Data Maintainer Record

WECC SRWG Meeting

November 4, 2015

Presented by: Jamie Weber, Ph.D.
Director of Software Development
weber@powerworld.com
217 384 6330 ext 13

PowerWorld Corporation

DataMaintainer

- Represents the entity responsible for maintaining the input data for an object
- Purpose in software
  - Provide contact information in the form of an email or phone number of someone to contact regarding an object
  - Provides ability in software to use this to filter output
    - Examples
      - "Write out some input data for Columbia Grid"
      - Write out some input data for Grant County PUD only
    - Makes it easier to piece together portions of a model

DataMaintainer Object

- DataMaintainer object has the following fields
  - **Contact** : Name of Person or Position
  - **Phone**, **Email**, **Company**, **Location**
  - **DataMaintainerAssign** (DataMaintainers can belong to other DataMaintainer – will discuss shortly)
  - Might want a generic email at a company too
    - For example: weccbasecase@bpa.gov

DataMa	Name	DataMaintainer	Contact	Phone	Email	Company	Location
1	WECC		Doug Tucker		dtucker@wecc.biz		Salt Lake City
2	Chelan PUD	Columbia Grid	Zachary Zornes		Zachary.Zornes@chelanpud.org		
3	Snohomish PUD	Columbia Grid	Long Duong		LTDuong@SNOPUD.com		
4	Grant PUD	Columbia Grid	Ken Che		kche@gcpud.org		
5	Puget Sound Energy	Columbia Grid	Eleanor Ewry		eleanor.ewry@pse.com		
6	Seattle City Light	Columbia Grid	Stephanie Lu		Stephanie.Lu@seattle.gov		
7	Bonneville Power Admin	Columbia Grid	Andrew Christensen		alchristensen@bpa.gov		
8	Tacoma Power	Columbia Grid	Khanh Thai		kthai@ci.tacoma.wa.us		
9	Avista Corp.	Columbia Grid	John Gross	509.495.4591	John.Gross@avistacorp.com	Avista	Spokane
10	Columbia Grid	WECC	Jonathan Young		Young@columbiagrid.org		

Assignment of Objects to DataMaintainer

- General concepts for DataMaintainer
  - Assign Some object types can be assigned a specific DataMaintainer
  - Inherit Some objects type may inherit their DataMaintainer from a related object
  - Not All Not all object types can even be considered to belong to a DataMaintainer
  - In end an object has only *one* DataMaintainer
    - However, a DataMaintainer can belong to another DataMaintainer allow you to create "groups"
- Ultimately, each ObjectType has 2 questions
  - Can it be assigned a DataMaintainer?
  - Can it inherit from a related object?
  - Two YES/NO questions give us 4 choices

Assign?/Inherit? = NO/NO

- Objects for which DataMaintainer is not supported at all
- These objects generally represent one of the following 4 ObjectTypes
  - Solution Options
  - Environment Options
  - Objects dynamically created by software
    - Island, AreaTieLine, ZoneTieLine, SuperBus
  - Results of a software calculation
    - ViolationCTG

