|  |  |
| --- | --- |
| *Pacific Gas and Electric Company* | |
| Patch 7.6 Installation Guide | |
|  |  |
|  |  |
| Project | ED AM/GIS |
|  |  |
| Prepared by | Ashish Narasimham |
| Date | 8/23/2013 |
| Version | 1.1 |
| Version Type | Final |

|  |  |  |  |
| --- | --- | --- | --- |
| Revision History | | | |
| Document # | Date | Author | Summary of Changes |
| 1.0 | 8/23/2013 | Ashish Narasimham | Initial Document Creation |
| 1.1 | 8/21/13 | Ashish Narasimham | Minor changes, formatting |
| 1.2 | 8/22/13 | Ashish Narasimham | Added in the 7 anno changes plus move Notes from Common Features to Electric |

# Introduction

## Purpose

This document is intended to detail the implementation and configuration steps required to implement the Patch 7.6 Installation Guide. This document describes the various configuration aspects required to complete any manual or automatic patch associated with this release. Each section in this document contains the steps required to patch the system in production.

## Terms Used

|  |  |
| --- | --- |
| OOTB | Out of the box. Unmodified from the commercial version. |
| TFS | Team Foundation Server |

## External Documents

Referenced are any external configuration documents or exports. These are documents that contain more detailed information about configuring a system or documents that can be loaded into an application to perform the configuration detailed in this document.

1. File Geodatabase Location (used starting in Section 4)
   1. The PGDB is located at: [\\sfetgis-nas01\sfgispoc\_data\Data Conversion\Team Members\Peaslee\Data Model\](file:///\\sfetgis-nas01\sfgispoc_data\Data%20Conversion\Team%20Members\Peaslee\Data%20Model\%20) . It is called *Schematic Annotation.gdb*

## Summary of Steps to Complete Patch

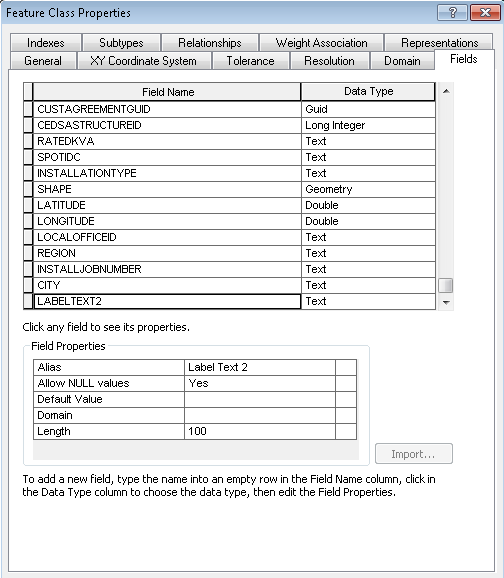
These are the high-level steps to complete the installation and configuration of the data model patch. Use this table as a guide for completing the installation. Links are provided that can lead either within the document for detailed explanations or to external sites such as Sharepoint.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Step Name** | **Description** | **CR (if applicable)** |
|  | Master For Data Model Release 7.5.9 - Schematics | Master for data model release 7.5.9, all related tickets are linked | [9452](http://edappgistfsprd1:8080/tfs/ElectricDistCollection/EDAMGIS/_workItems#id=9452&_a=edit) |
|  | [Open a Database Connection](#_Open_a_Database) | Open a database connection in ArcCatalog | - |
|  | [Add LabelText2 field to Transfomer](#_Add_LabelText2_field) | Added Label Text 2 field to transformer feature class | [9453](http://edappgistfsprd1:8080/tfs/ElectricDistCollection/EDAMGIS/_workItems#id=9453&_a=edit) |
|  | [Create All Schematics Annotation Classes](#_Create_All_Schematics) | Copy all Schematics annotation feature classes from a PGDB | [9342](http://edappgistfsprd1:8080/tfs/ElectricDistCollection/EDAMGIS/_workItems#id=9342&_a=edit)  [9343](http://edappgistfsprd1:8080/tfs/ElectricDistCollection/EDAMGIS/_workItems#id=9343&_a=edit)  [9344](http://edappgistfsprd1:8080/tfs/ElectricDistCollection/EDAMGIS/_workItems#id=9344&_a=edit)  [9345](http://edappgistfsprd1:8080/tfs/ElectricDistCollection/EDAMGIS/_workItems#id=9345&_a=edit)  [9346](http://edappgistfsprd1:8080/tfs/ElectricDistCollection/EDAMGIS/_workItems#id=9346&_a=edit)  [9347](http://edappgistfsprd1:8080/tfs/ElectricDistCollection/EDAMGIS/_workItems#id=9347&_a=edit)  [9348](http://edappgistfsprd1:8080/tfs/ElectricDistCollection/EDAMGIS/_workItems#id=9348&_a=edit)  [9349](http://edappgistfsprd1:8080/tfs/ElectricDistCollection/EDAMGIS/_workItems#id=9349&_a=edit)  [9406](http://edappgistfsprd1:8080/tfs/ElectricDistCollection/EDAMGIS/_workItems#id=9406&_a=edit)  [9407](http://edappgistfsprd1:8080/tfs/ElectricDistCollection/EDAMGIS/_workItems#id=9407&_a=edit)  [9408](http://edappgistfsprd1:8080/tfs/ElectricDistCollection/EDAMGIS/_workItems#id=9408&_a=edit) |
|  | [Create Relationship Classes for Each Schematics Anno Class](#_Create_Relationship_Classes) | Create a relationship class for each annotation feature class created in the previous step | - |
|  | [Set Permissions On Created Feature/Relationship Classes](#_Set_Permissions_on) | Set standard permissions on all created annotation feature classes and relationship classes | - |
|  | [Modify the Electric Annotation and Update them to fix spacing of the default anno class when placed.](#_Modify_the_Electric) | Change label expressions and SQL queries to fix spacing issue | [9588](http://edappgistfsprd1:8080/tfs/ElectricDistCollection/EDAMGIS/_workItems#_a=edit&id=9588) |
|  | [Bug 9539: Move Tables From Common Features Dataset To Electric Dataset](#_Bug_9539:_Move) | Move Note feature classes to ED so they can be versioned | [9567](http://edappgistfsprd1:8080/tfs/ElectricDistCollection/EDAMGIS/_workItems#_a=edit&id=9567) |
|  | [Register Dataset as Versioned](#_Register_the_Dataset) | Register the Electric Dataset as versioned | - |
|  | [Update Data Model Version Table](#_Update_Data_Model) | Update the data model version table to reflect this update | - |

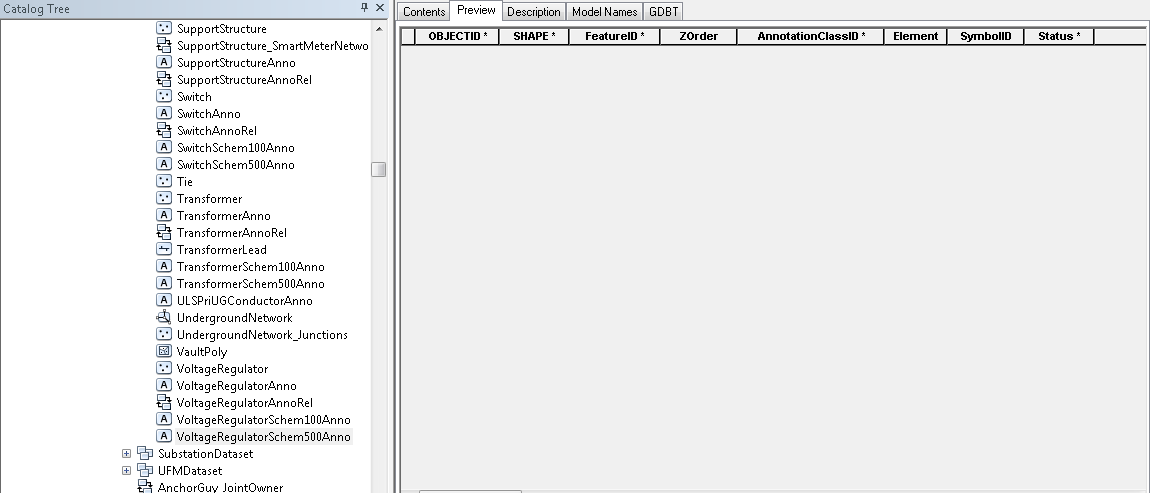
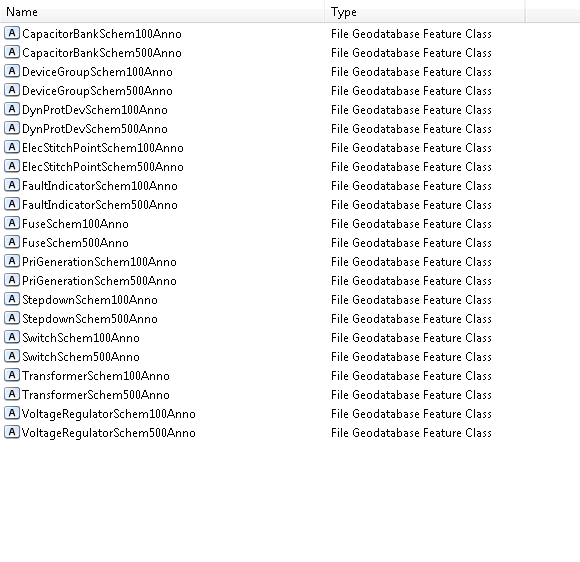
# Open a Database Connection in ArcCatalog

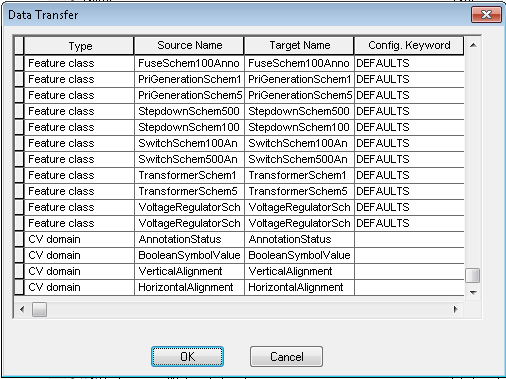
1. Open ArcCatalog.
2. Within the Catalog Tree, expand “Database Connections” and open the active connection for this process. This is the connection that is referenced in the change request associated with this document (EDGIS<DB name in the format X#Y> )

# Add LabelText2 field to Transfomer

1. In the Electric Dataset, navigate to the Transformer feature class, right click and open Properties
2. Select the Fields tab
3. Scroll to the bottom of the Fields list and add a new field, “LABELTEXT2”. Fill in the Field Properties section as below
4. Click OK to apply the changes

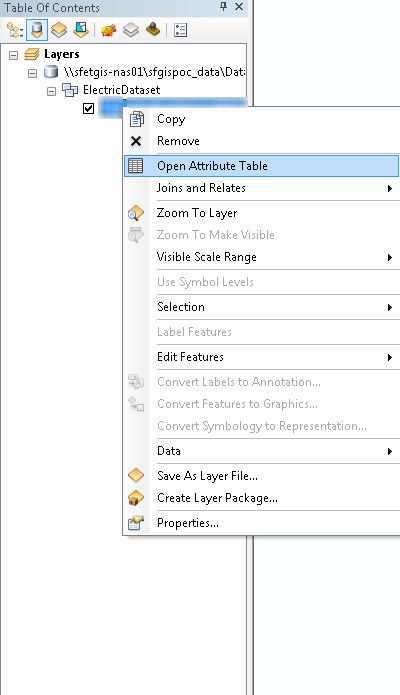
# Create All Schematics Annotation Classes

1. In the File Geodatabase provided (refer to Section 1.3), locate the 22 Schematics annotation feature classes under the Electric Dataset
2. Validate that the feature classes are empty of all data by clicking on the feature class in the left pane, clicking the Preview tab in the main window, and selecting Table as the value for the Preview: dropdown at the bottom of the window. If data exists in the Preview tab, please refer to the end of Section 4 to delete all data  
   
3. Select all 22 annotation feature classes, and then drag and drop them into the target SDE database  
   
4. Ensure there are no **red** items in the confirmation dialog and click OK



## Deleting All Data from a Table

**Only perform this step if you found data in any of the tables from step 2 of Section 4**

1. To delete all data from a table, open up ArcMap and log in
2. Click the Add Data… button, navigate to the table that has data in it, and double click it to add it to the data frame
3. Right click the table in the Data Source tab of the Table of Contents and click Open Attribute Table
4. Start editing by clicking Editor->Start Editing
5. Select all rows in the table and press Delete

