Sub FindGens()

'

' FindGens Macro

' Returns the list of generators based on the "To Station" & "To Feeder", and the "From Station" & "From Feeder(s)"

'

' Keyboard Shortcut: Ctrl+g

'

Dim rngTargetStation As Range

Dim rngTargetFeeder As Range

Dim lngRowCounter, CapacityRowCounter As Long

Dim Tfeeder, Ffeeder1, Ffeeder2, Ffeeder3, Hyplink, Hyplinks As String

Dim Tstation, Fstation1, Fstation2, Fstation3 As String

Dim NC, ProjectName, ProjectType, Size, CA, TSstation, TSfeeder, DSstation, DSfeeder, ProjectID, DCAlink As String

Dim dropdown As String

dropdown = "=Lists!C2:C3" 'yes/no dropdown

Worksheets("Capacity").Activate

Range("B5").Select

Tstation = ActiveCell.Value

Range("C5").Select

Tfeeder = ActiveCell.Value

Range("D5").Select

Fstation1 = ActiveCell.Value

Range("E5").Select

Ffeeder1 = ActiveCell.Value

Range("D6").Select

Fstation2 = ActiveCell.Value

Range("E6").Select

Ffeeder2 = ActiveCell.Value

Range("D7").Select

Fstation3 = ActiveCell.Value

Range("E7").Select

Ffeeder3 = ActiveCell.Value

Application.ScreenUpdating = False ' added to hide updating (flashing related to copying/pasting from one worksheet into another) from showing up.

Range("A28:P200").Select

Selection.ClearContents

Range("F25").Select

Worksheets("Infobase Distributed Generators").Activate

lngRowCounter = 4

CapacityRowCounter = 28

Set rngTargetStation = Range("R" & lngRowCounter)

'this removes the first characters in the DCA link which would otherwise prevent the link from working

'after the macro is run once, these instructions won't do anything

Worksheets("Infobase Distributed Generators").Columns("U").Replace \_

What:="<a target=\_blank href=", Replacement:="", \_

SearchOrder:=xlByColumns, MatchCase:=True

'this limits the characters in the DCA link, removing the last characters which would otherwise prevent the link from working

'after the macro is run once, these instructions won't do anything

Do While Not IsEmpty(rngTargetStation.Value)

Set rngTargetStation = Range("R" & lngRowCounter)

Hyplink = Range("U" & lngRowCounter).Value

Hyplinks = Left(Hyplink, 65)

Range("U" & lngRowCounter).Value = Hyplinks

lngRowCounter = lngRowCounter + 1

Loop

lngRowCounter = 4

If Tstation <> "" And Tfeeder <> "" Then

Set rngTargetStation = Range("R" & lngRowCounter)

Do While Not IsEmpty(rngTargetStation.Value)

Set rngTargetStation = Range("R" & lngRowCounter)

Set rngTargetFeeder = Range("O" & lngRowCounter)

If InStr(1, rngTargetStation.Value, Tstation) And InStr(1, rngTargetFeeder.Value, Tfeeder) <> 0 Then 'Checks for Projects that match the "To Station" and "To Feeder"

NC = Range("A" & lngRowCounter).Value 'If the station and feeder match, copy the generator information

ProjectName = Range("B" & lngRowCounter).Value

ProjectType = Range("H" & lngRowCounter).Value

Size = Range("J" & lngRowCounter).Value

CA = Range("K" & lngRowCounter).Value

TSstation = Range("R" & lngRowCounter).Value

TSfeeder = Range("O" & lngRowCounter).Value

DSstation = Range("N" & lngRowCounter).Value

DSfeeder = Range("M" & lngRowCounter).Value

DCAlink = Range("U" & lngRowCounter).Value

ProjectID = Range("G" & lngRowCounter).Value

Worksheets("Capacity").Activate 'Paste the Generator information into the Capacity worksheet