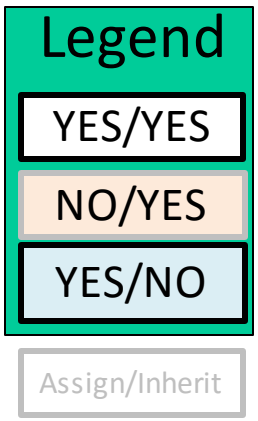
Assign?/Inherit? = YES/NO

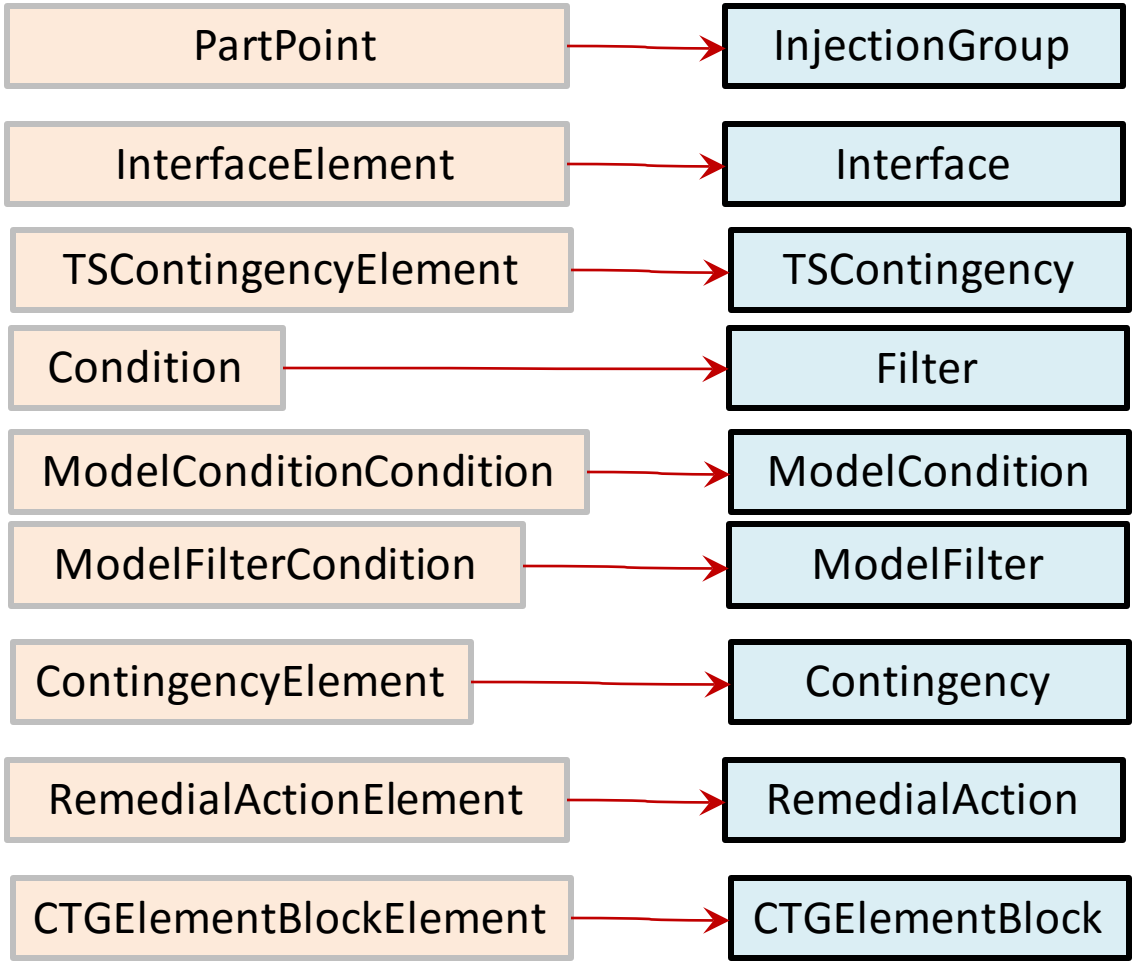
- ObjectTypes which can have a DataMaintainer specified but NEVER inherit
  - Aggregations Objects
    - Area, BalancingAuthority, Direction, InjectionGroup, Interface, Owner, StudyMWTransactions, Substation, SuperArea, VoltageControlGroup, XFCorrection, Zone
  - Transient and Contingency Analysis Objects
    - Contingency, CTGElementBlock, CustomMonitor, GlobalContingencyActionsElement, LimitSet, RemedialAction, ModelCondition, ModelExpression, ModelFilter, ModelStringExpression, TSContingency, TSLimitMonitor, DistributionEquivalent, LoadModelGroup, PlayIn
  - Case Information Objects
    - BGCalculatedField, CustomExpression, CustomExpressionStr, Filter, DataMaintainer

Why allow Inheritance?

- Make it easier for user
  - It is very natural to inherit this from either a Bus or Substation
- For some object types, specifying a DataMaintainer directly won't make any sense
  - InterfaceElement, etc…

