

Introduction to PowerWorld Simulator: Interface and Common Tools



I12: Auxiliary File Format: SCRIPT Section



PowerWorld
Corporation

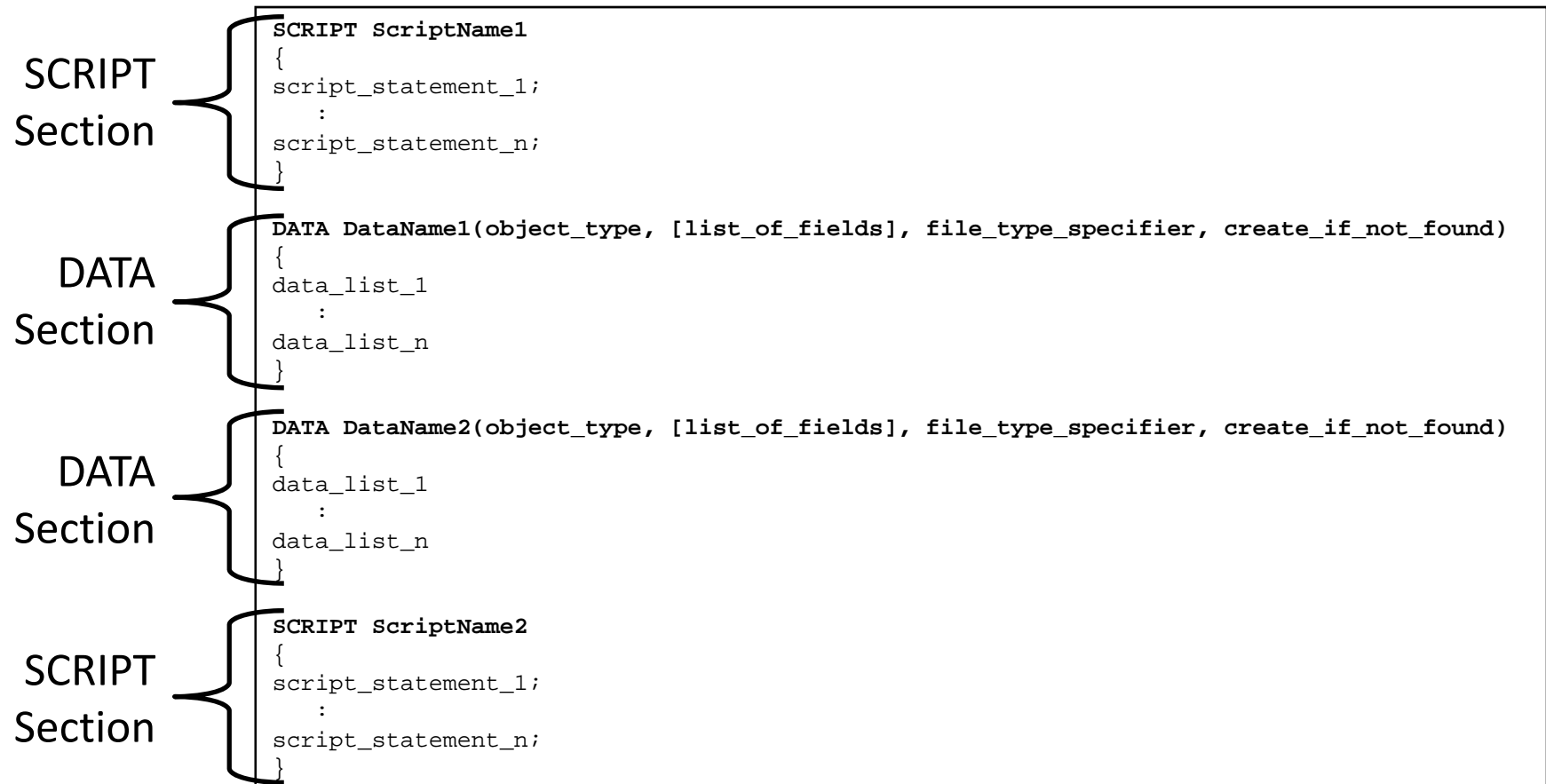
2001 South First Street
Champaign, Illinois 61820
+1 (217) 384.6330

