ArcGIS MultiSpeak® Data Model

Data Dictionary

Last Revised: March 19, 2010

Copyright © 2010 ESRI
All rights reserved.
Printed in the United States of America.

The information contained in this document is the exclusive property of ESRI. This work is protected under United States copyright law and other international copyright treaties and conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted in writing by ESRI. All requests should be sent to Attention: Contracts and Legal Services Manager, ESRI, 380 New York Street, Redlands, California 92373-8100, USA.

The information contained in this document is subject to change without notice.

U.S. GOVERNMENT RESTRICTED/LIMITED RIGHTS

Any software, documentation, and/or data delivered hereunder is subject to the terms of the License Agreement. In no event shall the U.S. Government acquire greater than RESTRICTED/LIMITED RIGHTS. At a minimum, use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR §52.227-14 Alternates I, II, and III (JUN 1987); FAR §52.227-19 (JUN 1987) and/or FAR §12.211/12.212 (Commercial Technical Data/Computer Software); and DFARS §252.227-7015 (NOV 1995) (Technical Data) and/or DFARS §227.7202 (Computer Software), as applicable. Contractor/Manufacturer is ESRI, 380 New York Street, Redlands, California 92373-8100, USA.

@esri.com, 3D Analyst, ACORN, ADF, AML, ArcAtlas, ArcCAD, ArcCatalog, ArcCOGO, ArcData, ArcDoc, ArcEdit, ArcEditor, ArcEurope, ArcExplorer, ArcExpress, ArcGIS, ArcGlobe, ArcGrid, ArcIMS, ARC/INFO, ArcInfo, ArcInfo Librarian, ArcInfo—Professional GIS, ArcInfo—The World's GIS, ArcLocation, ArcLogistics, ArcMap, ArcNetwork, *ArcNews*, ArcObjects, ArcOpen, ArcPad, ArcPlot, ArcPress, ArcQuest, ArcReader, ArcScan, ArcScene, ArcSchool, ArcSDE, ArcSdl, ArcSketch, ArcStorm, ArcSurvey, ArcTIN, ArcToolbox, ArcTools, ArcUSA, *ArcUser*, ArcView, ArcVoyager, *ArcWatch*, ArcWeb, ArcWorld, ArcXML, Atlas GIS, AtlasWare, Avenue, Business Analyst Online, BusinessMAP, Community, CommunityInfo, Data Automation Kit, Database Integrator, DBI Kit, EDN, ESRI, ESRI—Team GIS, ESRI—The GIS Company, ESRI—The GIS People, ESRI—The GIS Software Leader, FormEdit, Geographic Design System, ESRI BIS, Geography Matters, Geography Network, GIS by ESRI, GIS Day, GIS for Everyone, GISData Server, JTX, MapBeans, MapCafé, MapData, MapObjects, Maplex, MapStudio, ModelBuilder, MOLE, NetEngine, PC ARC/INFO, PC ARCPLOT, PC ARCSHELL, PC DATA CONVERSION, PC STARTER KIT, PC TABLES, PC ARCEDIT, PC NETWORK, PC OVERLAY, PLTS, Rent-a-Tech, RouteMAP, SDE, Site·Reporter, SML, Sourcebook·America, Spatial Database Engine, StreetEditor, StreetMap, Tapestry, the ARC/INFO logo, the ArcAtlas logo, the ArcCAD logo, the ArcCAD WorkBench logo, the ArcCOGO logo, the ArcData logo, the ArcData Online logo, the ArcEdit logo, the ArcEurope logo, the ArcExplorer logo, the ArcExpress logo, the ArcGIS logo, the ArcGIS Explorer logo, the ArcGrid logo, the ArcIMS logo, the ArcInfo logo, the ArcLogistics Route logo, the ArcNetwork logo, the ArcPad logo, the ArcPlot logo, the ArcPress for ArcView logo, the ArcPress logo, the ArcScan logo, the ArcScene logo, the ArcSDE CAD Client logo, the ArcSDE logo, the ArcStorm logo, the ArcTIN logo, the ArcTools logo, the ArcUSA logo, the ArcView 3D Analyst logo, the ArcView Business Analyst logo, the ArcView Data Publisher logo, the ArcView GIS logo, the ArcView Image Analysis logo, the ArcView Internet Map Server logo, the ArcView logo, the ArcView Network Analyst logo, the ArcView Spatial Analyst logo, the ArcView StreetMap 2000 logo, the ArcView StreetMap logo, the ArcView Tracking Analyst logo, the ArcWorld logo, the Atlas GIS logo, the Avenue logo, the BusinessMAP logo, the Community logo, the Data Automation Kit logo, the Digital Chart of the World logo, the ESRI Data logo, the ESRI globe logo, the ESRI Press logo, The Geographic Advantage, the Geography Network logo, the MapCafé logo, the MapObjects Internet Map Server logo, the MapObjects logo, the MOLE logo, the NetEngine logo, the PC ARC/INFO logo, the Production Line Tool Set logo, the RouteMAP IMS logo, the RouteMAP logo, the SDE logo, The World's Leading Desktop GIS, *Water Writes*, www.esri.com, www.esribis.com, www.geographynetwork.com, www.gis.com, www.gisday.com, and Your Personal Geographic Information System are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions.