Direct Inheritance Only Objects

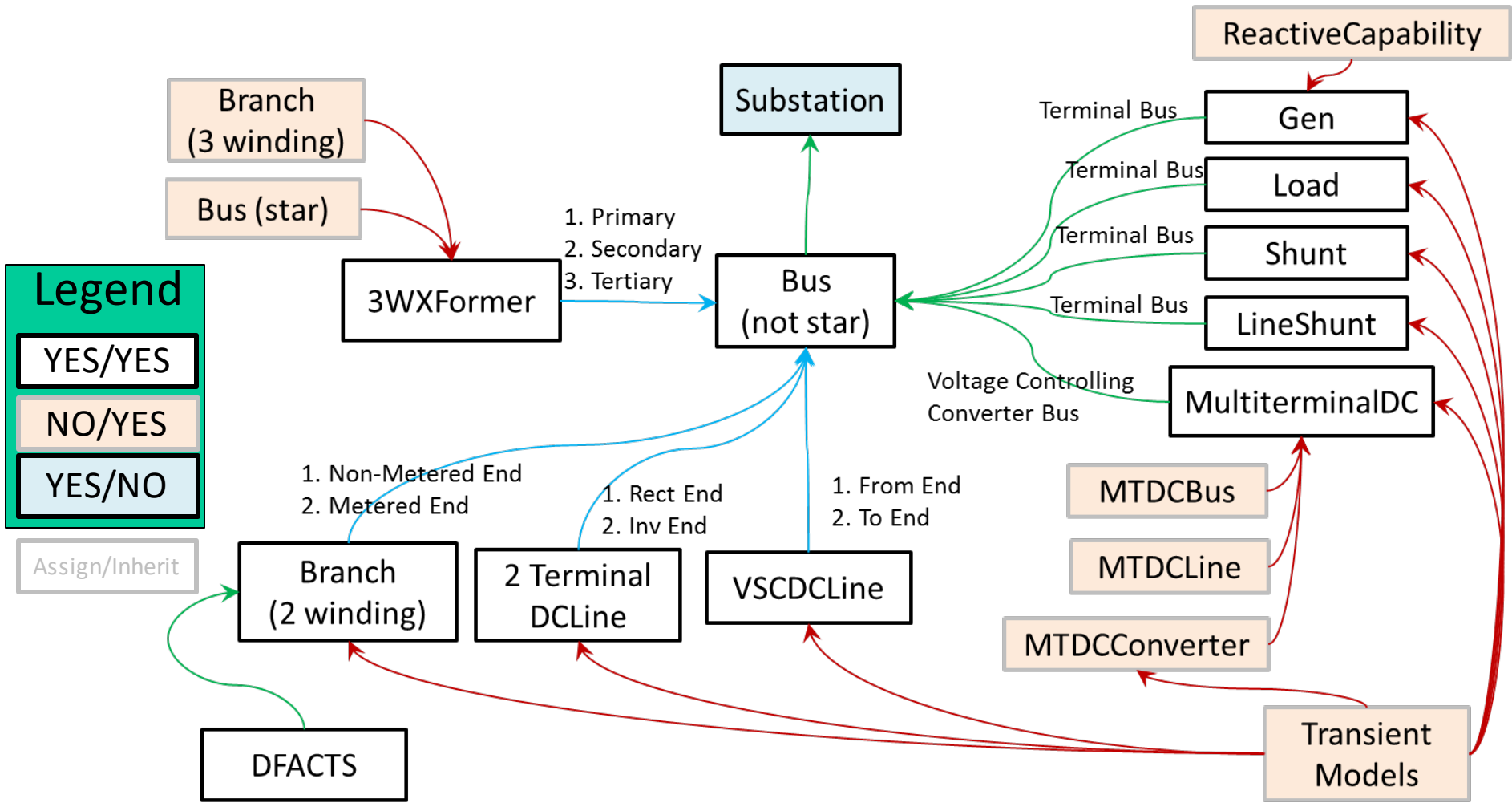




Expected Use of DataMaintainer Inheritance

- Create substations and assign a DataMaintainer to each substation and allow inheritance to work
  - Then just put in exceptions to this rule
  - We expect substations to get populated
    - GIS systems
    - GIC input data requires this
- Alternate: assign a Data Maintainer to each bus and use inheritance