# Create Relationship Classes for Each Schematics Anno Class

1. You will be creating a relationship class for each of the items in the following list:

CapacitorBankSchem100AnnoRel

CapacitorBankSchem500AnnoRel

DeviceGroupSchem100AnnoRel

DeviceGroupSchem500AnnoRel

DynProtDevSchem100AnnoRel

DynProtDevSchem500AnnoRel

ElecStitchPointSchem100AnnoRel

ElecStitchPointSchem500AnnoRel

FaultIndicatorSchem100AnnoRel

FaultIndicatorSchem500AnnoRel

FuseSchem100AnnoRel

FuseSchem500AnnoRel

PriGenerationSchem100AnnoRel

PriGenerationSchem500AnnoRel

StepdownSchem100AnnoRel

StepdownSchem500AnnoRel

SwitchSchem100AnnoRel

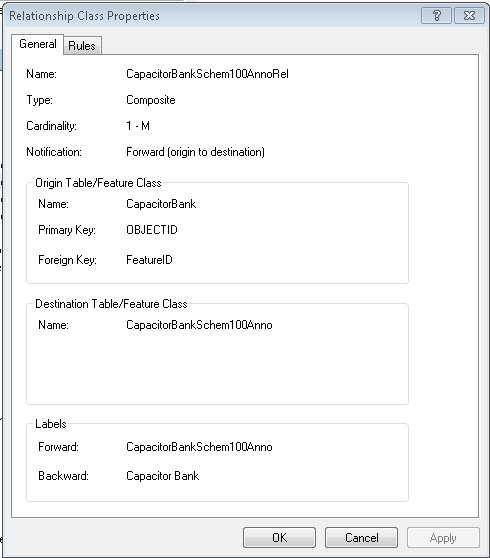
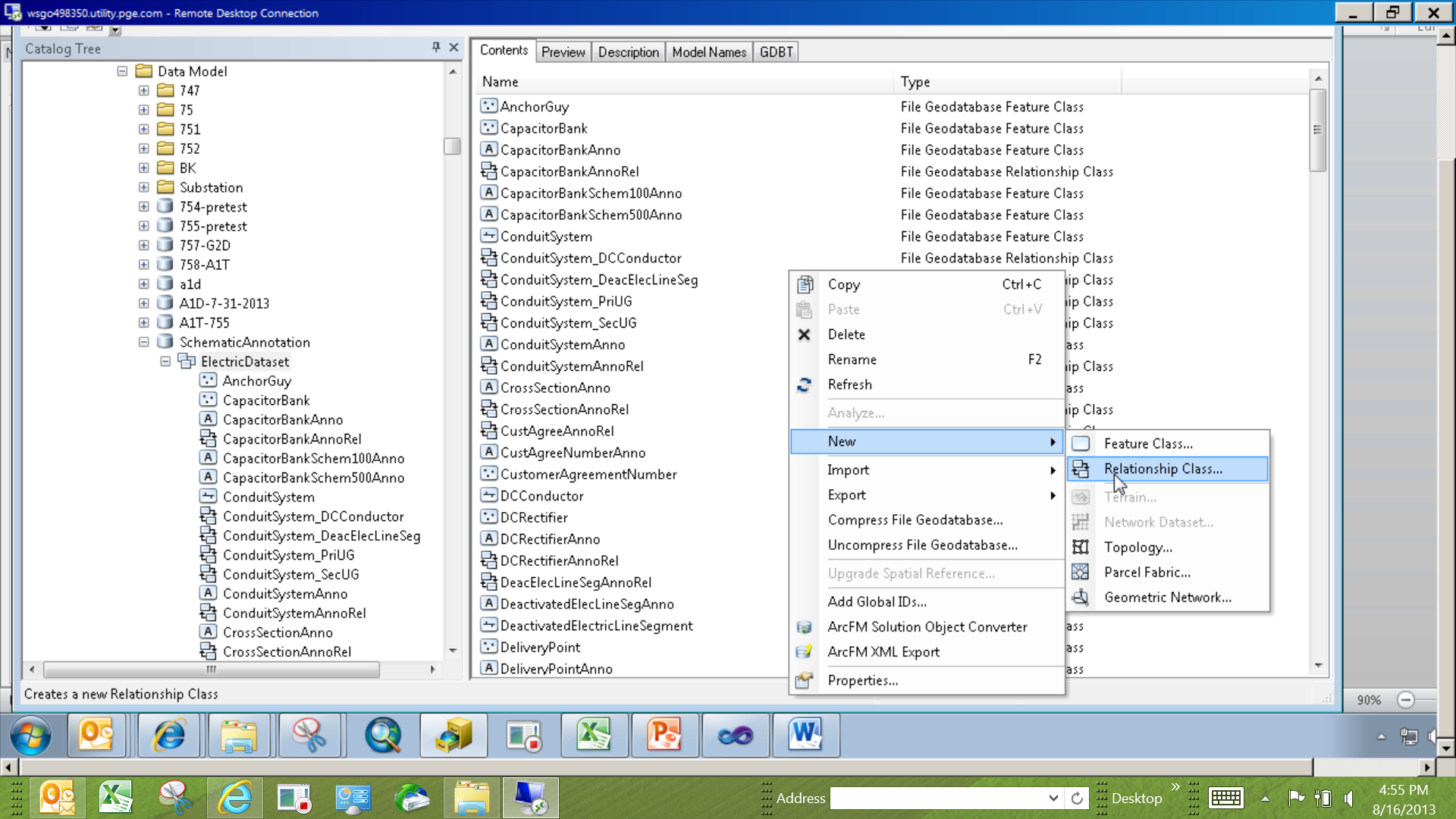
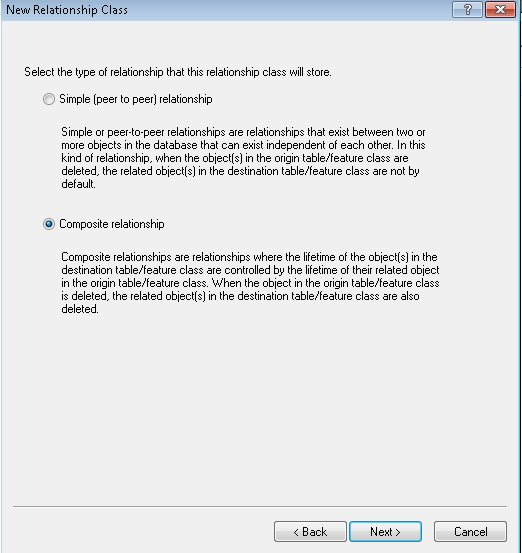
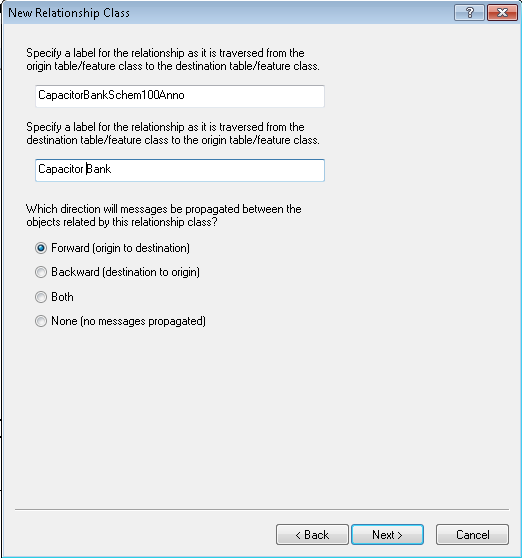
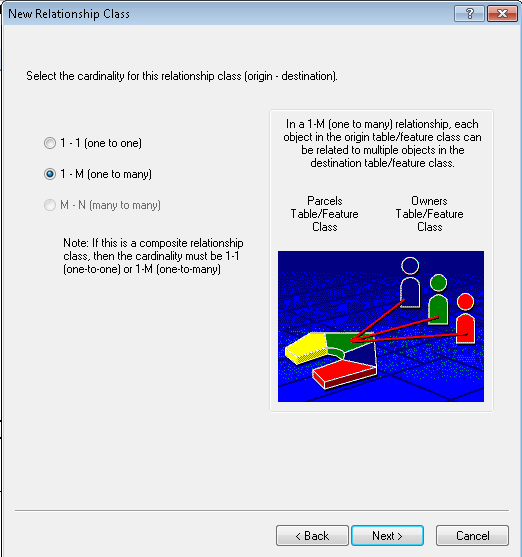
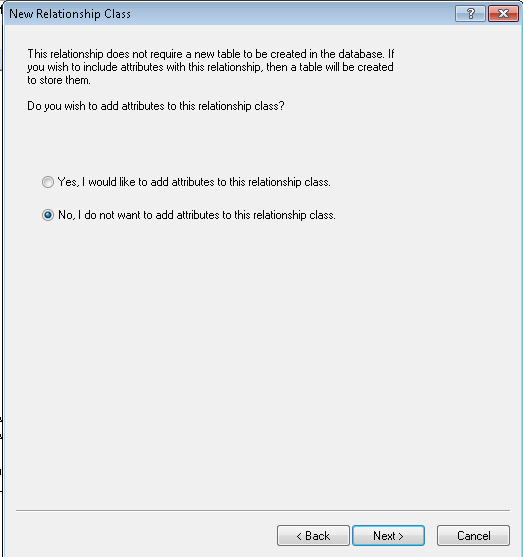
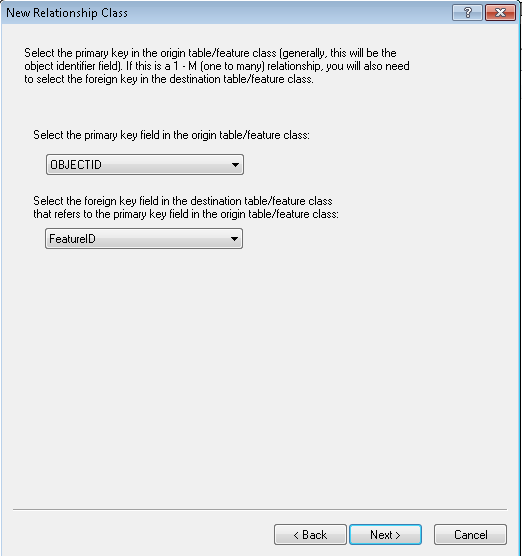
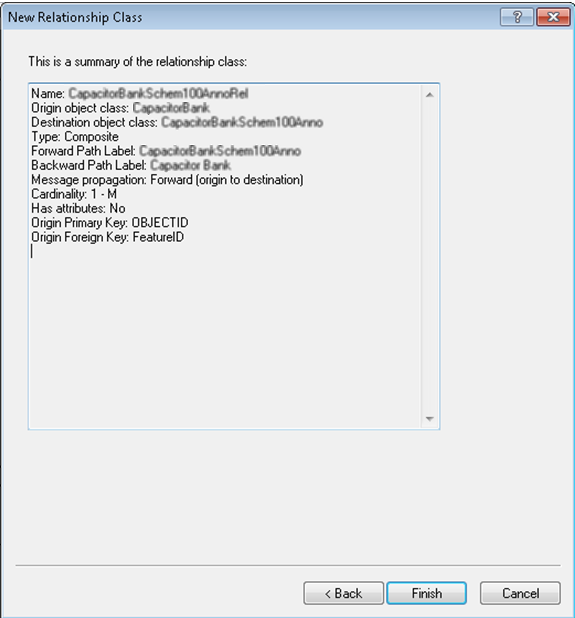
SwitchSchem500AnnoRel