Other companies and products mentioned in the ESRI Web site may be trademarks or registered trademarks of their respective trademark owners.

Table of Contents

1.0 Introduction ... 1
2.0 Data Model Classes ... 3
2.1 Abstract Classes ... 5
2.1.1 *MConnectivityLine* ... 5
2.1.2 *MConnectivityPoint* ... 5
2.1.3 *MDeviceObject* ... 6
2.1.4 *MElectricBankFeature* ... 6
2.1.5 *MElectricConductor* ... 7
2.1.6 *MElectricLineFeature* ... 7
2.1.7 *MElectricMotorGenerator* ... 8
2.1.8 *MElectricObject* ... 8
2.1.9 *MElectricOvercurrentDevice* ... 9
2.1.10 *MElectricPointFeature* ... 10
2.1.11 *MElectricSwitchingBankFeature* ... 10
2.1.12 *MElectricSwitchingDevice* ... 10
2.1.13 *MLineFeature* ... 11
2.1.14 *MPointFeature* ... 11
2.1.15 *MUGElectricLine* ... 12
2.2 Conductors, Network Devices and Administration ... 13
2.2.1 Breaker ... 13
2.2.2 Capacitor ... 13
2.2.3 CapacitorBank ... 13
2.2.4 Customer ... 14
2.2.5 Fuse ... 15
2.2.6 Generator ... 15
2.2.7 Meter ... 16
2.2.8 MeterBase ... 16
2.2.9 Motor ... 17
2.2.10 OHPrimaryLine ... 18
2.2.11 OHSecondaryLine ... 18
2.2.12 OvercurrentDeviceBank ... 18
2.2.13 Recloser ... 19
2.2.14 Regulator ... 19
2.2.15 RegulatorBank ... 20
2.2.16 Sectionalizer ... 20
2.2.17 ServiceLocation ... 21
2.2.18 Streetlight ... 21
2.2.19 Substation ... 22
2.2.20 Switch ... 23
2.2.21 SwitchDeviceBank ... 23
2.2.22 Transformer ... 23
2.2.23 TransformerBank ... 24
2.2.24 UGPrimaryLine ... 25