Data Maintain Inheritance from Network Structure



See last slide of presentation for specific details

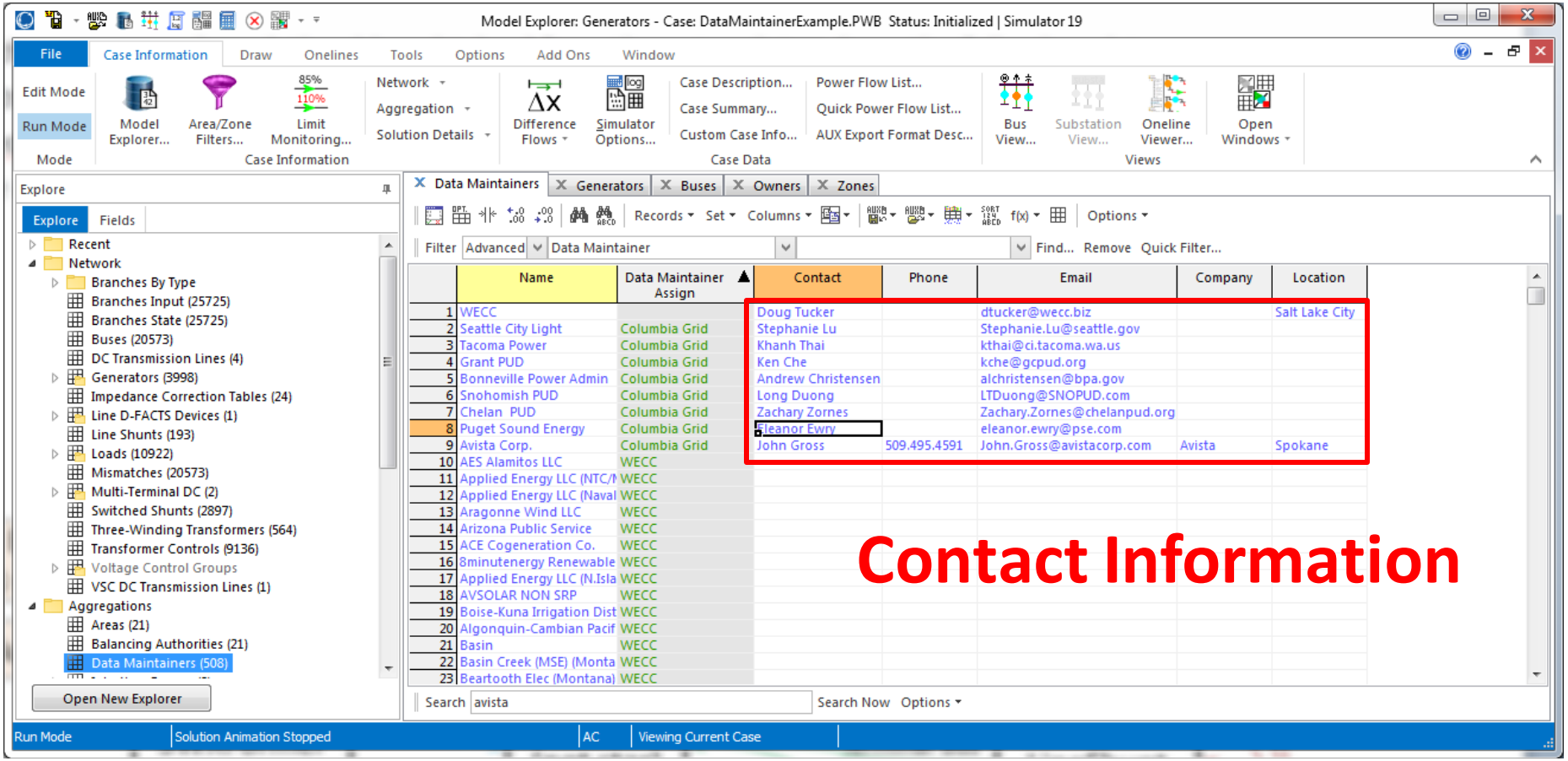
DataMaintainer of a DataMaintainer

- Allow a “Groups of DataMaintainers” which have hierarchy, with WECC at the top of hierarchy