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

Auxiliary File Format Overview



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



Auxiliary File DATA Sections



This was discussed in a previous section

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

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

Auxiliary File SCRIPT Sections



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

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

Script Command Execution Dialog



- To open go to the **Tools** or **Add Ons** ribbon tab and select **Script**
- Manually enter script commands
 - Useful for testing scripts
- Load auxiliary files
 - Validates and applies
- Validate auxiliary files
 - Receive messages in the message log if anything is incorrect in an auxiliary file before applying
- Quick Aux
 - Set up list of auxiliary files that are used frequently
 - Quickly reference a selected auxiliary file
 - Apply group of auxiliary files in a specified order

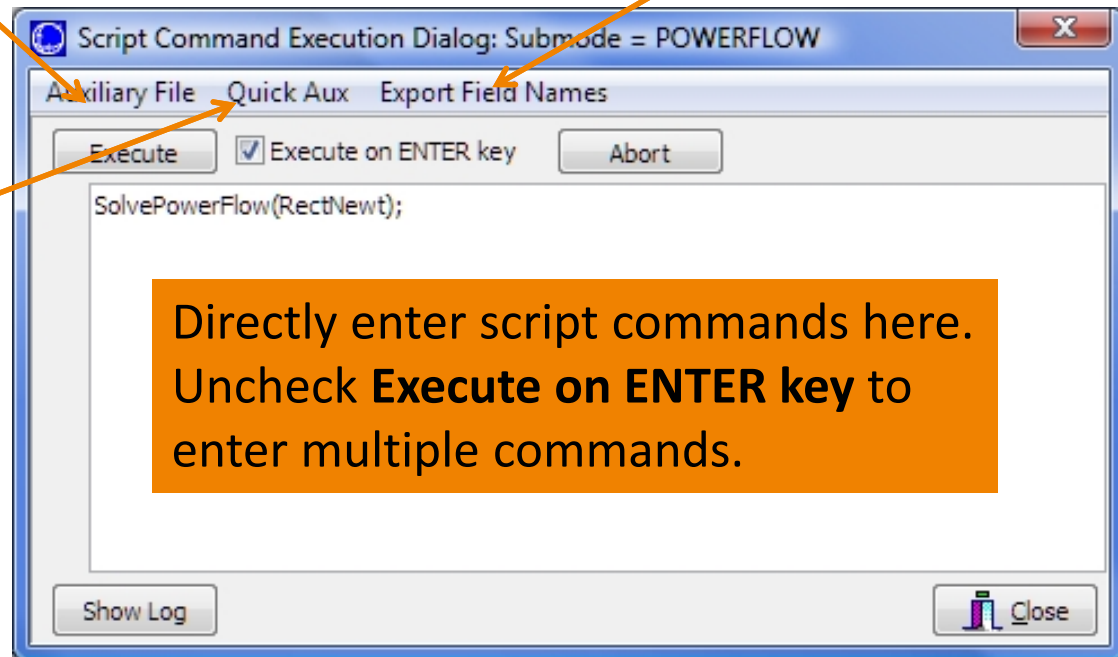
Script Command Execution Dialog



Load or validate auxiliary files

Get list of fields for each object type

Define list of frequently used auxiliary files for easy access



Edit and Run Mode



- The Simulator user-interface has an Edit Mode and a Run Mode
 - Edit Mode: for making changes to system topology (e.g. adding a bus, deleting a transmission line, creating an Equivalent case)
 - Run Mode: for performing calculations and analysis (e.g. solve power flow, run contingency analysis, calculate available transfer capability)
- These modes must also be obeyed in scripting; use *EnterMode* script action to change modes
- A list of script actions follows, but it may not be exhaustive
- These references contain a complete list of script commands and detailed information on the use of each:
 - Simulator program help
 - Window ribbon → **Auxiliary File Format**
 - <http://www.powerworld.com> – Go to Downloads → Simulator Help Files → PowerWorld Simulator Auxiliary File Format