Range("A" & CapacityRowCounter).Value = NC

Range("B" & CapacityRowCounter).Value = ProjectName

Range("C" & CapacityRowCounter).Value = ProjectID

Range("H" & CapacityRowCounter).Value = ProjectType

Range("I" & CapacityRowCounter).Value = Size

Range("K" & CapacityRowCounter).Value = CA

Range("L" & CapacityRowCounter).Value = TSstation

Range("M" & CapacityRowCounter).Value = TSfeeder

Range("N" & CapacityRowCounter).Value = DSstation

Range("O" & CapacityRowCounter).Value = DSfeeder

Range("P" & CapacityRowCounter).Value = DCAlink

With Range("E" & CapacityRowCounter).Validation

.Delete

.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, \_

Operator:=xlBetween, Formula1:=dropdown

.IgnoreBlank = True

.InCellDropdown = True

.InputTitle = ""

.InputMessage = ""

.ErrorMessage = ""

.ShowInput = True

.ShowError = True

End With

Range("E" & CapacityRowCounter).Value = "Yes"

Range("J" & CapacityRowCounter).Value = "Yes"

With Range("F" & CapacityRowCounter).Validation

.Delete

.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, \_

Operator:=xlBetween, Formula1:=dropdown

.IgnoreBlank = True

.InCellDropdown = True

.InputTitle = ""

.InputMessage = ""

.ErrorMessage = ""

.ShowInput = True

.ShowError = True

End With

With Range("G" & CapacityRowCounter).Validation

.Delete

.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, \_

Operator:=xlBetween, Formula1:=dropdown

.IgnoreBlank = True

.InCellDropdown = True

.InputTitle = ""

.InputMessage = ""

.ErrorMessage = ""

.ShowInput = True

.ShowError = True

End With

CapacityRowCounter = CapacityRowCounter + 1

Worksheets("Infobase Distributed Generators").Activate

End If

lngRowCounter = lngRowCounter + 1

Set rngTargetStation = Range("R" & lngRowCounter)

Set rngTargetFeeder = Range("O" & lngRowCounter)

Loop

End If

lngRowCounter = 4

If Fstation1 <> "" And Ffeeder1 <> "" Then

Set rngTargetStation = Range("R" & lngRowCounter)

Do While Not IsEmpty(rngTargetStation.Value)

Set rngTargetStation = Range("R" & lngRowCounter)

Set rngTargetFeeder = Range("O" & lngRowCounter)

If InStr(1, rngTargetStation.Value, Fstation1) And InStr(1, rngTargetFeeder.Value, Ffeeder1) <> 0 Then 'Checks for Projects that match the "From Station 1"

NC = Range("A" & lngRowCounter).Value 'If the station and feeder match, copy the generator information

ProjectName = Range("B" & lngRowCounter).Value

ProjectType = Range("H" & lngRowCounter).Value

Size = Range("J" & lngRowCounter).Value

CA = Range("K" & lngRowCounter).Value

TSstation = Range("R" & lngRowCounter).Value

TSfeeder = Range("O" & lngRowCounter).Value

DSstation = Range("N" & lngRowCounter).Value

DSfeeder = Range("M" & lngRowCounter).Value

DCAlink = Range("U" & lngRowCounter).Value

ProjectID = Range("G" & lngRowCounter).Value

Worksheets("Capacity").Activate 'Paste the Generator information into the Capacity worksheet

Range("A" & CapacityRowCounter).Value = NC

Range("B" & CapacityRowCounter).Value = ProjectName

Range("C" & CapacityRowCounter).Value = ProjectID

Range("H" & CapacityRowCounter).Value = ProjectType

Range("I" & CapacityRowCounter).Value = Size

Range("K" & CapacityRowCounter).Value = CA

Range("L" & CapacityRowCounter).Value = TSstation

Range("M" & CapacityRowCounter).Value = TSfeeder

Range("N" & CapacityRowCounter).Value = DSstation

Range("O" & CapacityRowCounter).Value = DSfeeder

Range("P" & CapacityRowCounter).Value = DCAlink

With Range("E" & CapacityRowCounter).Validation

.Delete

.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, \_

Operator:=xlBetween, Formula1:=dropdown

.IgnoreBlank = True

.InCellDropdown = True

.InputTitle = ""

.InputMessage = ""

.ErrorMessage = ""