2.2.25 UGSecondaryLine ... 25
2.3 Surface and Overhead Structures ... 26
2.3.1 Assembly ... 26
2.3.2 Guy ... 27
2.3.3 JointUse ... 27
2.3.4 Pole ... 28
2.3.5 PrimaryCabinet ... 28
2.3.6 Riser ... 29
2.3.7 SecondaryJunctionBox ... 29
2.3.8 SpanGuy ... 29
3.0 Domains and Relationships ... 30
3.1 Domains ... 30
3.1.1 eDomainAssemblyAction ... 30
3.1.2 eDomainAssemblySpanType ... 30
3.1.3 eDomainBooleanYesNo ... 31
3.1.4 eDomainConnectionType ... 31
3.1.5 eDomainConnectionTypeWyeDelta ... 31
3.1.6 eDomainDeviceMounting ... 32
3.1.7 eDomainGeneratorModel ... 32
3.1.8 eDomainGuyAgainst ... 32
3.1.9 eDomainGuyStatus ... 33
3.1.10 eDomainGuyType ... 33
3.1.11 eDomainImpedanceUnits ... 33
3.1.12 eDomainMeterBaseType ... 34
3.1.13 eDomainMotorStatus ... 34
3.1.14 eDomainMSPObjectAction ... 35
3.1.15 eDomainNEMATypeCode ... 35
3.1.16 eDomainPhase ... 36
3.1.17 eDomainPoleClass ... 36
3.1.18 eDomainPoleTreatmentType ... 37
3.1.19 eDomainPoleType ... 37
3.1.20 eDomainPoleUseType ... 38
3.1.21 eDomainPosition ... 38
3.1.22 eDomainPowerStatus ... 39
3.1.23 eDomainSecondaryType ... 39
3.1.24 eDomainSinglePhase ... 39
3.1.25 eDomainSoftStartTypeCodes ... 40
3.1.26 eDomainSourceSideConfig ... 40
3.1.27 eDomainSwitchStatus ... 40
3.1.28 eDomainSwitchType ... 41
3.1.29 eDomainTransformerWindingCode ... 41
3.1.30 eDomainYesNo ... 42
3.2 Relationships ... 43
3.2.1 CapacitorBankHasCapacitor ... 43
3.2.2 CustomerHasServiceLocation ... 43
3.2.3 MeterBaseHasMeter ... 43

3.2.4 OvercurrentDeviceBankHasBreaker 44
3.2.5 OvercurrentDeviceBankHasFuse 44
3.2.6 OvercurrentDeviceBankHasRecloser 44
3.2.7 OvercurrentDeviceBankHasSectionalizer 44
3.2.8 PoleHasAssembly 45
3.2.9 PoleHasJointUse 45
3.2.10 RegulatorBankHasRegulator 45
3.2.11 ServiceLocationHasMeterBase 46
3.2.12 SwitchDeviceBankHasSwitch 46
3.2.13 TransformerBankHasTransformer 46
4.0 Review and Testing Notes 47
Appendix A – ArcGIS Data Types 49