General Actions:

Available in either Run or Edit Mode



```
RenameFile("oldfilename", "newfilename");
CopyFile("oldfilename", "newfilename");
DeleteFile("filename");
LoadAux("filename", CreateIfNotFound);
LoadScript("filename", ScriptName);
LoadData("filename", DataName, CreateIfNotFound);
SelectAll(objecttype, filter);
UnselectAll(objecttype, filter);
Delete(objecttype, filter);
DeleteIncludingContents(objecttype, filter);
SaveData("filename", filetype, objecttype, [fieldlist],
         [subdatalist], filter, [SortFieldList]);
SaveDataWithExtra("filename", filetype, objecttype,
                 [fieldlist], [subdatalist], filter, [SortFieldList],
                 [Header_List], [Header_Value_List]);
SaveDataUsingExportFormat("filename", filetype, "FormatName",
                          ModelToUse);
SetData(objecttype, [fieldlist], [valuelist], filter);
CreateData(objecttype, [fieldlist], [valuelist]);
```


General Actions:

Available in either Run or Edit Mode



```
Scale(scaletype, basedon, [parameters], scalemarker);
SaveObjectFields("filename", objecttype, [fieldlist]);
WriteTextToFile("filename", "text...");
SetCurrentDirectory("fielddirectory", CreateIfNotFound);
OpenOnline("filename", "view", FullScreen, ShowFull);
SaveOnline("filename", "OnlineName", savefiletype);
CloseOnline("OnlineName");
LoadAXD("filename", "OnlineName", CreateIfNotFound);
ExportOnlineAsShapeFile("filename", "OnlineName",
    "shapefileDOC", UseLonLat);
ExitProgram;
NewCase;
OpenCase("filename", OpenFileType, [LoadTransactions,
    StarBus,MSLine, VarLimDead, PostCTGAGC]);
SaveCase("filename", SaveFileType);
EnterMode(mode);
LogClear;
LogSave("filename", AppendFile);
```

General Actions:

Available in either Run or Edit Mode



```
LogAdd("string...");
LogAddDateTime("label", includedate, includetime,
               includemilliseconds);
CaseDescriptionClear;
CaseDescriptionSet("text", Append);
SaveYbusInMatlabFormat("filename", IncludeVoltages);
SaveJacobian("JacFileName", "JIDFileName", FileType, JacForm)
SetParticipationFactors(Method, ConstantValue, Object);
    GenForceLDC_RCC(filter);
CalculateRXBGFromLengthConfigCondType(filter);
DirectionsAutoInsert(Source, Sink, DeleteExisting,
                     UseDisplayFilters, Start, Increment);
DetermineShortestPath([start], [end], BranchDistMeas, BranchFilter,
                      Filename);
DeterminePathDistance([start], BranchDistMeas, BranchFilter,
                      BusField);
ChangeSystemMVABase(NewBase);
```

General Actions:

Available in either Run or Edit Mode



```
InjectionGroupsAutoInsert;  
CloseOonline("OonlineName");  
SaveOonline("filename", "OonlineName", SaveFileType);  
ExportOonlineAsShapeFile("filename", "OonlineName",  
    "ShapeFileExportDescriptionName", UseLonLat);  
LoadAXD("filename", "OonlineName", CreateIfNotFound);
```

Available only with SimAuto or Retriever

```
OpenOonline("filename", "view", FullScreen, ShowFull);
```

Edit Mode Actions



```
Equivalence;  
DeleteExternalSystem;  
SaveExternalSystem("Filename", SaveFileType, WithTies);  
Move([elementA], [destination parameters],HowMuch);  
Combine([elementA], [elementB]);  
SplitBus([element], NewBusNumber, InsertBusTieLine, LineOpen);  
MergeBuses([element],Filter);  
MergeLineTerminals(Filter);  
TapTransmissionLine([element], PosAlongLine, NewBusNumber,  
    ShuntModel, TreatAsMSLine);  
InterfacesAutoInsert(Type, DeleteExisting,  
    UseFilters,"Prefix",Limits);  
AppendCase("filename",OpenFileType);  
AppendCase("filename",OpenFileType,[StarBus]);  
AppendCase("filename",OpenFileType,[MSLine, VarLimDead,  
    PostCTGAGC]);  
Renummer3WXFormerStarBuses("filename");  
RenummerMSLineDummyBuses("filename");  
AutoInsertTieLineTransactions;
```