.ShowInput = True

.ShowError = True

End With

Range("E" & CapacityRowCounter).Value = "Yes"

Range("J" & CapacityRowCounter).Value = "No"

With Range("F" & CapacityRowCounter).Validation

.Delete

.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, \_

Operator:=xlBetween, Formula1:=dropdown

.IgnoreBlank = True

.InCellDropdown = True

.InputTitle = ""

.InputMessage = ""

.ErrorMessage = ""

.ShowInput = True

.ShowError = True

End With

With Range("G" & CapacityRowCounter).Validation

.Delete

.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, \_

Operator:=xlBetween, Formula1:=dropdown

.IgnoreBlank = True

.InCellDropdown = True

.InputTitle = ""

.InputMessage = ""

.ErrorMessage = ""

.ShowInput = True

.ShowError = True

End With

CapacityRowCounter = CapacityRowCounter + 1

Worksheets("Infobase Distributed Generators").Activate

End If

lngRowCounter = lngRowCounter + 1

Set rngTargetStation = Range("R" & lngRowCounter)

Set rngTargetFeeder = Range("O" & lngRowCounter)

Loop

End If

lngRowCounter = 4

If Fstation2 <> "" And Ffeeder2 <> "" Then

Set rngTargetStation = Range("R" & lngRowCounter)

Do While Not IsEmpty(rngTargetStation.Value)

Set rngTargetStation = Range("R" & lngRowCounter)

Set rngTargetFeeder = Range("O" & lngRowCounter)

If InStr(1, rngTargetStation.Value, Fstation2) And InStr(1, rngTargetFeeder.Value, Ffeeder2) <> 0 Then 'Checks for Projects that match the "From Station 2"

NC = Range("A" & lngRowCounter).Value 'If the station and feeder match, copy the generator information

ProjectName = Range("B" & lngRowCounter).Value

ProjectType = Range("H" & lngRowCounter).Value

Size = Range("J" & lngRowCounter).Value

CA = Range("K" & lngRowCounter).Value

TSstation = Range("R" & lngRowCounter).Value

TSfeeder = Range("O" & lngRowCounter).Value

DSstation = Range("N" & lngRowCounter).Value

DSfeeder = Range("M" & lngRowCounter).Value

DCAlink = Range("U" & lngRowCounter).Value

ProjectID = Range("G" & lngRowCounter).Value

Worksheets("Capacity").Activate 'Paste the Generator information into the Capacity worksheet

Range("A" & CapacityRowCounter).Value = NC

Range("B" & CapacityRowCounter).Value = ProjectName

Range("C" & CapacityRowCounter).Value = ProjectID

Range("H" & CapacityRowCounter).Value = ProjectType

Range("I" & CapacityRowCounter).Value = Size

Range("K" & CapacityRowCounter).Value = CA

Range("L" & CapacityRowCounter).Value = TSstation

Range("M" & CapacityRowCounter).Value = TSfeeder

Range("N" & CapacityRowCounter).Value = DSstation

Range("O" & CapacityRowCounter).Value = DSfeeder

Range("P" & CapacityRowCounter).Value = DCAlink

With Range("E" & CapacityRowCounter).Validation

.Delete

.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, \_

Operator:=xlBetween, Formula1:=dropdown

.IgnoreBlank = True

.InCellDropdown = True

.InputTitle = ""

.InputMessage = ""

.ErrorMessage = ""

.ShowInput = True

.ShowError = True

End With

Range("E" & CapacityRowCounter).Value = "Yes"

Range("J" & CapacityRowCounter).Value = "No"

With Range("F" & CapacityRowCounter).Validation

.Delete

.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, \_

Operator:=xlBetween, Formula1:=dropdown

.IgnoreBlank = True

.InCellDropdown = True

.InputTitle = ""

.InputMessage = ""

.ErrorMessage = ""

.ShowInput = True

.ShowError = True

End With

With Range("G" & CapacityRowCounter).Validation

.Delete

.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, \_

Operator:=xlBetween, Formula1:=dropdown

.IgnoreBlank = True

.InCellDropdown = True

.InputTitle = ""

.InputMessage = ""