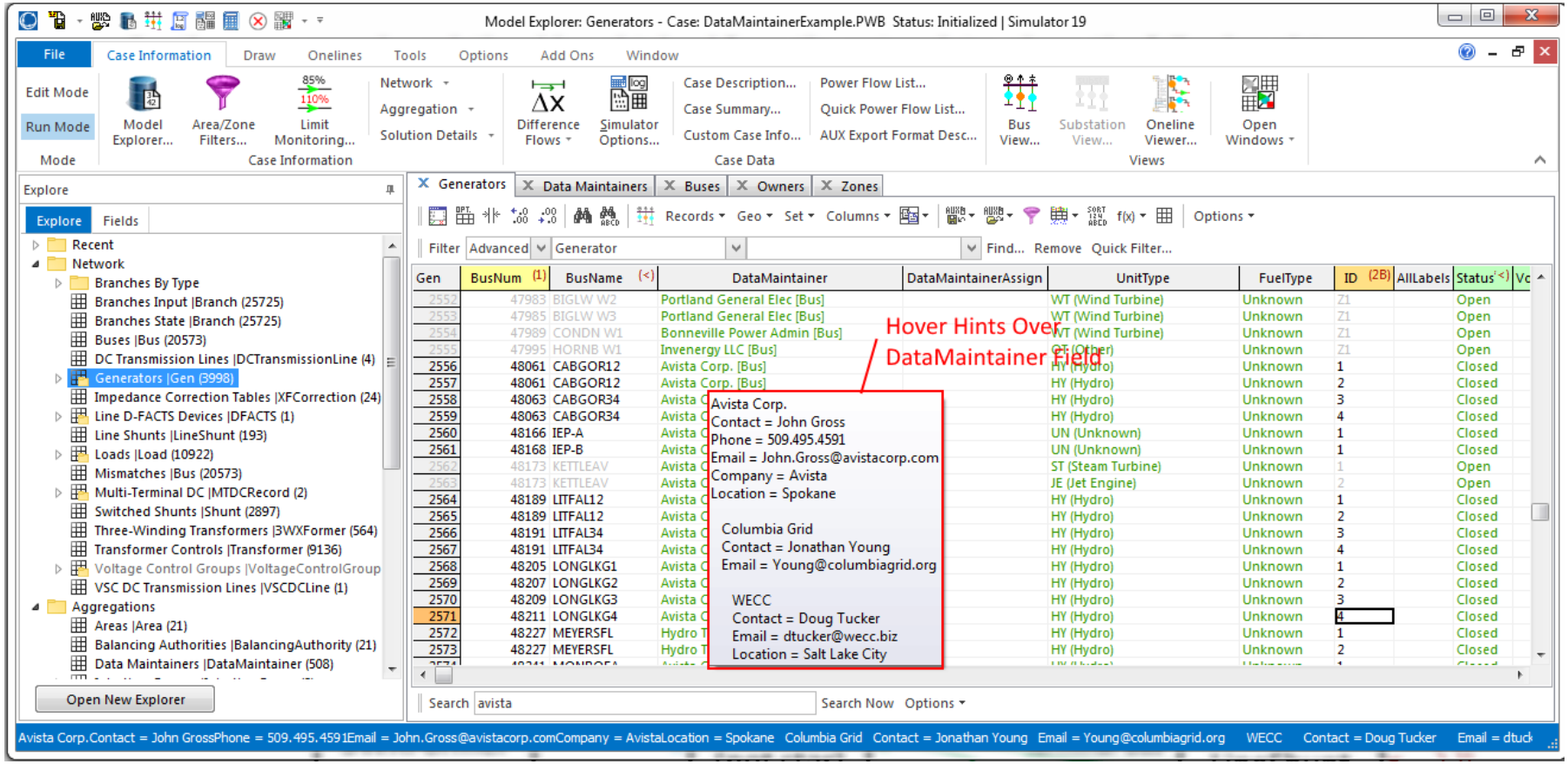
Hypothetical Structure	Representation in an Auxiliary File (Omitting Contact Info)

• WECC
o ColumbiaGrid
▪ Avista
▪ BPA
▪ Chelan
▪ Grant
▪ PSE
▪ Seattle
▪ SnoPUD
▪ Tacoma
o California
▪ SMUD
▪ TID
▪ LADWP
▪ ImperialCA
▪ SCE
▪ SANDIEGO
▪ MEXICO-CFE
o PG&E
o Arizona
o El Paso
o New Mexico
o Nevada
	DataMaintainer (Name, DataMaintainerAssign)
	{
	"WECC" ""
	"Columbia Grid" "WECC"
	"Avista" "Columbia Grid"
	"BPA" "Columbia Grid"
	"Chelan" "Columbia Grid"
	"Grant" "Columbia Grid"
	"PSE" "Columbia Grid"
	"Seattle" "Columbia Grid"
	"SnoPUD" "Columbia Grid"
	"Tacoma" "Columbia Grid"
	"California" "WECC"
	"SMUD" "California"
	"TID" "California"
	"LADWP" "California"
	"ImperialCA" "California"
	"SCE" "California"
	"SANDIEGO" "California"
	"MEXICO-CFE" "California"
	"PG&E" "California"
	"Arizona" "WECC"
	"El Paso" "WECC"
	"New Mexico" "WECC"
	"Nevada" "WECC"
	}