Run Mode Actions



```
Animate(DoAnimate);
CalculatePTDF([transactor seller], [transactor buyer],
    LinearMethod);
CalculatePTDFMultipleDirections(StoreForBranches,
    StoreForInterfaces, LinearMethod);
CalculateLODF([BRANCH nearbusnum farbusnum ckt], LinearMethod);
CalculateLODFMatrix(WhichOnes, filterProcess, filterMonitor,
    MonitorOnlyClosed, LinearMethod);
CalculateTLR([flow element], direction, [transactor], LinearMethod);
CalculateTLRMultipleElement(TypeElement, WhichElement, direction,
    [transactor], LinearMethod);
CalculateVoltSense([BUS num]);
CalculateFlowSense([flow element], FlowType);
CalculateLossSense (FunctionType);
CalculateVoltToTransferSense([transactor seller], [transactor
    buyer], TransferType, TurnOffAVR);
CalculateVoltSelfSense(filter);
SetSensitivitiesAtOutOfServiceToClosest;
ZeroOutMismatches;
```

Run Mode Actions



```
DoCTGAction([contingency action]);
SolvePowerFlow(SolMethod, "filename1", "filename2",
    CreateIfNotFound1, CreateIfNotFound2);
ResetToFlatStart(FlatVoltagesAngles, ShuntsToMax, LTCsToMiddle,
    PSAnglesToMiddle);
SolvePrimalLP("filename1", "filename2", CreateIfNotFound1,
    CreateIfNotFound2);
InitializeLP("filename1", "filename2", CreateIfNotFound1,
    CreateIfNotFound2);
SolveSinglePrimalLPOuterLoop("filename1", "filename2",
    CreateIfNotFound1, CreateIfNotFound2);
SolveFullSCOPF(BCMethod, "filename1", "filename2",
    CreateIfNotFound1, CreateIfNotFound2);
OPFWriteResultsAndOptions("filename");
DiffFlowSetAsBase;
DiffFlowClearBase;
DiffFlowMode(diffmode);
CTGSolveAll(DoDistributed);
CTGSolve("ContingencyName");
```

Run Mode Actions



```
CTGSetAsReference;  
CTGRestoreReference;  
CTGProduceReport("filename");  
CTGWriteResultsAndOptions("filename", [opt1,...,opt9], KeyField,  
    UseDATASection);  
CTGAutoInsert;  
CTGConvertAllToDeviceCTG;  
CTGCalculateOTDF([transactor seller], [transactor buyer],  
    LinearMethod);  
CTGWriteFilePTI("filename", BusFormat, TruncateCTGLabels);  
ATCDetermine([transactor seller], [transactor buyer], ApplyTransfer,  
    DoDistributed);  
ATCRestoreInitialState;  
ATCIncreaseTransferBy(amount);  
ATCTakeMeToScenario(RL, G, I);  
ATCDetermineATCFor(RL, G, I, ApplyTransfer);  
ATCWriteResultsAndOptions("filename", AppendFile);  
ATCWriteToExcel("worksheetname");  
ATCWriteToText("filename", filetype);
```

Run Mode Actions



```
Fault([Bus num, faulttype, R, X]);
Fault([BRANCH nearbusnum farbusnum ckt], faultlocation, faulttype,
      R, X]);
PVSetSourceAndSink([elementSource], [elementSink]);
PVRun;
PVRun([element source], [element sink]);
PVClear;
PVStartOver;
PVDestroy;
PVWriteResultsAndOptions("filename");
RefineModel(objecttype, filter, Action, Tolerance);
PVQVTrackSingleBusPerSuperBus;
QVRun("filename", InErrorMakeBaseSolvable);
QVWriteResultsAndOptions("filename");
RefineModel(objecttype, filter, Action, Tolerance);
QVSelectSingleBusPerSuperBus;
SaveConsolidatedCase("filename", filetype, [BusFormat,
      TruncateCtgLabels]);
```