.ErrorMessage = ""

.ShowInput = True

.ShowError = True

End With

CapacityRowCounter = CapacityRowCounter + 1

Worksheets("Infobase Distributed Generators").Activate

End If

lngRowCounter = lngRowCounter + 1

Set rngTargetStation = Range("R" & lngRowCounter)

Set rngTargetFeeder = Range("O" & lngRowCounter)

Loop

End If

lngRowCounter = 4

If Fstation3 <> "" And Ffeeder3 <> "" Then

Set rngTargetStation = Range("R" & lngRowCounter)

Do While Not IsEmpty(rngTargetStation.Value)

Set rngTargetStation = Range("R" & lngRowCounter)

Set rngTargetFeeder = Range("O" & lngRowCounter)

If InStr(1, rngTargetStation.Value, Fstation3) And InStr(1, rngTargetFeeder.Value, Ffeeder3) <> 0 Then 'Checks for Projects that match the "From Station 3"

NC = Range("A" & lngRowCounter).Value 'If the station and feeder match, copy the generator information

ProjectName = Range("B" & lngRowCounter).Value

ProjectType = Range("H" & lngRowCounter).Value

Size = Range("J" & lngRowCounter).Value

CA = Range("K" & lngRowCounter).Value

TSstation = Range("R" & lngRowCounter).Value

TSfeeder = Range("O" & lngRowCounter).Value

DSstation = Range("N" & lngRowCounter).Value

DSfeeder = Range("M" & lngRowCounter).Value

DCAlink = Range("U" & lngRowCounter).Value

ProjectID = Range("G" & lngRowCounter).Value

Worksheets("Capacity").Activate 'Paste the Generator information into the Capacity worksheet

Range("A" & CapacityRowCounter).Value = NC

Range("B" & CapacityRowCounter).Value = ProjectName

Range("C" & CapacityRowCounter).Value = ProjectID

Range("H" & CapacityRowCounter).Value = ProjectType

Range("I" & CapacityRowCounter).Value = Size

Range("K" & CapacityRowCounter).Value = CA

Range("L" & CapacityRowCounter).Value = TSstation

Range("M" & CapacityRowCounter).Value = TSfeeder

Range("N" & CapacityRowCounter).Value = DSstation

Range("O" & CapacityRowCounter).Value = DSfeeder

Range("P" & CapacityRowCounter).Value = DCAlink

With Range("E" & CapacityRowCounter).Validation

.Delete

.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, \_

Operator:=xlBetween, Formula1:=dropdown

.IgnoreBlank = True

.InCellDropdown = True

.InputTitle = ""

.InputMessage = ""

.ErrorMessage = ""

.ShowInput = True

.ShowError = True

End With

Range("E" & CapacityRowCounter).Value = "Yes"

Range("J" & CapacityRowCounter).Value = "No"

With Range("F" & CapacityRowCounter).Validation

.Delete

.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, \_

Operator:=xlBetween, Formula1:=dropdown

.IgnoreBlank = True

.InCellDropdown = True

.InputTitle = ""

.InputMessage = ""

.ErrorMessage = ""

.ShowInput = True

.ShowError = True

End With

With Range("G" & CapacityRowCounter).Validation

.Delete

.Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, \_

Operator:=xlBetween, Formula1:=dropdown

.IgnoreBlank = True

.InCellDropdown = True

.InputTitle = ""

.InputMessage = ""

.ErrorMessage = ""

.ShowInput = True

.ShowError = True

End With

CapacityRowCounter = CapacityRowCounter + 1

Worksheets("Infobase Distributed Generators").Activate

End If

lngRowCounter = lngRowCounter + 1

Set rngTargetStation = Range("R" & lngRowCounter)

Set rngTargetFeeder = Range("O" & lngRowCounter)

Loop

End If

Worksheets("Capacity").Activate

Range("G19").Value = CapacityRowCounter - 28

Application.ScreenUpdating = True ' added to stop hiding worksheet.