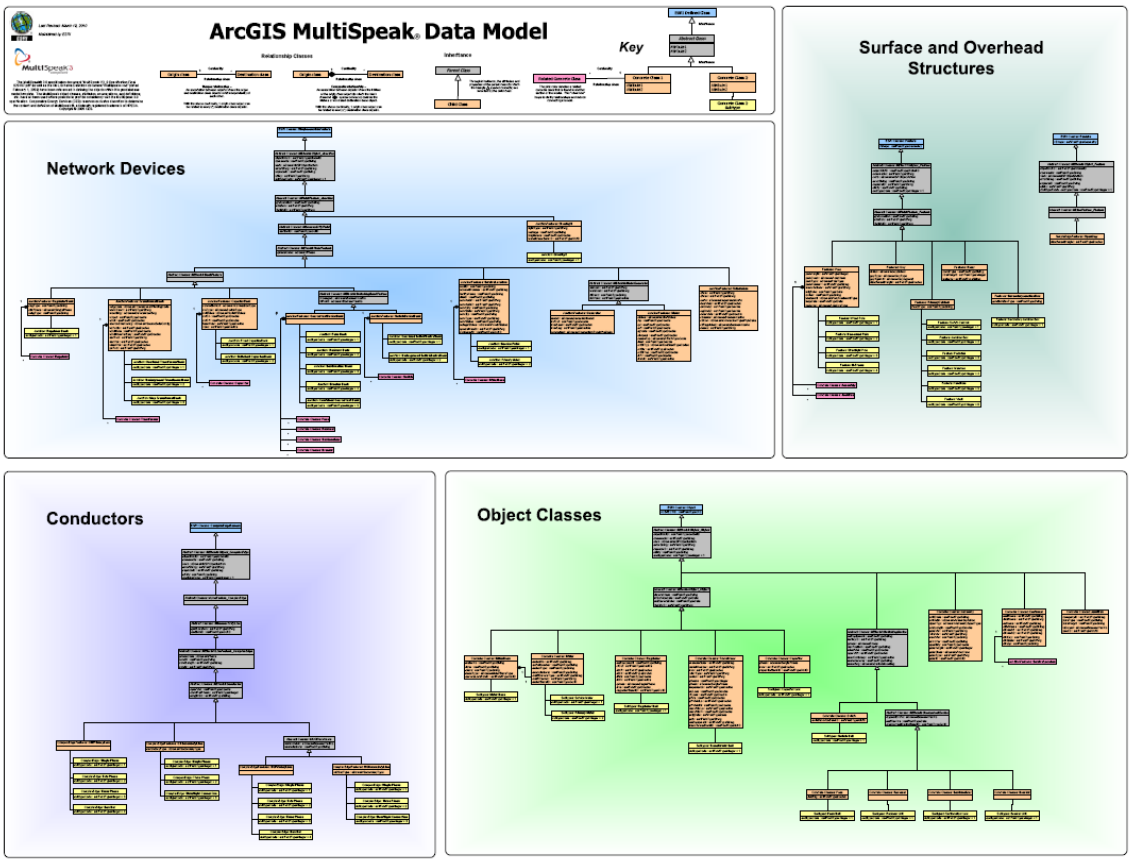
1.0 Introduction

The ArcGIS MultiSpeak® Data Model is a ready-to-use geodatabase data model template that closely conforms to the MultiSpeak® 3.0 specification, which is an industry-wide software standard that facilitates interoperability of diverse business and automation applications used in electric utilities. The ArcGIS MultiSpeak® Data Model template can be configured and customized to meet the needs of ESRI users that design and maintain electric networks and associated infrastructure, and will be of particular benefit to those that are planning to implement a MultiSpeak® 3.0 interface with their GIS.

The MultiSpeak® 3.0 specification document "MultiSpeak V3_0 Specification Final 020107.pdf" as well as the XML Schema Definition document "multispeakx.xsd" (dated February 1, 2009) have been referenced in defining the objects within this geodatabase model template. The MultiSpeak object classes, attributes, enumerations, and definitions, etc. have all been used where possible to provide consistency with the MultiSpeak 3.0 specification. Cooperative Energy Services (CES) reserves exclusive discretion to determine the content and definition of MultiSpeak ®, a federally registered trademark of NRECA. Copyright © *2000 CES.*

Where possible, the model template has been designed to conform to the MultiSpeak® 3.0 specification, and the template has been limited to those network classes which would commonly be maintained within a user's geodatabase. This data dictionary document provides an overview of the template's data elements, including all of the abstract and concrete classes, with basic descriptions of each, and their respective attributes and domains where applicable. It is meant to be used in conjunction with the physical model UML and/or PDF diagram, as seen below in Figure 1.1, which depicts the object inheritance structure which the model uses. In addition to class descriptions, an overview of the template's relationship classes and domains is provided in this dictionary document as well.

Figure 1.1 ArcGIS MultiSpeak® Data Model Diagram



For further information on the MultiSpeak® 3.0 specification and details, users of this template can consult the primary documents which are listed above, through the MultiSpeak website at: http://www.multispeak.org/

For further information on geodatabase modeling concepts, the following ESRI Press books are recommended:

- *Modeling our World: The ESRI Guide to Geodatabase Design*, by Michael Zeiler.
  http://esripress.esri.com/display/index.cfm?fuseaction=display&websiteID=27

- *Designing Geodatabases: Case Studies in GIS Data Modeling*, by David Arctur and Michael Zeiler.
  http://esripress.esri.com/display/index.cfm?fuseaction=display&websiteID=85

2.0 Data Model Classes

As with other ESRI data model templates, and with the MultiSpeak® 3.0 specification, the ArcGIS MultiSpeak® Data Model leverages object inheritance to reduce redundant listings of attributes within the model and clarify the data elements. In this section, each of the abstract and concrete object classes within the model will be addressed and described.

The format for listing the specifications for each class in the model will include the following items:

- Type – Whether it is an abstract or concrete class, and if concrete, whether it is a feature or object class.
- Inherits From – What abstract classes the class inherits from in the model.
- Geometry – If it is a feature class, the geometry type.
- Attributes – The attributes for the class, with their respective aliases, data types, and attribute descriptions (as described further below).
- Subtypes – The subtypes for the class and their descriptions.
- Relationships – The relationship classes which the class participates in.
- Notes – Supplemental information where applicable.

Using the review items listed above, the following is the table structure which will be used to provide information about each class. Additional information is provided about the attribute definitions.