Run Mode Actions



```
TSSolveAll;  
TSSolve("ContingencyName", [StartTime, StopTime, StepSize]);  
TSSolveOptions("FileName", [SaveDynamicModel, SaveStabilityOptions,  
    SaveStabilityEvents, SaveResultsEvents, SavePlotDefinitions],  
    KeyField);  
TSSaveTwoBusEquivalent ("AuxFileName", [BUS]);  
TSCalculateSMIBEigenValues;  
TSLoadGE("FileName", GENCCYN);  
TSLoadPTI("FileName", "MCREfilename", "MTRLOfilename",  
    "GNETfilename", "BASEGENfilename");  
TSLoadBPA("FileName");
```

Display Auxiliary (axd) Script Actions



```
AutoInsertBorders;  
AutoInsertBuses(LocationSource, MapProjection, AutoInsertBranches,  
    InsertIfNotAlreadyShown, "filename", FileCoordinates);  
AutoInsertLoads(MinkV, InsertTextFields);  
AutoInsertSwitchedShunts(MinkV, InsertTextFields);  
AutoInsertLines(MinkV, InsertTextFields, InsertEquivObjects,  
    InsertZBRPieCharts, InsertMSLines, ZBRImpedance,  
    NoStubsZBRs, SingleCBZRs);  
AutoInsertLineFlowObjects(MinkV, InsertOnlyIfNotAlreadyShown,  
    LineLocation, Size, FieldDigits, FieldDecimals,  
    TextPosition, ShowMW, ShowMvar, ShowMVA, ShowUnits,  
    ShowComplex);  
AutoInsertSubStations(LocationSource, MapProjection,  
    AutoInsertBranches, InsertIfNotAlreadyShown, "filename",  
    FileCoordinates);  
AutoInsertLineFlowPieCharts(MinkV, InsertOnlyIfNotAlreadyShown,  
    InsertMSLines, Size);  
AutoInsertInterfaces(InsertPieCharts, PieChartSize)  
ResetStubLocations(ZBRImpedance, NoStubsZBRs)
```

Display Auxiliary (axd) Script Actions



- Simulator also supports these General Actions in the axd format

```
ExitProgram  
LoadScript  
LoadData  
SelectAll  
UnselectAll  
SetData  
SaveData  
SaveDataWithExtra  
CreateData  
DeleteFile  
RenameFile  
CopyFile  
SetCurrentDirectory  
SaveObjectFields
```

Auxiliary Files and Difference Cases



- The Difference Case Tool may be used to build an auxiliary file that shows the topology difference between 2 cases
- Uses
 - Document topology changes in cases
 - Modify a case with Simulator GUI and capture changes in auxiliary file; the auxiliary file can be run on a different copy of the Base Case to produce the Changed Case

Auxiliary Files and Difference Cases



- Example
 - Open “Case A” and choose **Tools → Difference Flows → Set Present Case as Base Case**
 - Open “Case B” and choose **Tools → Difference Flows → Present Topological Differences From Base Case**