What does GUI look like? Aggregations/DataMaintainer

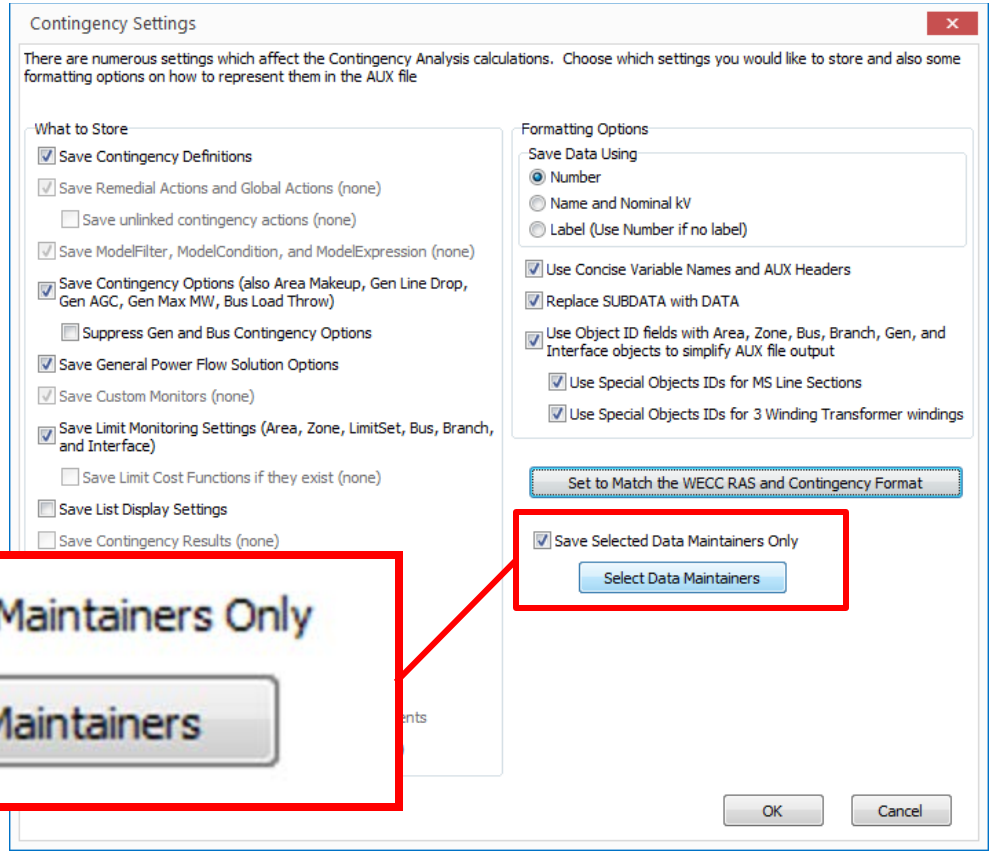


Hover Hints on DataMaintainer field



Saving the WECC RAS and Contingency Format

- "Save Selected Data Maintainers Only"
- This works because all ObjectTypes have a DataMaintainer



Use the DataMaintainer

- If you populate it → features will come
  - Built-In filtering based on DataMaintainers

Inheritance Precedence for Each 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
Branch	YES	1. Non-Metered Bus 2. Non-Metered Bus' Substation
	Except for windings of a 3WXFormer which inherit from the 3WXFormer	
Bus	YES	1. Substation
	Except for internal buses of 3WXFormer which inherits 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. Branches' Non-Metered Bus 2. Branches' Non-Metered Bus' Substation
Gen	YES	1. Bus 2. Bus' Substation
InterfaceElement	NO	Interface
LineShunt	YES	1. Bus 2. Bus' Substation
Load	YES	1. Bus 2. Bus' Substation
ModelConditionCondition	NO	ModelCondition
ModelFilterCondition	NO	ModelFilter

ObjectType	Assign	Inheritance Precedence
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	InjectionGroup
PlayInInfo	NO	PlayIn
PlayInSignal	NO	PlayIn
ReactiveCapability	NO	1. Gen 2. Gen's Bus 3. Gen's Bus' Substation
RemedialActionElement	NO	RemedialAction
Shunt	YES	1. Bus 2. Bus' Substation
StudyMWTransactionsBid	NO	StudyMWTransactions
Transient Stability Model (all)	NO	Object to which the stability model is assigned
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