Class Name	***<Name>***
Type	
Inherits From	
Geometry	
Attributes	**attributeName (the attribute name is in bold)** – *Attribute Alias (the attribute alias is in italics)* – <text, 255> (data type is in "< >" brackets) **-** This is the definition of the attribute (a high-level definition is provided at the end, and if the attribute references a domain, then the domain name is provided which can be looked up in Section 3.1). Example: **comments –** *Comments* - <text, 255> - User or Designer Comments.
Subtypes	
Relationships	
Notes	

Below are some further design considerations regarding the model template:

- It is important to understand the differences between abstract and concrete classes within the model. The abstract classes defined in the modeling process do not have a direct representation in the geodatabase. They do serve an important purpose in that they contain attributes which are common to many tables or feature classes in a single location, and which can be inherited by other "child" classes. The abstract classes have direct equivalents within the Physical UML document which accompanies this data dictionary and can be referred to better understand the inheritance. When the model is then used to create a geodatabase schema, the attributes in the "parent" abstract classes are then included in the child classes, which may also be another abstract class, and which are all inherited by concrete geodatabase feature classes and object classes.

- The MultiSpeak® 3.0 specification includes classes which have both required and optional attributes, with most attributes being optional. The user could consult the specification and refine the model template as needed to remove any non-required attributes which are not seen as necessary to the user's business processes. It should be noted that any new attributes which are added to the model would need to be addressed with an additional revision to the MultiSpeak® 3.0 interface specification. To support key geodatabase functionality for subtypes, the attribute "subtypeCode" will be included in most all concrete classes. This is the only consistent example of an attribute added in the model template which extends outside the MultiSpeak® 3.0 specification.

- Geodatabase domains have only been provided when there was an enumeration that was found within the MultiSpeak® 3.0 specification. This being said, there is ample room for the model template to be extended to accommodate new domain lists which may fit the user's business processes. A key point is that the data type for the domain values should accommodate the default values if the user does not wish to extend their MultiSpeak® 3.0 specification.

- All text attributes throughout the model template have been set to a length of 255 to accommodate the maximum allowable by the MultiSpeak® 3.0 specification. Users may choose to constrain the length of these attributes to whatever works best to meet their data management requirements.

2.1 Abstract Classes

This section is comprised of descriptions of all the abstract classes in the model. For more information on abstract classes, see the introduction to Section 2.0.

2.1.1 *MConnectivityLine*

MConnectivityLine is an abstract class which provides attributes unique to line features which require maintaining connectivity information in the electric network. *MConnectivityLine* provides attributes which are introduced in the MultiSpeak specification's *mspConnectivityLine* abstract object class.

Class Name	***MConnectivityLine***
Type and Purpose	Abstract class
Inherits From	*MLineFeature*
Geometry	Not applicable
Attributes	**gridLocation** - *Grid Location* - <text, 255> - This is a utility-assigned location identifier that may or may not be tied to any GIS coordinate system. Sometimes referred to by the utility as "map number" or "grid number". **sectionID** – *Section ID* - <GUID> -- This is a name for this section, it is represented by the value of UID. This attribute is used in the case of a line sectional model, as opposed to a nodal model.
Subtypes	Not applicable
Relationships	Not applicable
Notes	

2.1.2 *MConnectivityPoint*

MConnectivityPoint is an abstract class which provides attributes unique to point features which require maintaining connectivity information in the electric network. *MConnectivityPoint* provides attributes which are introduced in the MultiSpeak specification's *mspConnectivityPoint* abstract object class.

Class Name	***MConnectivityPoint***
Type and Purpose	Abstract class
Inherits From	*MPointFeature*
Geometry	Not applicable
Attributes	**sectionID** – *Section ID* - <GUID> -- This is a name for this section, it is represented by the value of UID. This attribute is used in the case of a line sectional model, as opposed to a nodal model.
Subtypes	Not applicable
Relationships	Not applicable
Notes	

2.1.3 *MDeviceObject*

MDeviceObject is an abstract class which provides attributes for the meter device object class. *MDeviceObject* provides attributes which are introduced in the MultiSpeak specification's *mspDevice* abstract object class.