End Sub

Sub CalculateCurt()

Dim Cap, FromGen, TotalGen, TTGen, CurtailedGen, RemainingCap, ScaleFactor As Double

Dim Include, TT As Boolean

Dim CapRowCounter, Stations, Gens As Integer

CapRowCounter = 28

TotalGen = 0

Stations = 0

TTGen = 0

AltNonTT = 0

CurtailedGen = 0

RemainingCap = 0

ScaleFactor = 0

FromGen = 0

If Range("G1").Value = "" Then

Msg = "Please run the capacity calculator and input a capacity amount" ' ensures capacity calculator is run.

Style = vbCritical

Response = MsgBox(Msg, Style)

Range("G1").Select

Exit Sub

End If

If Range("G2").Value = "" Then

Msg = "Please input voltage" ' ensures capacity calculator is run.

Style = vbCritical

Response = MsgBox(Msg, Style)

Range("G2").Select

Exit Sub

End If

If Range("B20").Value = "" Then

Msg = "Please input loading" ' ensures capacity calculator is run.

Style = vbCritical

Response = MsgBox(Msg, Style)

Range("B20").Select

Exit Sub

End If

Worksheets("Capacity").Activate

Set Stations = Range("A" & CapRowCounter)

Do While Not IsEmpty(Stations.Value)

Set Stations = Range("A" & CapRowCounter)

If Range("J" & CapRowCounter).Value = "Yes" And Range("E" & CapRowCounter).Value = "Yes" Then 'Generation remaining on normal supply is not curtailed

Range("D" & CapRowCounter).Value = Range("I" & CapRowCounter).Value

FromGen = FromGen + Range("I" & CapRowCounter).Value

With Range("D" & CapRowCounter).Font

.ColorIndex = 1

.Bold = False

End With

End If

If Range("J" & CapRowCounter).Value = "No" And Range("E" & CapRowCounter).Value = "Yes" Then 'Generation on alternate feeder is reset

Range("D" & CapRowCounter).Value = ""

End If

If Range("E" & CapRowCounter).Value = "Yes" Then 'Sums generation MW

TotalGen = TotalGen + Range("I" & CapRowCounter).Value

Gens = Gens + 1

If Range("F" & CapRowCounter).Value = "Yes" Or (Range("G" & CapRowCounter).Value = "Yes" And Range("J" & CapRowCounter).Value = "Yes") Then

TTGen = TTGen + Range("I" & CapRowCounter).Value 'Finds TT Generation MW total

End If

End If

If Range("G" & CapRowCounter).Value = "Yes" And Range("J" & CapRowCounter).Value = "No" Then

Range("D" & CapRowCounter).Value = 0 'If an alternate load displacement unit has TT, curtail to 0 MW.

With Range("D" & CapRowCounter).Font

.ColorIndex = 1

.Bold = False

End With

End If

If Range("G" & CapRowCounter).Value = "No" And Range("J" & CapRowCounter).Value = "No" Then

Range("D" & CapRowCounter).Value = Range("I" & CapRowCounter) 'If an alternate load displacement unit does not have TT, do not curtail

With Range("D" & CapRowCounter).Font

.ColorIndex = 1

.Bold = False

End With

End If

If Range("F" & CapRowCounter).Value = "Yes" And Range("J" & CapRowCounter).Value = "No" Then

Range("D" & CapRowCounter).Value = Range("I" & CapRowCounter) 'If alternate gen has TT with To feeder, do not curtail

With Range("D" & CapRowCounter).Font

.ColorIndex = 1

.Bold = False

End With

End If

If Range("E" & CapRowCounter).Value = "No" Then 'excludes DGs which are not included so they do not consume any generation capacity

Range("D" & CapRowCounter).Value = ""

With Range("D" & CapRowCounter).Font

.ColorIndex = 1

.Bold = False

End With

End If

If Range("D" & CapRowCounter).Value = "" And Range("J" & CapRowCounter).Value = "No" And Range("E" & CapRowCounter).Value = "Yes" Then

CurtailedGen = CurtailedGen + Range("I" & CapRowCounter).Value

End If

CapRowCounter = CapRowCounter + 1

Loop

RemainingCap = Range("G23").Value - WorksheetFunction.Sum(Range("D28:D" & CapRowCounter - 2)) + TTGen

Range("I24").Value = RemainingCap

If RemainingCap > Range("G22").Value Then 'Ensures that remainingcap does not exceed feeder limitations (used to have "- FromGen" as part of formula

RemainingCap = Range("G22").Value

Range("I24").Value = RemainingCap

End If

If RemainingCap < 0 Then

Msg = "By following NMI 9023, we have a potential island situation. Adjust as required" ' checks to see if there is capacity to allow downstream gens to connect.