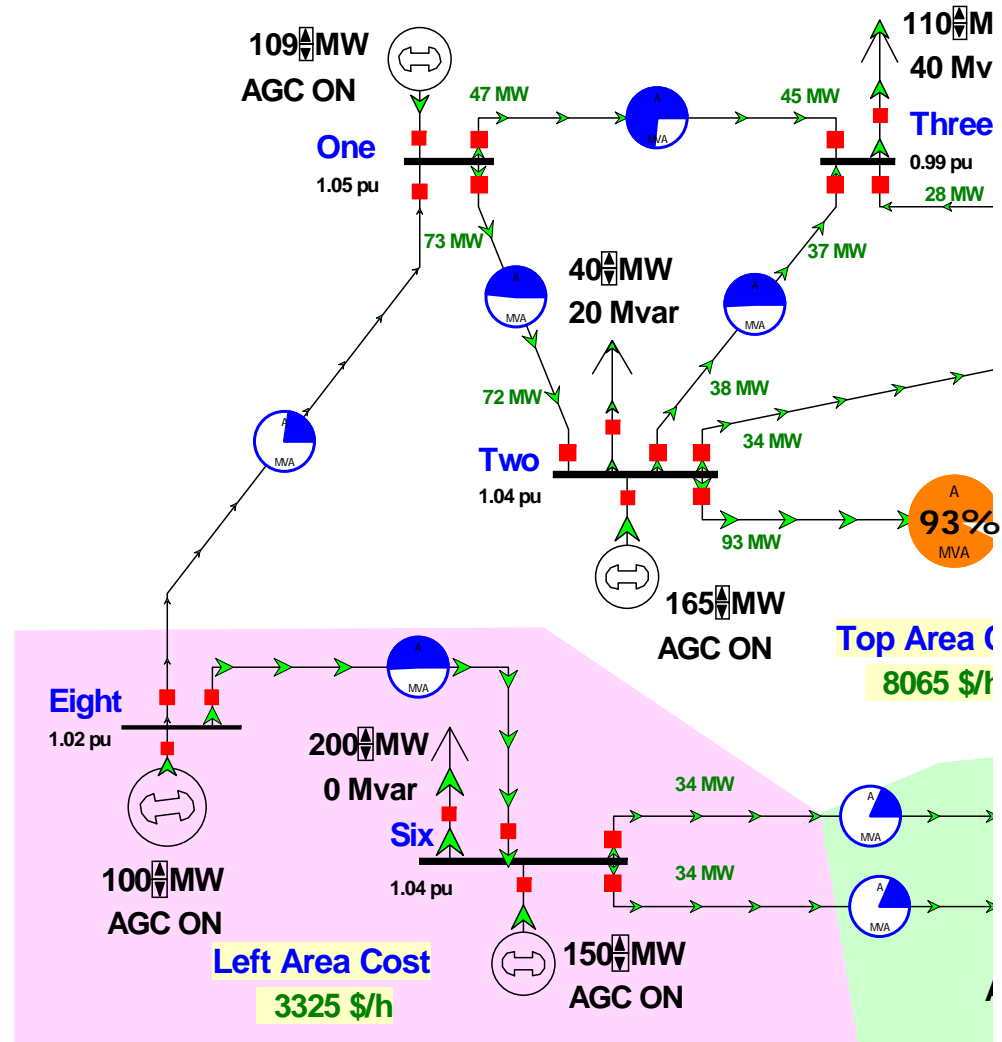
Use Difference Cases



- First, save new items (*Only Elements Added*) to auxiliary file
- Then save removed items (*Only Elements Removed*) items: append to the same auxiliary file if desired
- Add a script statement at the beginning of the newly created file to switch to edit mode so that objects may be added and deleted

Example

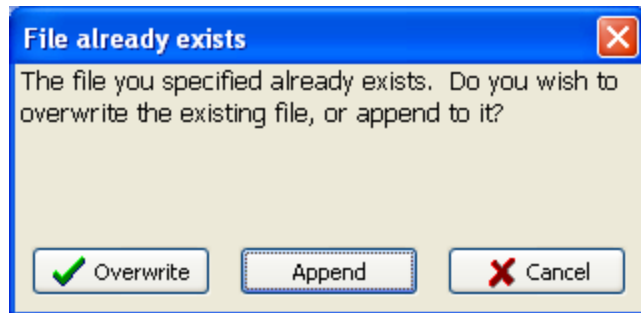
- Open B7FLAT.pwb
- Set Present Case as Base Case
- Add new Bus 8, new Gen at Bus 8, and new Branches connecting Bus 8 to Bus 1 and Bus 6
- Remove Branch between Bus 2 and Bus 6



Present Topological Differences from Base Case



Save New items to Aux File,
then Save Removed items
to same Aux File (append)



Present Case Topological Differences from the Base Case

Summary Elements Added Elements Removed Elements In Both Create Bus Swap List

Below is a summary of the comparison between the present case: G:\pw\version.130\Training\Training Cases\M09_OPF Automation\B8FLAT.pwb and the base case: B7FLAT.pwb saved from the Difference Flows Dialog.