Class Name	***MDeviceObject***
Type and Purpose	Abstract class
Inherits From	*MElectricObject*
Geometry	Not applicable
Attributes	**deviceClass** – *Device Class* - <text, 255> - A high-level description of this type of object (e.g., "kWh meter", "demand meter", etc.). **inServiceDate** – *In-Service Date* - <date> – The date and time that a device was placed into active service. **outServiceDate** – *Out-Service Date* - <date> - The date and time that a device was removed from active service. **facilityID** – *Facility ID* - <text, 255> - A utility-defined string designation for this device.
Subtypes	Not applicable
Relationships	Not applicable
Notes	

2.1.4 *MElectricBankFeature*

MElectricBankFeature is an abstract class which provides attributes for point feature classes representing electric point bank features which contain individual unit objects, which are related to the bank features. Although it does not introduce any attributes, *MElectricBankFeature* is meant to help guide users back to the MultiSpeak structure, and also users can choose to extend and add attributes as needed from the template and base specification. *MElectricBankFeature* provides attributes which are introduced in the MultiSpeak specification's *mspBankObject* abstract object class.

Class Name	***MElectricBankFeature***
Type and Purpose	Abstract class
Inherits From	*MElectricPointFeature*
Geometry	Not applicable
Attributes	***All attributes are inherited.***
Subtypes	Not applicable
Relationships	Not applicable
Notes	

2.1.5 *MElectricConductor*

MElectricConductor is an abstract class which provides common attributes for all electric conductor features. There is no equivalent abstract class in the MultiSpeak specification, this class is provided for consistent attributes found in each MultiSpeak overhead and underground conductor class.

Class Name	***MElectricConductor***
Type and Purpose	Abstract class
Inherits From	*MElectricLineFeature*
Geometry	Not applicable
Attributes	**operVolt** – *Operating Voltage* - <double> - Operating voltage. **condPerPhase** – *Conductors Per Phase* - <integer> -- Number of conductors per phase. **lengthSrc** – *Length Source* - <text, 255> - Metadata about source of length information, used to determine accuracy.
Subtypes	Not applicable
Relationships	Not applicable
Notes	

2.1.6 *MElectricLineFeature*

MElectricLineFeature is an abstract class which provides attributes for electric line features which participate in the electric network, with phase and conductor information. *MElectricLineFeature* provides attributes which are introduced in the MultiSpeak specification's *mspElectricLine* abstract object class.

Class Name	***MElectricLineFeature***
Type and Purpose	Abstract class
Inherits From	*MLineFeature*
Geometry	Not applicable
Attributes	**phaseCode** – *Phase Code* - <text, 255> - The phase which the inheriting electric device is on. Uses the *eDomainPhase* domain. **condN** – *Conductor Neutral* - <text, 255> - Description of neutral conductor used in this section. **condLength** – *Conductor Length* - <text, 255> - Length of conductor in section. *Note: This will normally be expressed in the coordinate system units.* **constr** – *Construction* - <text, 255> - Description of construction type used in section.
Subtypes	Not applicable
Relationships	Not applicable
Notes	

2.1.7 *MElectricMotorGenerator*

MElectricMotorGenerator is an abstract class which provides attributes for motor and generator features. *MElectricMotorGenerator* provides attributes which are introduced in the MultiSpeak specification's *mspMotorGenerator* abstract object class.

Class Name	***MElectricMotorGenerator***
Type and Purpose	Abstract class
Inherits From	*MElectricPointFeature*
Geometry	Not applicable
Attributes	**ssDesc** – *Steady-state Device Description* - <text, 255> - Description of steady-state device type. **tranDesc** – *Transient Device Description* - <text, 255> - Description of transient device type. **stDesc** – *Subtransient Device Description* - <text, 255> - Description of subtransient device type. **rtdVolts** – *Rated Voltage in kV* - <double> - Rated voltage in kV.
Subtypes	Not applicable
Relationships	Not applicable
Notes	

2.1.8 *MElectricObject*

MElectricObject is the highest level abstract class, providing attributes which are inherited by every object in the data model. *MElectricObject* provides attributes which are introduced in the MultiSpeak specification's *mspObject* abstract object class.