TransformerSchem100AnnoRel

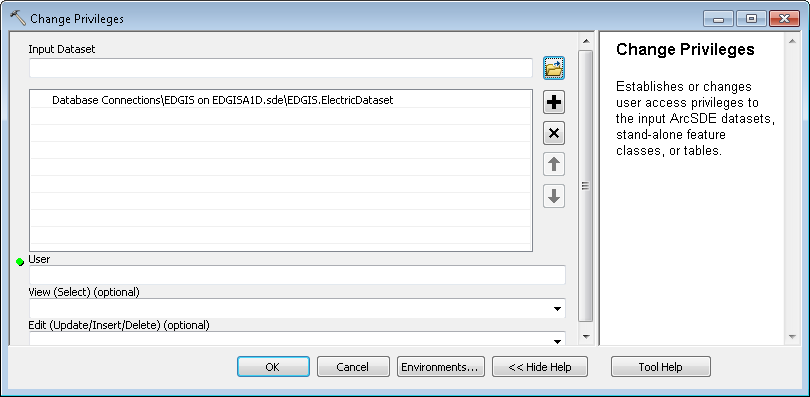
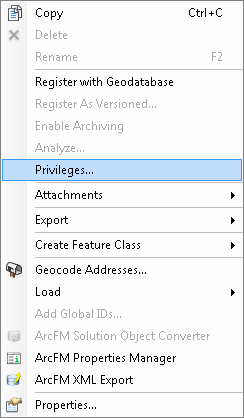
TransformerSchem500AnnoRel

VoltageRegulatorSchem100AnnoRel

VoltageRegulatorSchem500AnnoRel

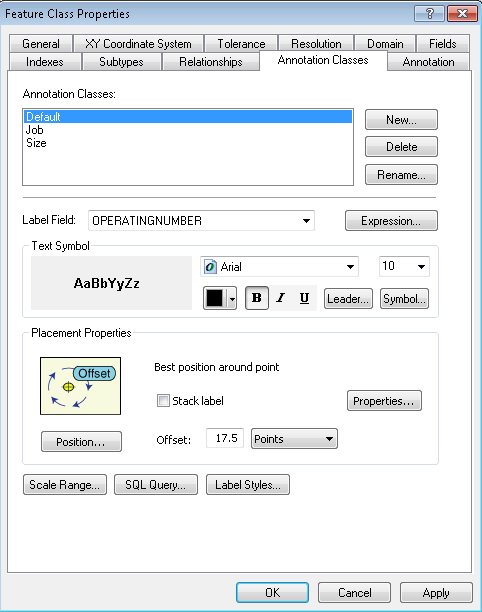
1. Use the following diagram as a reference if necessary:  
   
2. Right click on an empty area on the right pane in ArcCatalog inside the Electric Dataset of the database you are updating, select New->Relationship Class  
   
3. Fill in the “Name”, “Origin Table”, and “Destination Table” with the appropriate values and click Next. Refer to the diagram at the end of this section to determine the appropriate values.   
     
   Set the name to the relationship class’s name, the origin table to the feature class the anno is for, and the destination table to the annotation class that you dragged and dropped earlier. For example, CapacitorBankSchem100AnnoRel would have a name of “CapacitorBankSchem100AnnoRel”, Origin Table: “CapacitorBank”, Destination Table: “CapacitorBankSchem100Anno”. Note that the table here is the annotation feature class, not the relationship class
4. Set to Composite Relationship and click Next  
   
5. Set the forward and backward labels as below, set the direction to Forward, and click Next. Be sure to set the origin-destination label to the name of the annotation class and the destination-origin label to the name of the feature class that the annotation is for  
   
6. Verify that the “1-M” is already set on the next page or set it if not set already. Click Next  
   
7. Verify that the attribute settings are set to “No” or set them if not set already. Click Next  
   
8. Set the relationship keys to match the following screenshot. Primary key=OBJECTID and foreign key=FeatureID  
   
9. Verify your choices on the next page:  
   

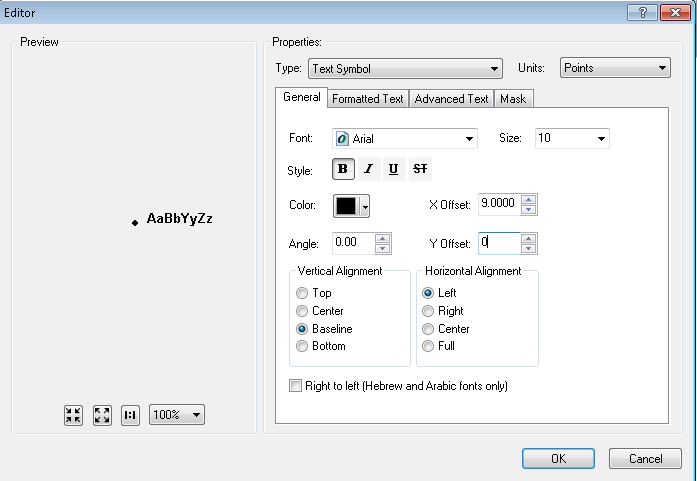
# Set Permissions on Created Feature/Relationship Classes

1. Right click the Electric Dataset and select ‘Privileges…’ Assign the standard set of permissions (SDE\_EDITOR, SDE\_VIEWER, etc.).   
   

# Modify the Electric Annotation and Update them to fix spacing of the default anno class when placed.

Use the following screenshots as a reference to modify all relevant screens. “Expressions” refers to the “Expression…” button on the main screen, “SQL Query” refers to the “SQL Query…” button on the main screen, and the “Editor” window can be found by navigating to Symbol->Edit Symbol.  
**Capacitor**



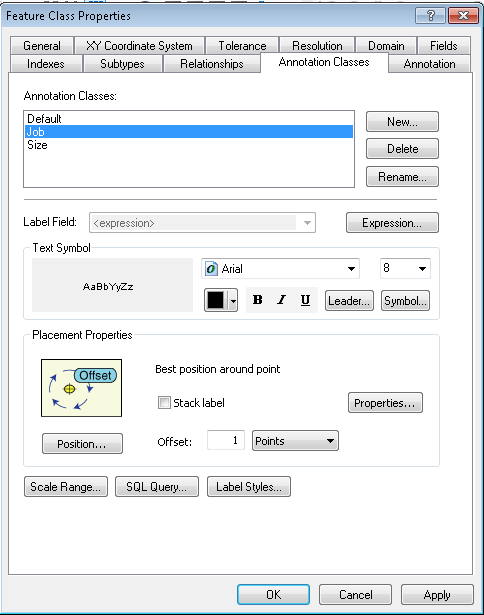


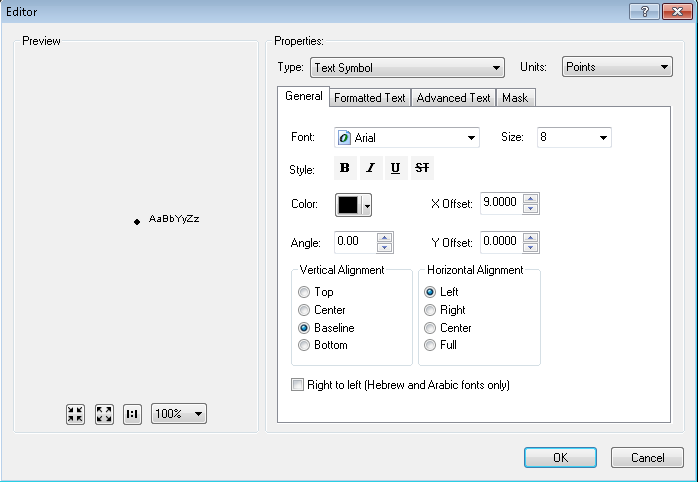
Expression:

OPERATINGNUMBER

SQL Query:

OPERATINGNUMBER is not null





Expression:

Function FindLabel ( [INSTALLJOBPREFIX] , [INSTALLJOBNUMBER] , [INSTALLJOBYEAR] )

myString = ""

if left(ucase( [INSTALLJOBPREFIX] ), 3) <> "UNK" and ucase( [INSTALLJOBPREFIX] ) <> "FIF" and \_

len( [INSTALLJOBPREFIX] ) > 0 then

myString = [INSTALLJOBPREFIX]

end if

if len( [INSTALLJOBYEAR] ) > 0 then

if len( [INSTALLJOBYEAR] ) = 4 then

FindLabel = myString + [INSTALLJOBNUMBER] + " '"+ right( [INSTALLJOBYEAR] ,2)

elseif len( [INSTALLJOBYEAR] ) = 2 then

FindLabel = myString + [INSTALLJOBNUMBER] + " '"+ [INSTALLJOBYEAR]

end if

else

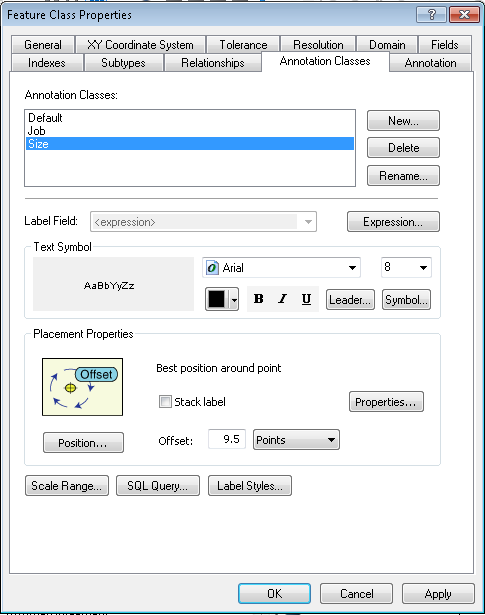
FindLabel = myString + [INSTALLJOBNUMBER]

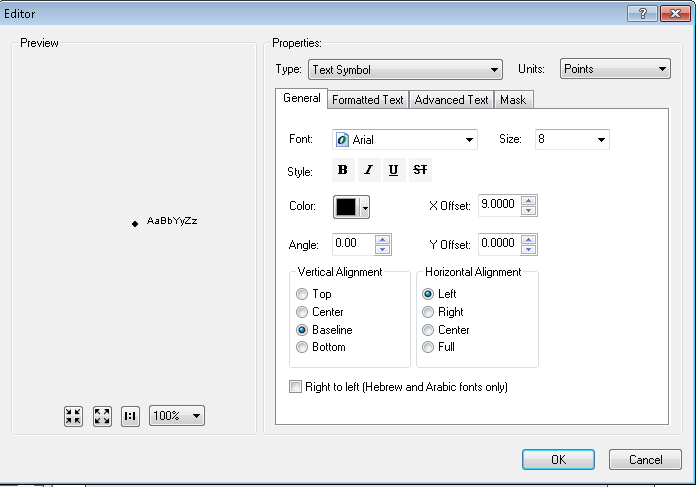
end if

End Function

SQL Query:

not (INSTALLJOBNUMBER is null or INSTALLJOBNUMBER = '0' or INSTALLJOBNUMBER like '%CEDSA%')





Expression:

Function FindLabel ( [TotalKVAR], [SubtypeCD] )

dim myVar

MyTotalKVAR = [TotalKVAR]

if [SubtypeCD] = "Fixed Bank Capacitor" then

myVar = "F"

else

myVar = "S"

end if

if ([SubtypeCD] = "Fixed Bank Capacitor" or [SubtypeCD] = "Switched Bank Capacitor") then

FindLabel = [TotalKVAR] + myVar

else

FindLabel = [TotalKVAR]

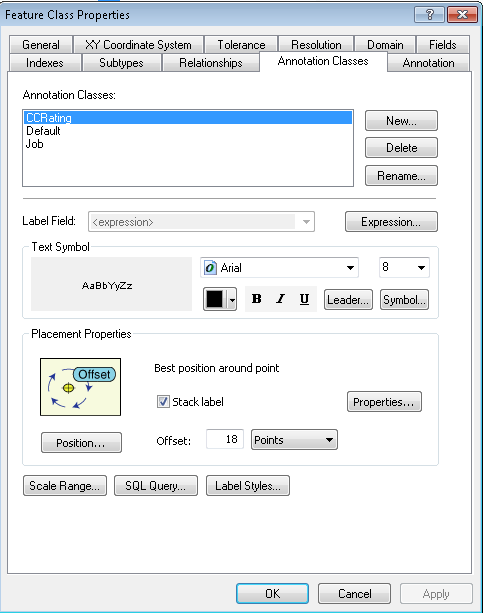
end if

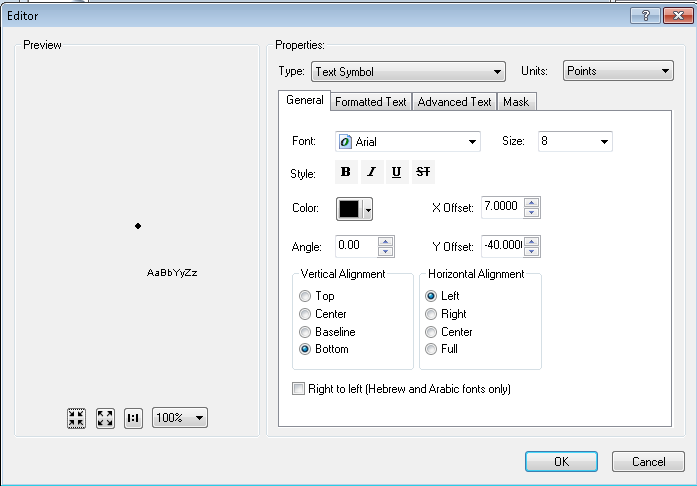
End Function

SQL Query:

TOTALKVAR is not null

**Dynamic Protective Device**





Expression:

Function FindLabel ( [CCRating], [SwitchModeIdc])

if [SwitchModeIdc]= "Yes" then

FindLabel = [CCRating]& "A-SM"

else

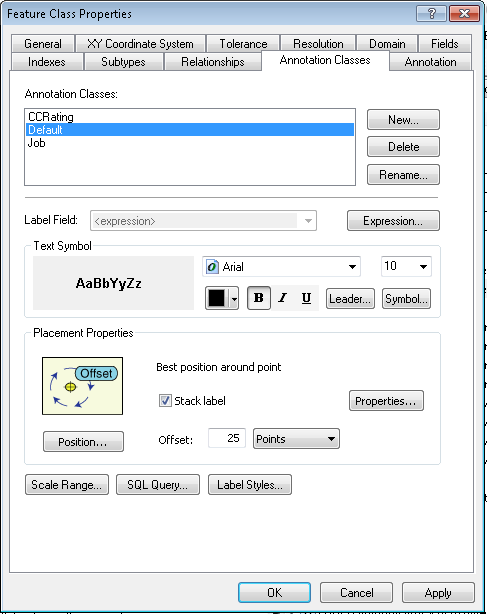
FindLabel = [CCRating]& "A"

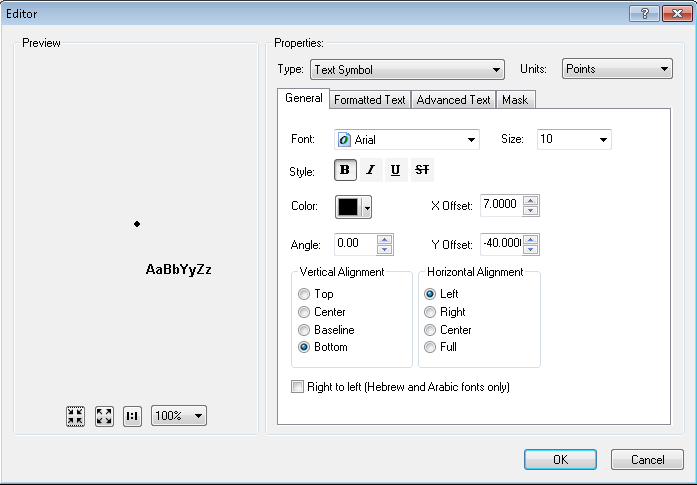
End if

End Function

SQL Query:

SubtypeCD = 2 and CCRating is not null





Expression:

Function FindLabel ( [SubtypeCD] ,[OperatingNumber], [MULTIFUNCTIONALIDC] )

If [SubtypeCD] = "Recloser" and [MULTIFUNCTIONALIDC] = "Yes" then

FindLabel = [OperatingNumber] & "-MF"

else

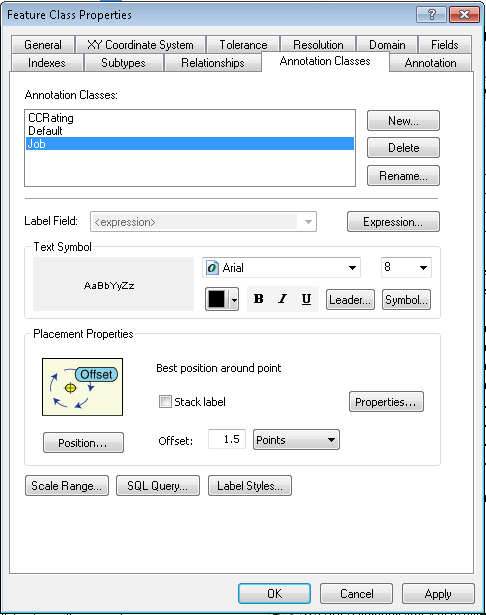
FindLabel = [OperatingNumber]

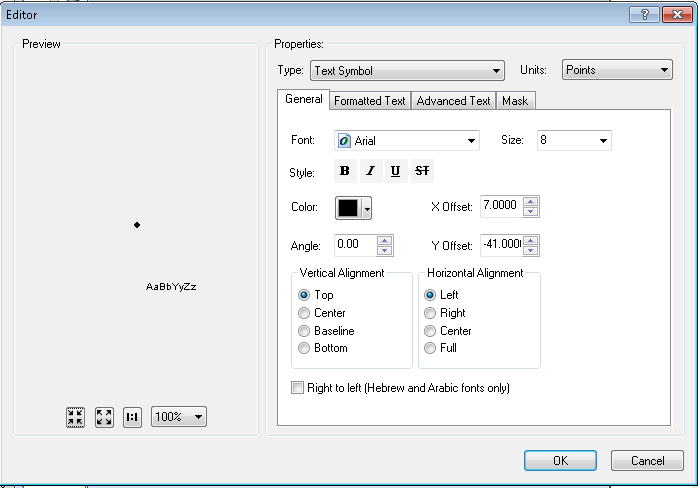
end if

End Function

SQL Query:

**DO NOT PUT IN AN SQL QUERY**





Expression:

Function FindLabel ( [InstallJobPrefix], [InstallJobNumber], [InstallJobYear], [SUBTYPECD], [SWITCHMODEIDC] )

myString = ""

if isNull( [INSTALLJOBNUMBER] ) or [INSTALLJOBNUMBER] = "0" or \_

instr(ucase( [INSTALLJOBNUMBER] ), "CEDSA") then

else

if left(ucase( [INSTALLJOBPREFIX]), 3 ) <> "UNK" and \_

ucase( [INSTALLJOBPREFIX] ) <> "FIF" and \_

len( [INSTALLJOBPREFIX] ) > 0 then

myString = [INSTALLJOBPREFIX]

end if

if len( [INSTALLJOBYEAR] ) > 0 then

if len( [INSTALLJOBYEAR] ) = 4 then

myString = myString + [INSTALLJOBNUMBER] + " '"+ right( [INSTALLJOBYEAR] ,2)

elseif len( [INSTALLJOBYEAR] ) = 2 then

myString = myString + [INSTALLJOBNUMBER] + " '"+ [INSTALLJOBYEAR]

end if

else

myString = myString + [INSTALLJOBNUMBER]

end if

end if

If [SubtypeCD] = "Recloser" or [SubtypeCD] = "Sectionalizer" or [SubtypeCD] = "Interrupter" then

If [SubtypeCD] <> "Interrupter" then

FindLabel = myString + vbCrLf + chr(129) + vbCrLf + chr(129)

else

FindLabel = myString + vbCrLf + chr(129)

end if

else

FindLabel = myString + vbCrLf + chr(129) + vbCrLf + chr(129) + vbCrLf + chr(129)

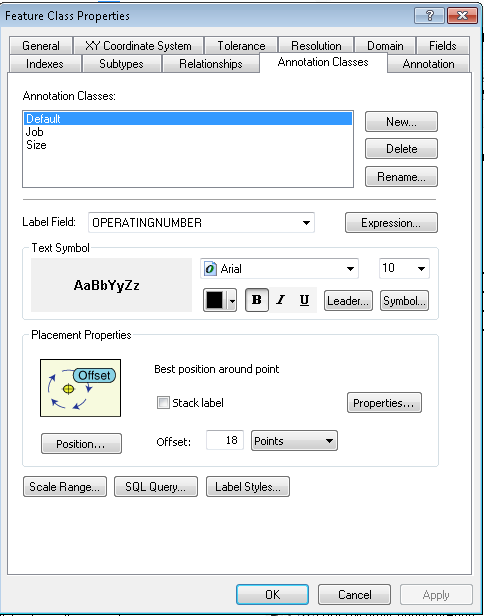
end if

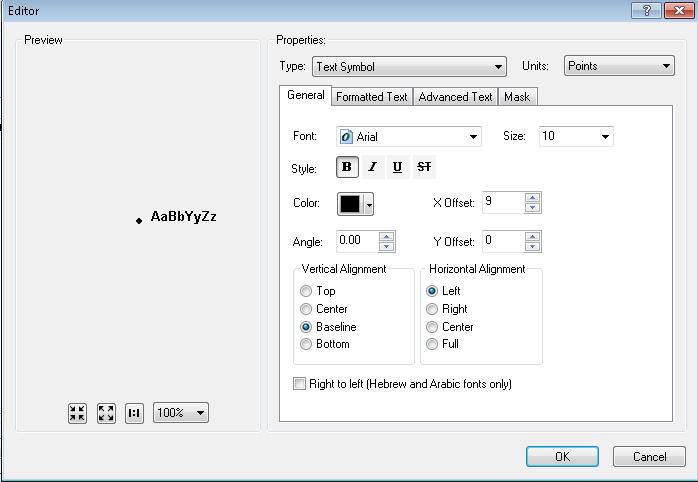
End Function

SQL Query:

**DO NOT PUT IN AN SQL QUERY**

**Fuse**



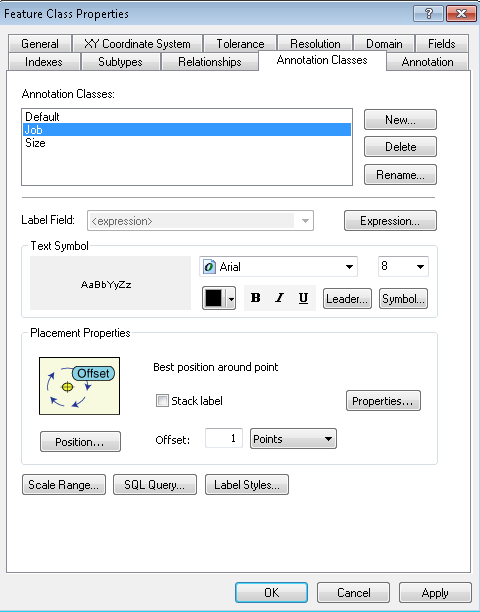


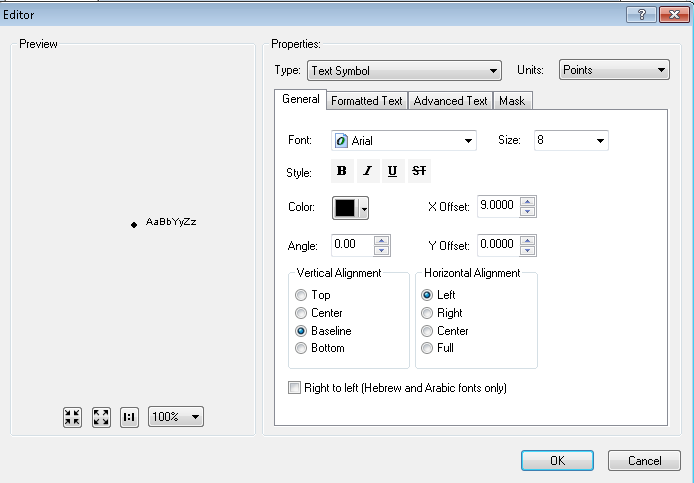
Expression:

OPERATINGNUMBER

SQL Query:

OPERATINGNUMBER is not null





Expression:

Function FindLabel ( [INSTALLJOBPREFIX], [INSTALLJOBNUMBER], [INSTALLJOBYEAR])

myString = ""

if left(ucase( [INSTALLJOBPREFIX]), 3 ) <> "UNK" and \_

ucase( [INSTALLJOBPREFIX] ) <> "FIF" and \_

len( [INSTALLJOBPREFIX] ) > 0 then

myString = [INSTALLJOBPREFIX]

end if

if len( [INSTALLJOBYEAR] ) > 0 then

if len( [INSTALLJOBYEAR] ) = 4 then

FindLabel = myString + [INSTALLJOBNUMBER] + " '"+ right( [INSTALLJOBYEAR] ,2)

elseif len( [INSTALLJOBYEAR] ) = 2 then

FindLabel = myString + [INSTALLJOBNUMBER] + " '"+ [INSTALLJOBYEAR]

end if

else

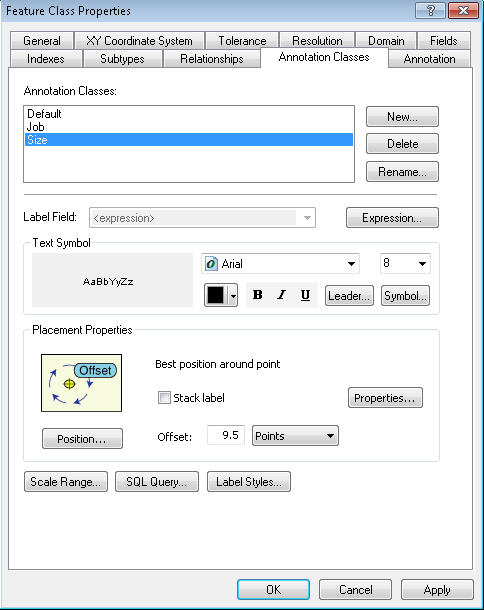
FindLabel = myString + [INSTALLJOBNUMBER]

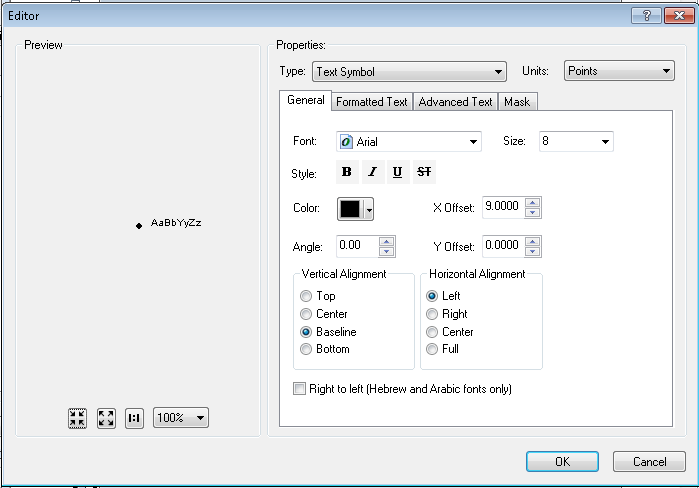
end if

End Function

SQL Query:

not (INSTALLJOBNUMBER is null or INSTALLJOBNUMBER = '0' or INSTALLJOBNUMBER like '%CEDSA%')





Expression:

Function FindLabel ( [LinkRating], [LinkType])

myLink = ""

if uCase([LINKTYPE]) <> "UNSPECIFIED" and uCase( [LINKTYPE] ) <> "NOT ON LIST" then

myLink = [LINKTYPE]

end if

myRatinglen = len([LinkRating]) - 2

myRating = mid([LinkRating],1,myRatinglen)

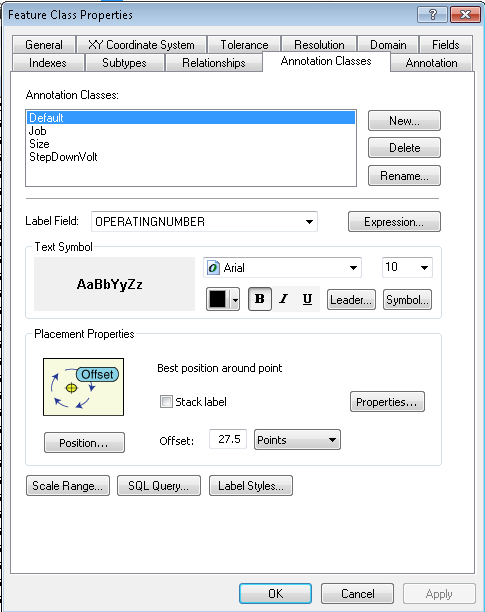
FindLabel = myRating + myLink

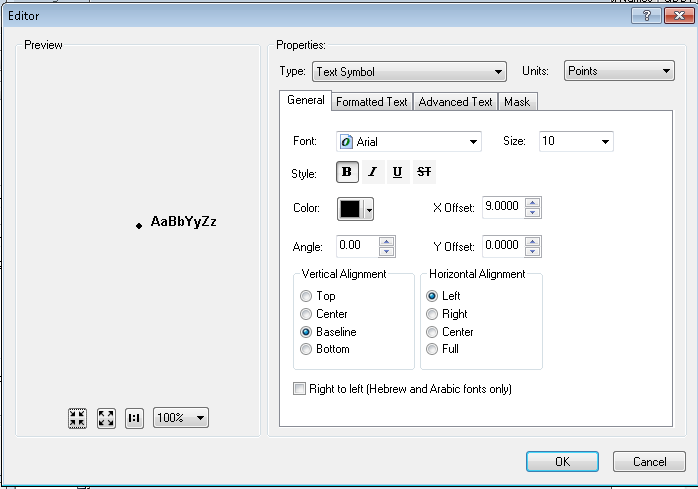
End Function

SQL Query:

LINKRATING is not null or LINKRATING <> -99

**STEPDOWN**



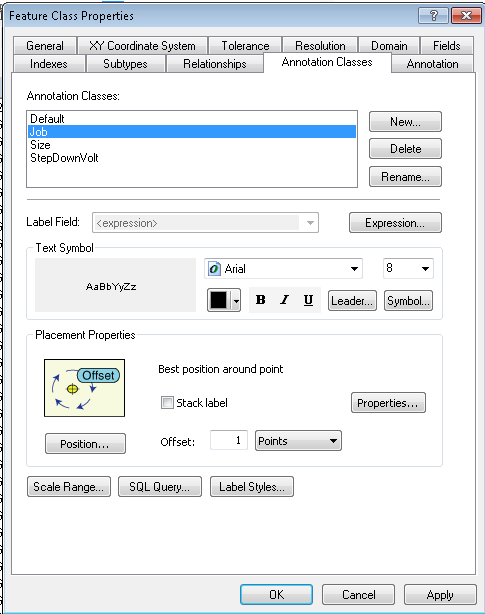


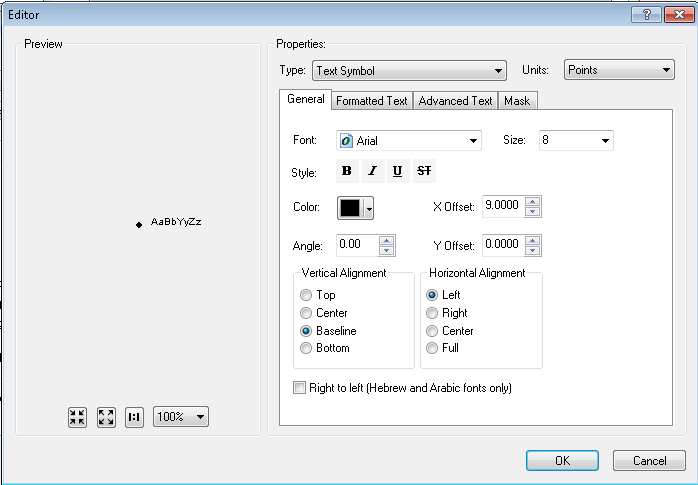
Expression:

OPERATINGNUMBER

SQL Query:

OPERATINGNUMBER is not null





Expression:

Function FindLabel( [InstallJobPrefix], [InstallJobNumber], [InstallJobYear], [INSTALLATIONTYPE])

myString = ""

if left(ucase( [INSTALLJOBPREFIX]), 3 ) <> "UNK" and \_

ucase( [INSTALLJOBPREFIX] ) <> "FIF" and \_

len( [INSTALLJOBPREFIX] ) > 0 then

myString = [INSTALLJOBPREFIX]

end if

if len( [INSTALLJOBYEAR] ) > 0 then

if len( [INSTALLJOBYEAR] ) = 4 then

FindLabel = myString + [INSTALLJOBNUMBER] + " '"+ right( [INSTALLJOBYEAR] ,2)

elseif len( [INSTALLJOBYEAR] ) = 2 then

FindLabel = myString + [INSTALLJOBNUMBER] + " '"+ [INSTALLJOBYEAR]

end if

else

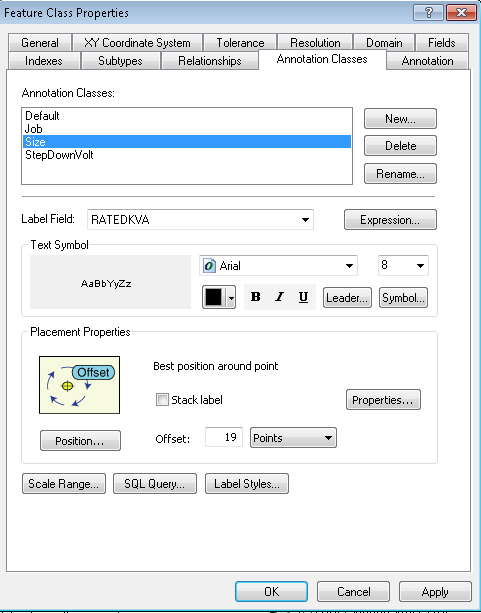
FindLabel = myString + [INSTALLJOBNUMBER]

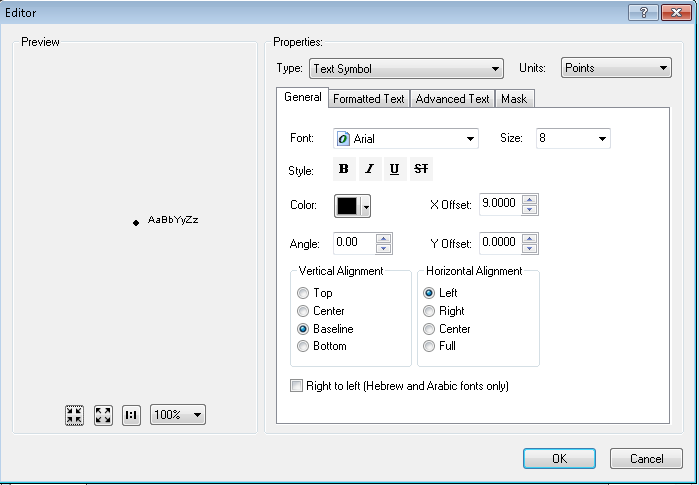
end if

End Function

SQL Query:

not (INSTALLJOBNUMBER is null or INSTALLJOBNUMBER = '0' or INSTALLJOBNUMBER like '%CEDSA%')





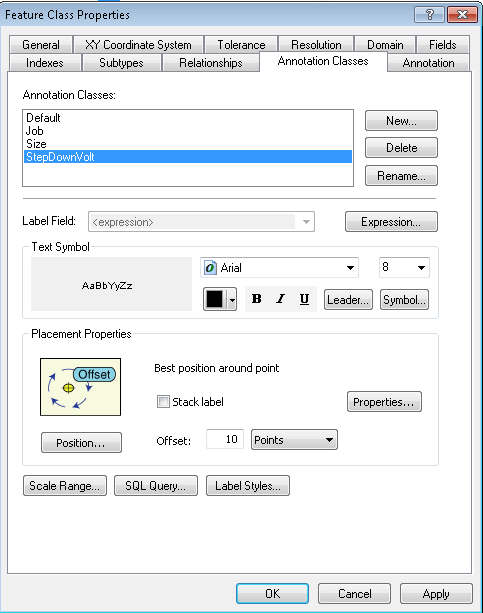
Expression:

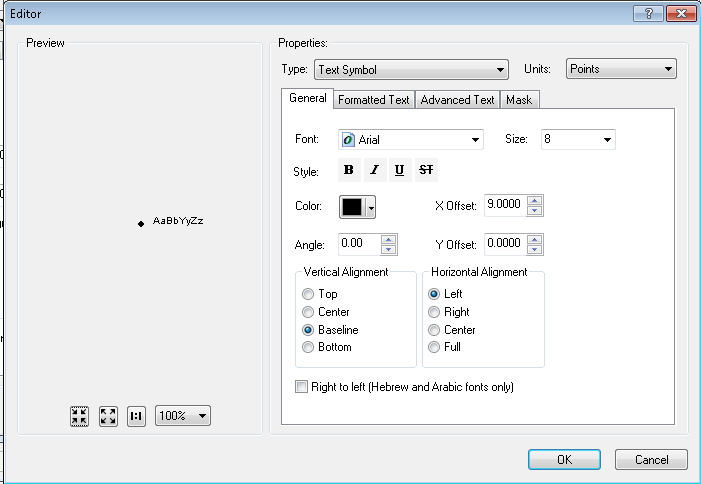
RATEDKVA

SQL Query:

RATEDKVA is not null

STEPDOWNVOLT Subclass is **NEW**





Expression:

Function FindLabel ( [OperatingVoltage], [OutputVoltage], [INSTALLATIONTYPE] )

myOplen = len([OperatingVoltage])

if myOplen > 1 then

myVolt = mid( [OperatingVoltage],myOplen-1,1)

else

myVolt = ""

end if

myOutlen = len([OutputVoltage])

if myOplen > 1 then

myOut = mid([OutputVoltage] ,myOutlen-1,1)

else

myOut = ""

end if

if myVolt = "k" then

myVolt = mid([OperatingVoltage],1,myOplen-3)

else

if myOplen > 2 then

myVolt = mid([OperatingVoltage],1,myOplen-2)

else

myVolt = [OperatingVoltage]

end if

end if

if myOut = "k" then

myOut = mid([OutputVoltage],1,myOutlen-3)

else

if myOutlen > 2 then

myOut = mid([OutputVoltage],1,myOutlen-2)

else

myOut = [OutputVoltage]

end if

end if

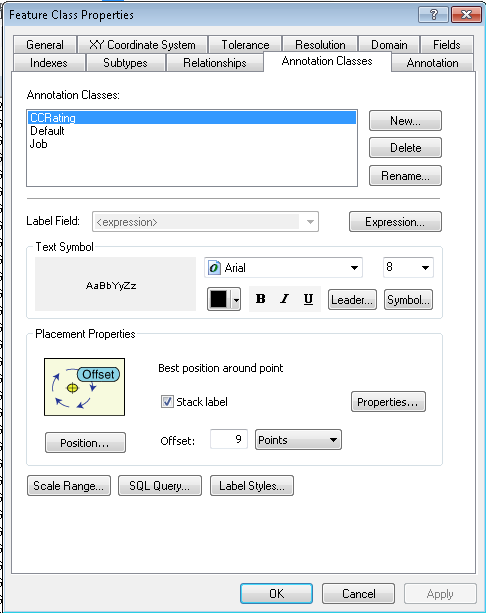
FindLabel = myVolt + "/" + myOut

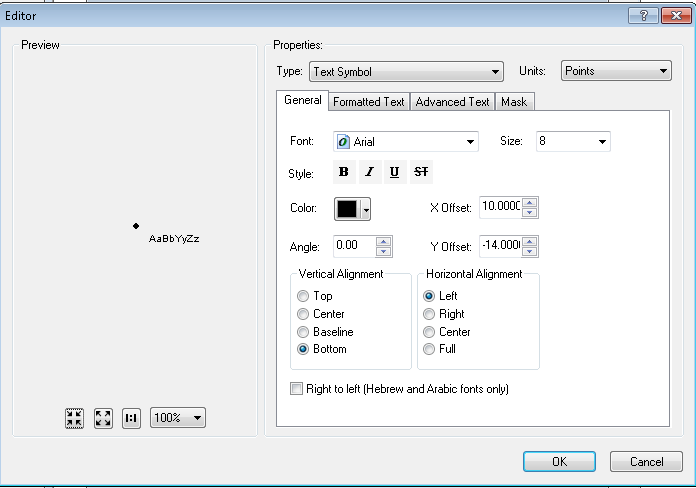
End Function

SQL Query:

**DO NOT PUT IN AN SQL QUERY**

**SWITCH**





Expression:

Function FindLabel ( [SubtypeCD] , [CCRating] )

If [SubtypeCD] = "Overhead Switch" and [CCRating] = "600A" then

FindLabel = [CCRating]

elseif [SubtypeCD] = "Overhead Switch" and [CCRating] = "800A" then

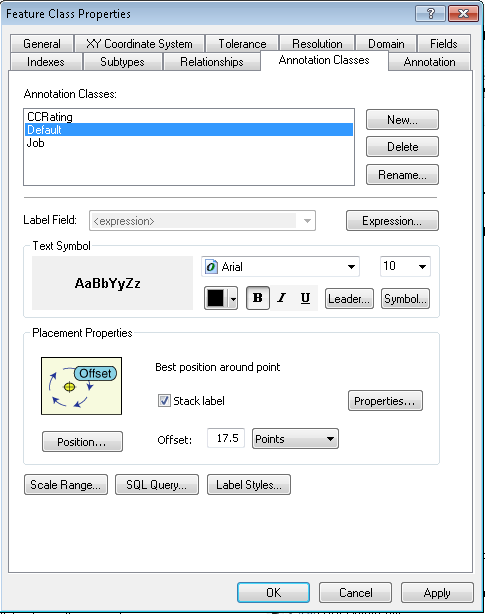
FindLabel = [CCRating]

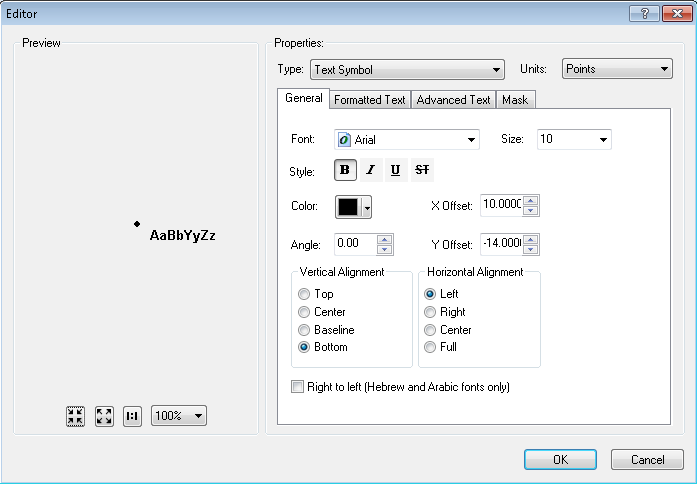
End if

End Function

SQL Query

**DO NOT PUT IN AN SQL QUERY**





Expression:

Function FindLabel ( [SubtypeCD], [OperatingNumber], [Class] ,[ComplexDeviceIdc] )

if [SubtypeCD] = "Overhead Disconnect" and [ComplexDeviceIdc] = "No" Then

FindLabel = [OperatingNumber] + chr(129) + vbCrLf + "SB" + vbCrLf + chr(129)

elseif [SubtypeCD] = "Overhead Switch" and [Class] = "II" Then

FindLabel = [OperatingNumber] + " II" + vbCrLf + chr(129)

else

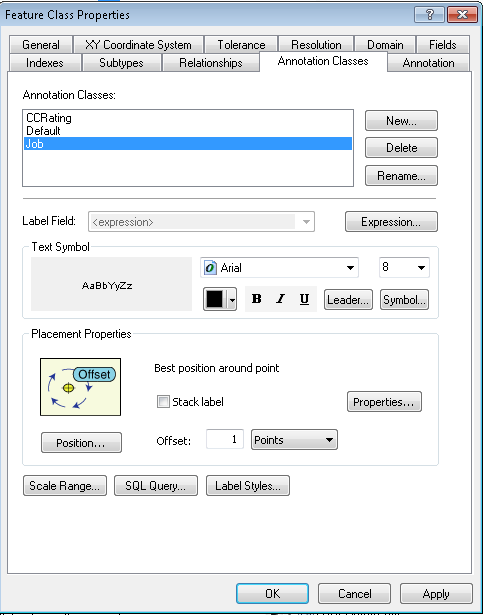
FindLabel = [OperatingNumber] + vbCrLf + chr(129)

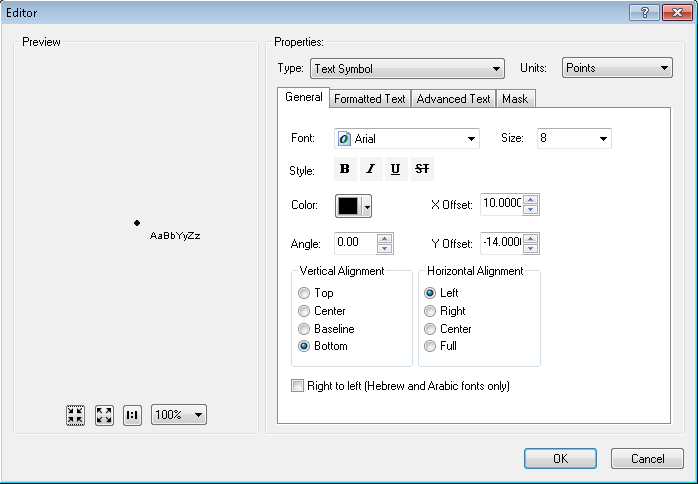
End if

End Function

SQL Query

**DO NOT PUT IN AN SQL QUERY**





Expression:

Function FindLabel ( [INSTALLJOBPREFIX] , [INSTALLJOBNUMBER] , [INSTALLJOBYEAR], [SUBTYPECD], [CCRATING] )

pad = vbCrLf + chr(129)

If [SubtypeCD] = "Overhead Switch" then

if [CCRating] = "600A" or [CCRating] = "800A" then

pad = ""

end if

End if

myString = ""

if isNull( [INSTALLJOBNUMBER] ) or [INSTALLJOBNUMBER] = "0" or \_

instr(ucase( [INSTALLJOBNUMBER] ), "CEDSA") then

else

if left(ucase( [INSTALLJOBPREFIX]), 3 ) <> "UNK" and \_

ucase( [INSTALLJOBPREFIX] ) <> "FIF" and \_

len( [INSTALLJOBPREFIX] ) > 0 then

myString = [INSTALLJOBPREFIX]

end if

if len( [INSTALLJOBYEAR] ) > 0 then

if len( [INSTALLJOBYEAR] ) = 4 then

FindLabel = myString + [INSTALLJOBNUMBER] + " '"+ right( [INSTALLJOBYEAR] ,2) + vbCrLf + chr(129) + pad

elseif len( [INSTALLJOBYEAR] ) = 2 then

FindLabel = myString + [INSTALLJOBNUMBER] + " '"+ [INSTALLJOBYEAR] + vbCrLf + chr(129) + pad

end if

else

FindLabel = myString + [INSTALLJOBNUMBER] + vbCrLf + chr(129) + pad

end if

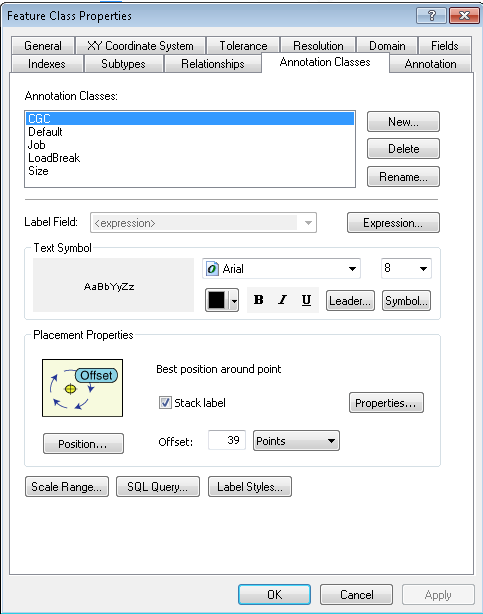
end if

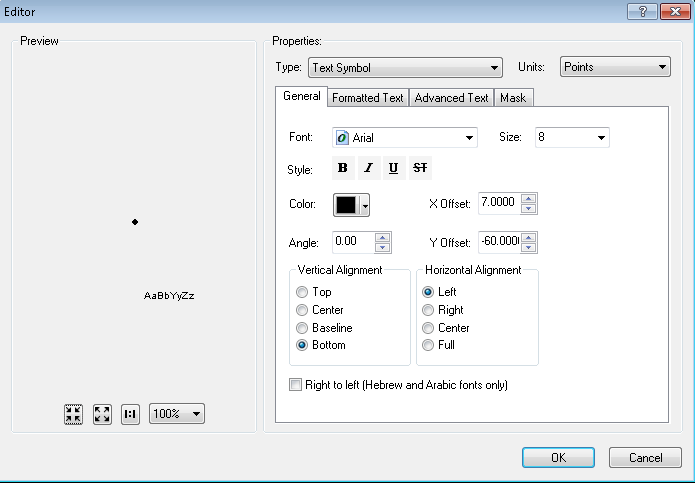
End Function

SQL Query:

**DO NOT PUT IN AN SQL QUERY**

**Transformer**





Expression:

Function FindLabel ( [SUBTYPECD], [CGC12], [INSTALLATIONTYPE], [LABELTEXT])

pos1 = InStr (1, [LABELTEXT], chr(10))

if pos1 > 0 then

pos2 = InStr(pos1 + 1, [LABELTEXT], chr(10))

else

pos2 = -1

end if

if ( pos1 > 0) and (pos2 > 0) and (pos2 - pos1 > 1) then

lineCount = 3

elseif pos1 > 0 then

lineCount = 2

else

lineCount = 1

end if

if [SUBTYPECD] = 1 or \_

([SUBTYPECD] = 4 and [INSTALLATIONTYPE] = "OH") or \_

([SUBTYPECD] = 7 and [INSTALLATIONTYPE] = "OH") or \_

([SUBTYPECD] = 8 and [INSTALLATIONTYPE] = "OH") then

myX = mid([CGC12],4,4)

myY = mid([CGC12],9,4)

if LineCount = 3 then

FindLabel = myX + "-" + myY

elseif LineCount = 2 then

FindLabel = myX + "-" + myY + vbCrLf + chr(129)

else

FindLabel = myX + "-" + myY + vbCrLf + chr(129) + vbCrLf + chr(129)

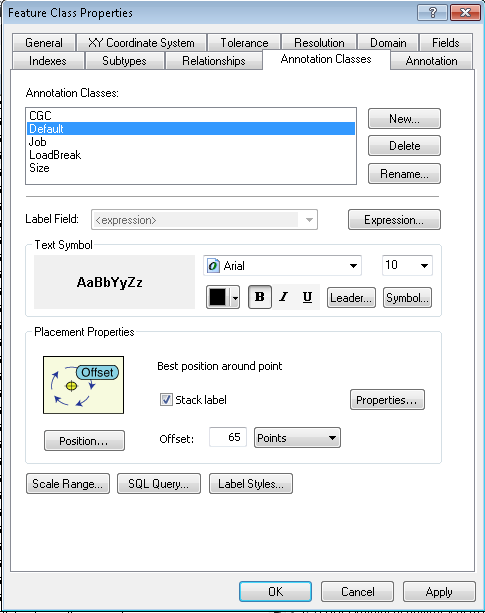
end if

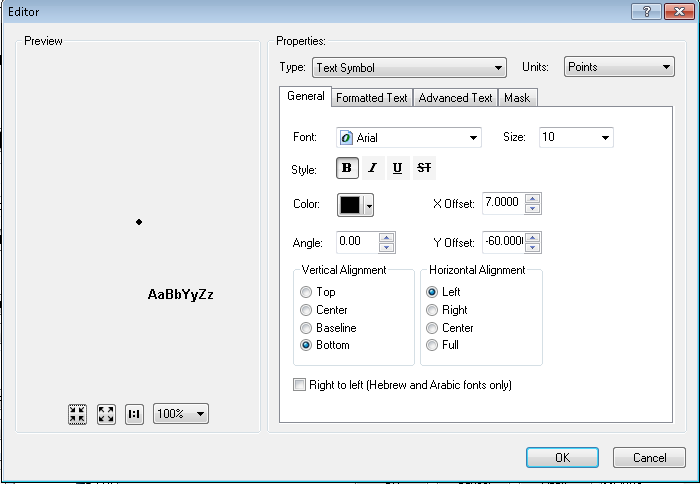
end if

End Function

SQL Query:

**DO NOT PUT IN AN SQL QUERY**





Expression:

Function FindLabel ( [OPERATINGNUMBER], [SUBTYPECD], [INSTALLATIONTYPE] )

if [SUBTYPECD] <> 1 then

if IsNull( [INSTALLATIONTYPE] ) or \_

[INSTALLATIONTYPE] <> "OH" then

FindLabel = [OPERATINGNUMBER]

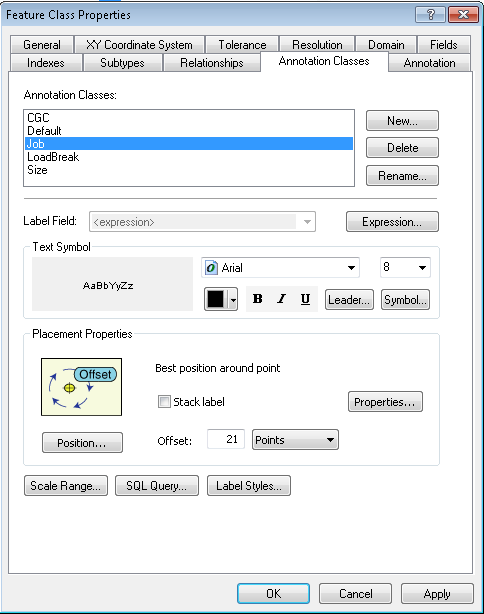
end if

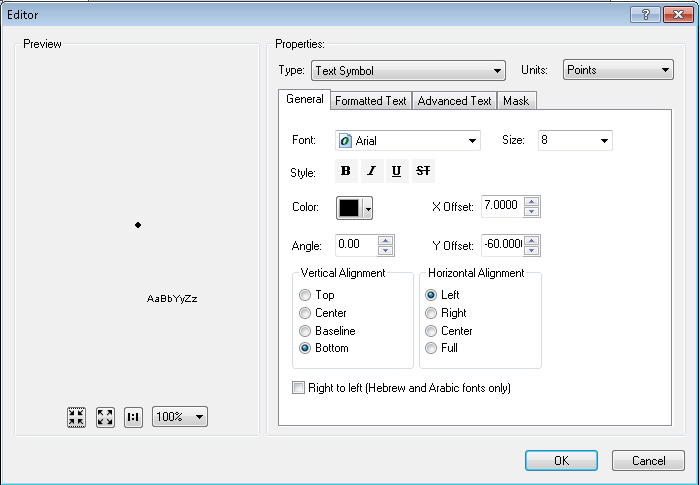
end if

End Function

SQL Query:

**DO NOT PUT IN AN SQL QUERY**





Expression:

Function FindLabel ( [InstallJobPrefix], [InstallJobNumber], [InstallJobYear], [LABELTEXT], [SUBTYPECD], [INSTALLATIONTYPE], [CGC12])

pos1 = InStr (1, [LABELTEXT], chr(10))

if pos1 > 0 then

pos2 = InStr(pos1 + 1, [LABELTEXT], chr(10))

else

pos2 = -1

end if

if ( pos1 > 0) and (pos2 > 0) and (pos2 - pos1 > 1) then

lineCount = 3

elseif pos1 > 0 then

lineCount = 2

else

lineCount = 1

end if

myString = ""

if isNull( [INSTALLJOBNUMBER] ) or [INSTALLJOBNUMBER] = "0" or \_

instr(ucase( [INSTALLJOBNUMBER] ), "CEDSA") then

else

if left(ucase( [INSTALLJOBPREFIX]), 3 ) <> "UNK" and \_

ucase( [INSTALLJOBPREFIX] ) <> "FIF" and \_

len( [INSTALLJOBPREFIX] ) > 0 then

myString = [INSTALLJOBPREFIX]

end if

if len( [INSTALLJOBYEAR] ) > 0 then

if len( [INSTALLJOBYEAR] ) = 4 then

myString = myString + [INSTALLJOBNUMBER] + " '"+ right( [INSTALLJOBYEAR] ,2)

elseif len( [INSTALLJOBYEAR] ) = 2 then

myString = myString + [INSTALLJOBNUMBER] + " '"+ [INSTALLJOBYEAR]

end if

else

myString = myString + [INSTALLJOBNUMBER]

end if

end if

if [SUBTYPECD] = 1 or \_

([SUBTYPECD] = 4 and [INSTALLATIONTYPE] = "OH") or \_

([SUBTYPECD] = 7 and [INSTALLATIONTYPE] = "OH") or \_

([SUBTYPECD] = 8 and [INSTALLATIONTYPE] = "OH") then

if LineCount = 3 then

FindLabel = myString + vbCrLf + chr(129)

elseif LineCount = 2 then

FindLabel = myString + vbCrLf + chr(129) + vbCrLf + chr(129)

else

FindLabel = myString + vbCrLf + chr(129) + vbCrLf + chr(129) + vbCrLf + chr(129)

end if

else

if LineCount = 3 then

FindLabel = myString + vbCrLf + chr(129)

elseif LineCount = 2 then

FindLabel = myString + vbCrLf + chr(129) + vbCrLf + chr(129)

else

FindLabel = myString + vbCrLf + chr(129) + vbCrLf + chr(129) + vbCrLf + chr(129)

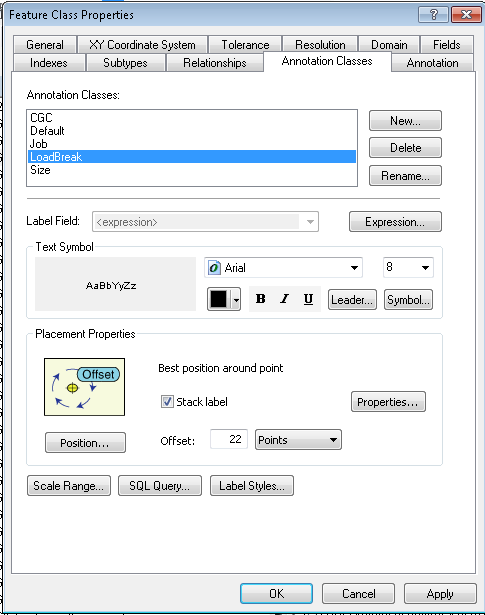
end if

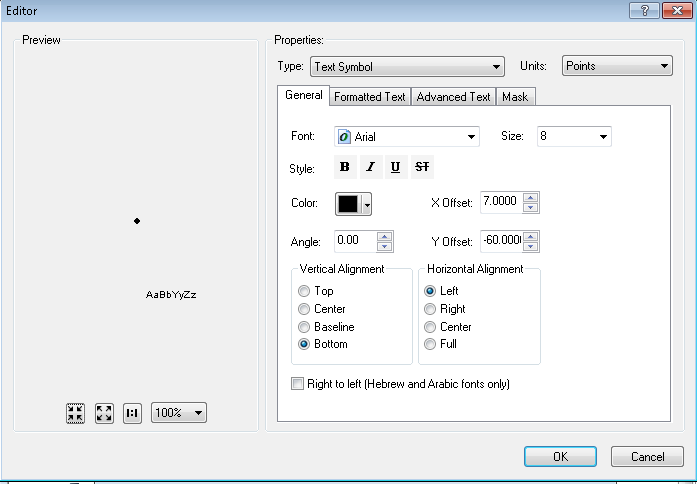
end if

End Function

SQL Query:

**DO NOT PUT IN AN SQL QUERY**





Expression:

Function FindLabel ( [SubtypeCD], [LiveFrontIdc], [LABELTEXT], [INSTALLATIONTYPE], [CGC12] )