Style = vbCritical

Response = MsgBox(Msg, Style)

Range("D27").Select

Exit Sub

End If

CapRowCounter = 28

Set Stations = Range("A" & CapRowCounter)

Do While Not IsEmpty(Stations.Value) 'allocates generation to downstream gens (DS gens) on alternate feeders and projects <250kW

Set Stations = Range("A" & CapRowCounter)

If Range("D" & CapRowCounter).Value = "" And Range("J" & CapRowCounter).Value = "No" And Range("F" & CapRowCounter).Value = "No" And Range("E" & CapRowCounter).Value = "Yes" And (Range("N" & CapRowCounter).Value <> "" Or Range("I" & CapRowCounter).Value < "0.250") Then

Range("D" & CapRowCounter).Value = Range("I" & CapRowCounter).Value

With Range("D" & CapRowCounter).Font

.ColorIndex = 1

.Bold = False

End With

RemainingCap = RemainingCap - Range("I" & CapRowCounter).Value

CurtailedGen = CurtailedGen - Range("I" & CapRowCounter).Value

End If

CapRowCounter = CapRowCounter + 1

Loop

If RemainingCap < 0 Then

Msg = "Potential island situation. Small DS generators may need to be curtailed. Override Size column as required" ' if by adding downstream gens, capacity is taken up, then alert the user

Style = vbCritical

Response = MsgBox(Msg, Style)

Range("D27").Select

Exit Sub

End If

If CurtailedGen > 0 Then

ScaleFactor = RemainingCap / CurtailedGen

End If

If CurtailedGen = 0 Then

Msg = "No generators to curtail" ' if there's nothing to curtail, alert user

Style = vbCritical

Response = MsgBox(Msg, Style)

Range("D27").Select

Exit Sub

End If

If ScaleFactor > 1 Then

ScaleFactor = 1

End If

CapRowCounter = 28

Set Stations = Range("A" & CapRowCounter)

Do While Not IsEmpty(Stations.Value) 'allocates generation to those on alternate feeders, curtailed based on scaling factor

Set Stations = Range("A" & CapRowCounter)

If Range("D" & CapRowCounter).Value = "" And Range("J" & CapRowCounter).Value = "No" And Range("F" & CapRowCounter).Value = "No" And Range("E" & CapRowCounter).Value = "Yes" Then

Range("D" & CapRowCounter).Value = Range("I" & CapRowCounter).Value \* ScaleFactor

If Range("D" & CapRowCounter).Value = Range("I" & CapRowCounter).Value Then

With Range("D" & CapRowCounter).Font

.ColorIndex = 1

.Bold = False

End With

End If

If Range("D" & CapRowCounter).Value < Range("I" & CapRowCounter).Value Then

With Range("D" & CapRowCounter).Font

.ColorIndex = 3

.Bold = True

End With

End If

End If

CapRowCounter = CapRowCounter + 1

Loop

CapRowCounter = 28

Range("I19").Value = Gens

Range("I20").Value = TotalGen

Range("I21").Value = TTGen

Range("I22").Value = FromGen

Range("I23").Value = TotalGen - FromGen

Range("I25").Value = ScaleFactor

End Sub

Sub LoadDCAs()

Dim RowCounter As Integer

Dim State As Range

Dim URL As String

RowCounter = 28

Set State = Range("A" & RowCounter)

Do While Not IsEmpty(State.Value)

Set State = Range("A" & RowCounter)

URL = Range("P" & RowCounter).Text

Shell "C:\Program Files\Internet Explorer\iexplore.exe " & URL, vbNormalFocus 'vbNormalFocus

RowCounter = RowCounter + 1

Loop

End Sub