X"
"Fault Long"    "ReallyBad" "NO" "My Memo Y"
"Fault Near DC" "NotSoBad"  "NO" "My Memo Z"
}

TSContingencyElement (Contingency,Time,Object,Action,Criteria,CriteriaStatus,Comment)
{
"Double Outage" 0.50 "Gen 14931 '1'"  "OPEN"            "" "ALWAYS" ""
"Double Outage" 0.50 "Gen 14932 '1'"  "OPEN"            "" "ALWAYS" ""
"Fault Short"   0.50 "Bus 'ROSS_345'" "FAULT 3PB SOLID" "" "ALWAYS" ""
"Fault Short"   0.60 "Bus 'ROSS_345'" "CLEARFAULT"      "" "ALWAYS" ""
"Fault Long"    0.50 "Bus 'ROSS_345'" "FAULT 3PB SOLID" "" "ALWAYS" ""
"Fault Long "   1.50 "Bus 'ROSS_345'" "CLEARFAULT"      "" "ALWAYS" ""
"Fault Near DC" 0.50 "Bus 'MONA_345'" "FAULT 3PB SOLID" "" "ALWAYS" ""
"Fault Near DC" 0.60 "Bus 'MONA_345'" "CLEARFAULT"      "" "ALWAYS" ""
}
```

Transient Stability Models in Power Flow

- Some transient stability models can be used in the steady state power flow
- TSModelsTrip
  - Models that are listed in TSModelsTrip will model the steady state behavior of the dynamic model
- TSModelsMonitor
  - Models listed in TSModelsMonitor will not cause actions to occur, but the contingency processor will report if the dynamic model would have caused something to happen
- TSMaxModelDelay
  - Any model that would not respond within the time delay (TSMaxModelDelay) will not act in the steady state power flow

Conclusions

- WECC common data format is basically a subset of the PowerWorld aux file format
  - http://www.powerworld.com/files/Auxiliary-File-Format-18.pdf
- This isn't perfect, but it's a start
- Having something that works for 80% of what you want to do while you figure out the other 20% is better than waiting to figure out everything beforehand