If [SubtypeCD] = 2 or [SubtypeCD] = 3 then

pos1 = InStr (1, [LABELTEXT], chr(10))

if pos1 > 0 then

pos2 = InStr(pos1 + 1, [LABELTEXT], chr(10))

else

pos2 = -1

end if

if ( pos1 > 0) and (pos2 > 0) and (pos2 - pos1 > 1) then

lineCount = 3

elseif pos1 > 0 then

lineCount = 2

else

lineCount = 1

end if

if [LiveFrontIdc] = "Y" then

if LineCount = 1 then

FindLabel = "LF" + vbCrLf + chr(129) + vbCrLf + chr(129)

elseif LineCount = 2 then

FindLabel = "LF" + vbCrLf + chr(129)

else

FindLabel = "LF"

end if

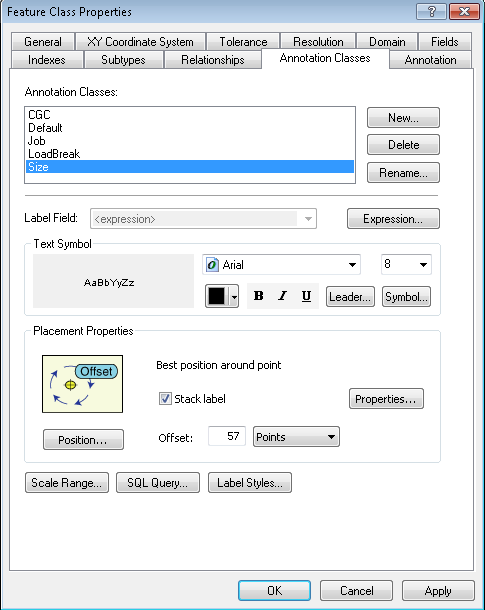
End if

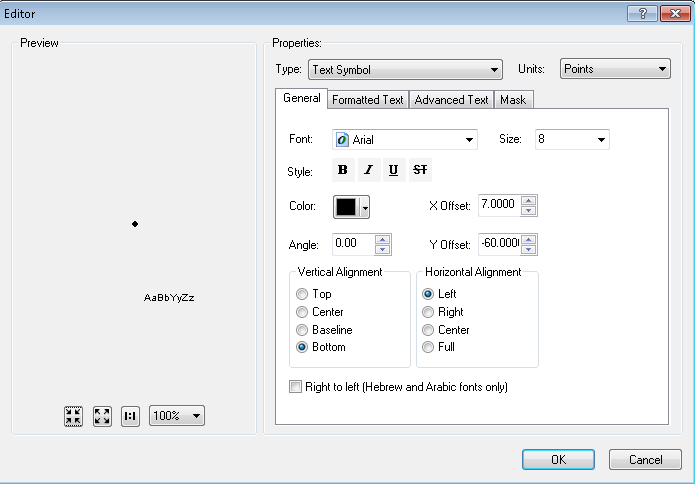
End If

End Function

SQL Query:

**DO NOT PUT IN AN SQL QUERY**





Expression:

Function FindLabel ( [LABELTEXT], [SUBTYPECD], [INSTALLATIONTYPE] )

if [SUBTYPECD] = 1 or \_

([SUBTYPECD] = 4 and [INSTALLATIONTYPE] = "OH") or \_

([SUBTYPECD] = 7 and [INSTALLATIONTYPE] = "OH") or \_

([SUBTYPECD] = 8 and [INSTALLATIONTYPE] = "OH") then

FindLabel = chr(129) + vbCrLf + [LABELTEXT] + vbCrLf + chr(129)

else

FindLabel = chr(129) + vbCrLf + [LABELTEXT]

end if

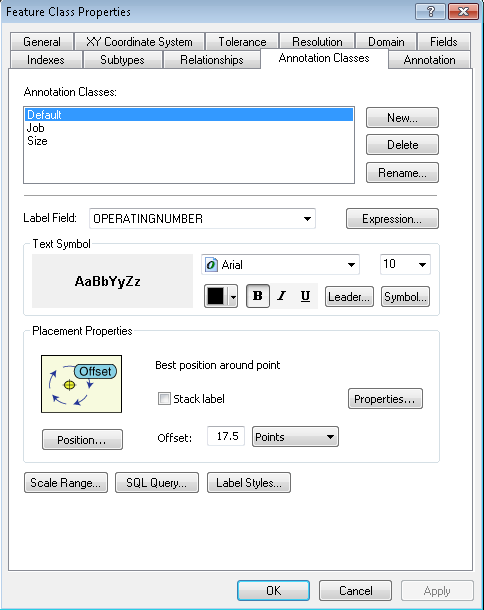
FindLabel = replace(FindLabel, chr(10) + " ", chr(10))

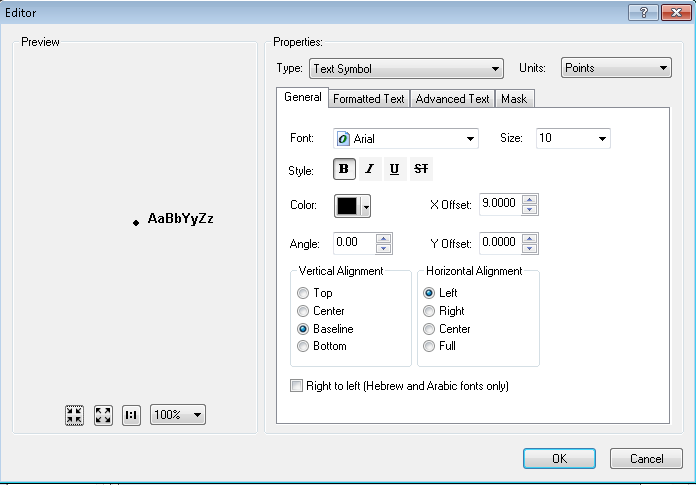
End Function

SQL Query:

**DO NOT PUT IN AN SQL QUERY**

**VoltageRegulator**



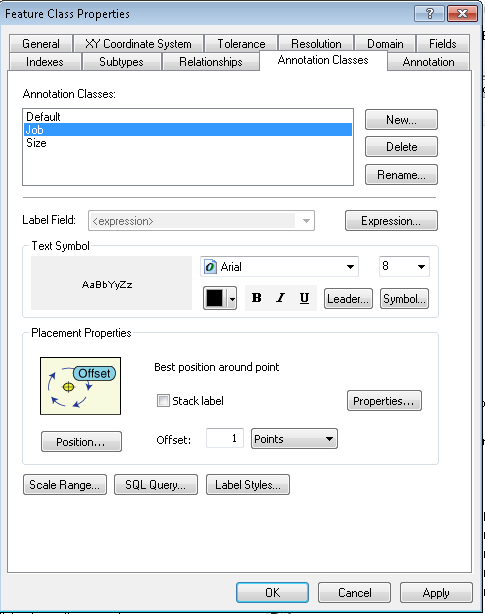


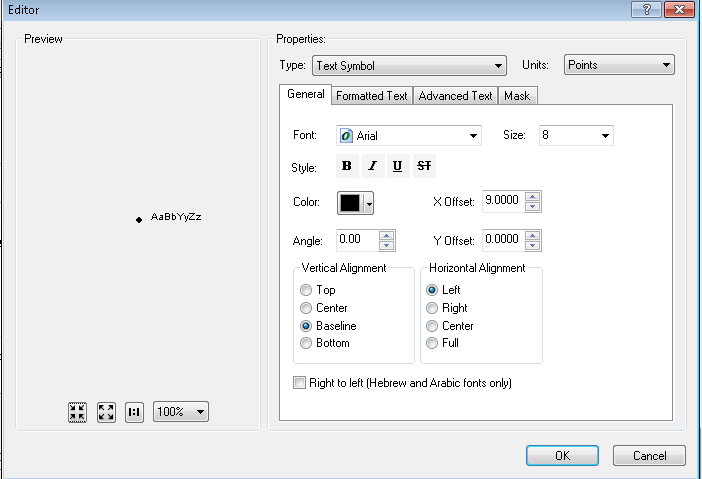
Expression:

OPERATINGNUMBER

SQL Query:

OPERATINGNUMBER is not null





Expression:

Function FindLabel ( [INSTALLJOBPREFIX], [INSTALLJOBNUMBER], [INSTALLJOBYEAR] )

if left(ucase( [INSTALLJOBPREFIX]), 3 ) <> "UNK" and \_

ucase( [INSTALLJOBPREFIX]) <> "FIF" and \_

len( [INSTALLJOBPREFIX]) > 0 then

myString = [INSTALLJOBPREFIX]

end if

if len( [INSTALLJOBYEAR]) > 0 then

if len( [INSTALLJOBYEAR]) = 4 then

FindLabel = myString + [INSTALLJOBNUMBER] + " '"+ right( [INSTALLJOBYEAR],2)

elseif len( [INSTALLJOBYEAR]) = 2 then

FindLabel = myString + [INSTALLJOBNUMBER] + " '"+ [INSTALLJOBYEAR]

end if

else

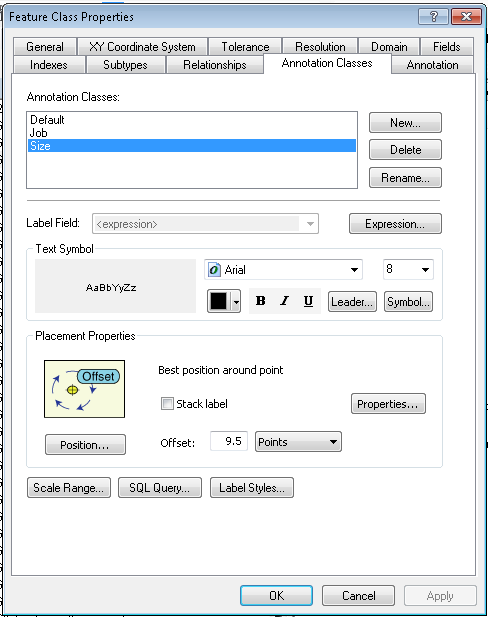
FindLabel = myString + [INSTALLJOBNUMBER]

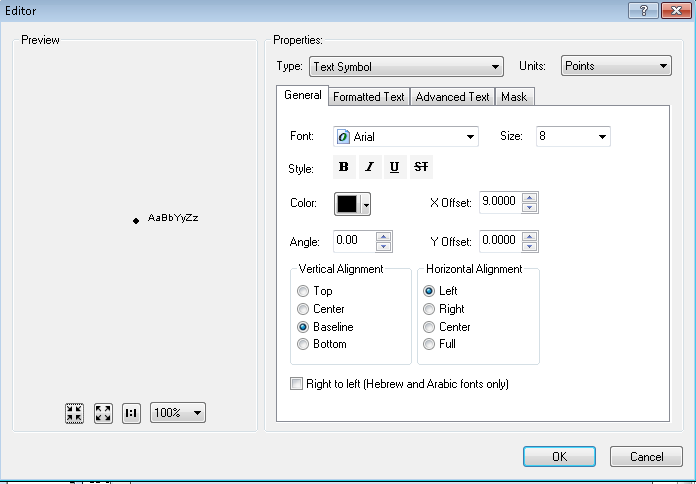
end if

End Function

SQL Query:

not (INSTALLJOBNUMBER is null or INSTALLJOBNUMBER = '0' or INSTALLJOBNUMBER like '%CEDSA%')





Expression:

Function FindLabel ([UnitCount], [RatedAmps] )

If [UnitCount]> 1 then

FindLabel = [UnitCount]& "-" & [RatedAmps]

else

Findlabel = [RatedAmps]

End if

End Function

SQL Query:

RATEDAMPS is not null

# Bug 9539: Move Tables From Common Features Dataset To Electric Dataset

1. Refer to the following feature classes when performing this step:

Job Feature Note

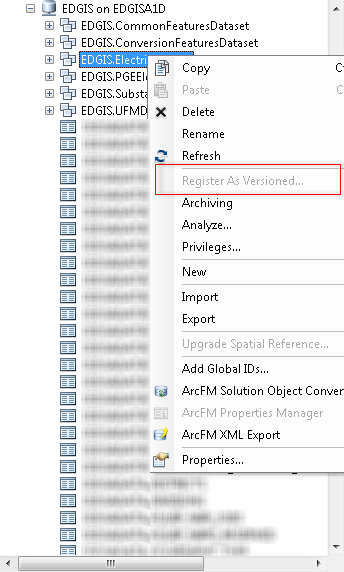
Job History Note

Job Number Anno

Job Number Anno Relationship

1. For each of the above feature classes, enter the Common Features Dataset, select the feature class, and drag and drop it into the Electric Dataset.

# Register the Dataset as Versioned

Right click the Electric Dataset in the target SDE database and select ‘Register As Versioned’  


# Update Data Model Version Table

**Database Configuration:**

1. Open SQL Plus.
2. Log in using the same server and user as was used in section 2.  
     
   
3. Run the SQL below:

update pgedatamodelversion set currentidc='N' where currentidc='Y';

insert into pgedatamodelversion (OBJECTID, CURRENTIDC, DATEAPPLIED, APPLIEDBYPERSONNAME, MODELVERSION) values (**32**,'Y',sysdate,'**<INSERT TEAM MEMBER DONE BY>**','**7.5.9** GOLD **CR9452**');

# Known Issues

<Please List any other issues encountered here while following the document>