Element Type	New	Removed	Both
Bus	1	0	7
Load	0	0	6
Switched Shunt	0	0	0
Generator	1	0	5
Branch	2	1	10
DC Line	0	0	0
Area	0	0	3
Zone	0	0	1
SuperArea	0	0	0
Transformer	0	0	0
Interface	0	0	3
Injection Group	0	0	0
Substation	0	0	0
Nomogram	0	0	0
MT DC Record	0	0	0

Assume base case Areas/Zones which are not in present case meet the Area/Zone Filters

Save and Send Option
Only Elements Added

Use Area/Zone Filters when saving to Auxiliary File

Send To Excel Save to Text File
Save To Aux File Load Aux File

Close

Edit the Resulting Aux File



- Optionally add SCRIPT statement to change to EDIT mode to ensure that objects may be created and removed
- Then re-open B7FLAT.pwb and load the aux file

```
SCRIPT
{
EnterMode(EDIT);
}

//-----
// THE FOLLOWING SECTION CONTAINS THE ELEMENTS ADDED IN PRESENT CASE
//-----
DATA (BUS,
[BusNum, BusName, BusNomVolt, BusPUVolt, BusAngle, BusG:1, BusB:1, AreaNum, ZoneNum,
SubNum, BusSlack, BusStatus, BusNum:1, BusName_NomVolt:1, BusLongName, BusKVVolt,
.
.
.
```

Save New and Modified Display Objects to AXD File



- Select objects on One-line
- Onelines → List Display → Only Selected Display Objects...

Right Click →
Save As → Display
Auxiliary File...

The screenshot shows a one-line diagram with various components labeled: 109 MW AGC ON, 47 MW, 73 MW, 72 MW, 1.04 pu, 32 MW, 200 MW 0 Mvar, 150 MW AGC ON, 34 MW, 100 MW AGC ON, 1.02 pu, and Left Area Cost 3325 \$/h. The 'Display Explorer: All Objects' window is open, showing a tree view of objects and a table of object details.

Type	X/Longitude Location	Y/Latitude Location	Layer Name	Layer Shown	Selectable Mode
1 DisplayBus	-10.000	25.000	Default Layer	YES	YES
2 DisplayBusField	-15.000	28.000	Default Layer	YES	YES
3 DisplayTransmissionLine	-7.000	25.000	Default Layer	YES	YES
4 DisplayCircuitBreaker	-7.000	27.000	Default Layer	YES	YES
5 DisplayTransmissionLine	-4.000	25.000	Default Layer	YES	YES
6 DisplayCircuitBreaker	-4.000	27.000	Default Layer	YES	YES
7 DisplayGen	-7.000	25.000	Default Layer	YES	YES
8 DisplayBusField	-15.000	25.000	Default Layer	YES	YES
9 DisplayGenField	-12.000	16.000	Default Layer	YES	YES
10 DisplayGenField	-11.000	13.000	Default Layer	YES	YES
11 DisplayBranchPie	9.913	29.000	Default Layer	YES	YES
12 DisplayBranchPie	0.889	44.209	Default Layer	YES	YES
13 DisplayCircuitBreaker	10.000	61.000	Default Layer	YES	YES
14 DisplayCircuitBreaker	16.000	18.000	Default Layer	YES	YES
15 DisplayBus	10.000	16.000	Default Layer	YES	YES
16 DisplayBusField	8.000	15.000	Default Layer	YES	YES
17 DisplayBusField	8.000	19.000	Default Layer	YES	YES
18 DisplayLoad	12.000	16.000	Default Layer	YES	YES
19 DisplayLoadField	2.000	22.000	Default Layer	YES	YES
20 DisplayLoadField	2.000	25.000	Default Layer	YES	YES

Blank Page

Blank Page