Class Name	***MElectricObject***
Type	Abstract class
Inherits From	Not Applicable
Geometry	Not applicable
Attributes	**objectGUID** – *Object GUID* - <GlobalID> - A Globally Unique Identifier (GUID) value for the object. Global ID and GUID data types store registry style strings consisting of 36 characters enclosed in curly brackets (e.g., {098f2470-bae0-11cd-b579-08002b30bfeb}). These strings uniquely identify a feature or table row within a geodatabase and across geodatabases. In all relationships, the SDSID with a Global ID data type is the origin key, and so a GUID data type field will always be the foreign key datatype. Note: Equivalent in MultiSpeak is objectID. **comments –** *Comments* - <text, 255> - User or Designer Comments. **verb** – *Verb* - <text, 255> - Verbs identify what action the client is to take in response to receiving the data object. See Section 3.1 of the MultiSpeak 3.0 specification for a discussion of the use of these verbs in MultiSpeak messages. References the *eDomainMSPObjectAction* domain. **errorString** – *Error String* - <text, 255> - This attribute provides the capability to pass an error message string. **replaceID** – *Replace ID* - <text, 255> - This attribute is used in the event

	that a data package is sent indicating that the client system shall replace one instance of a data object with another. For instance, if a message directs the client to replace an existing meter with a new meter, the replaceID attribute will carry the objectID of the meter that is to be replaced by the new item. **utility** – *Utility* - <text, 255> - This attribute may be used to carry information about to which utility the object belongs. This attribute may be important where information generated by multiple utilities is used by one software application. **subtypeCode** – *Subtype Code* - <integer> - a subset identifier for features in a feature class or objects in a table that share the same attributes. List of available subtypes for the feature or object class, if applicable, will be listed for this attribute, with associated subtype codes.
Subtypes	Not applicable
Relationships	Not applicable
Notes	Abstract class used for the storage of foundational attributes. As this class is at the base of all classes, it is replicated for all feature types, and for geometric network participants – e.g., *MElectricObject _ComplexEdge, MElectricObject_Junction, MElectricObject_Feature and MElectricObject_Object.*

2.1.9 *MElectricOvercurrentDevice*

MElectricOvercurrentDevice is an abstract class which provides attributes for overcurrent devices which require bypass information. *MElectricOvercurrentDevice* provides attributes which are introduced in the MultiSpeak specification's *mspOvercurrentDevice* abstract object class. *MElectricOvercurrentDevice* also provides a common attribute "overcurrentDeviceBankID" which is a foreign key attribute found in all inheriting concrete classes.

Class Name	***MElectricOvercurrentDevice***
Type and Purpose	Abstract class
Inherits From	*MElectricSwitchingDevice*
Geometry	Not applicable
Attributes	**bypassExists** – *Bypass Exists* - <integer> Bypass switch exists at this location (yes or no)? Uses the *eDomainBooleanYesNo* domain. **lastService** – *Date of Last Service* - <date> - Date of last service. **overcurrentDeviceBankID** – *Overcurrent Device Bank ID* - <GUID> - The foreign key to the related Overcurrent Bank feature. Applies to Fuse, Recloser, Breaker and Sectionalizer objects.
Subtypes	Not applicable
Relationships	Not applicable
Notes	

2.1.10 *MElectricPointFeature*

MElectricPointFeature is an abstract class which provides attributes for electric point features which participate in the electric network, with phase and load information. *MElectricPointFeature* provides attributes which are introduced in the MultiSpeak specification's *mspElectricPoint* abstract object class.

Class Name	***MElectricPointFeature***
Type and Purpose	Abstract class
Inherits From	*MConnectivityPoint*
Geometry	Not applicable
Attributes	**phaseCode** – *Phase Code* - <text, 255> - The phase which the inheriting electric device is on. Uses the *eDomainPhase* domain.
Subtypes	Not applicable
Relationships	Not applicable
Notes	

2.1.11 *MElectricSwitchingBankFeature*

MElectricSwitchingBankFeature is an abstract class which provides attributes for switching bank features. *MElectricSwitchingBankFeature* provides attributes which are introduced in the MultiSpeak specification's *mspSwitchingBank* abstract object class.

Class Name	*MElectricSwitchingBankFeature*
Type and Purpose	Abstract class
Inherits From	*MElectricBankFeature*
Geometry	Not applicable
Attributes	**isGanged** – *Is Ganged? (Y/N)* - <integer> - Whether the switching bank works together as a bank. Will be "Yes" if all phases contain similar devices that work together as a unit. Uses the *eDomainBooleanYesNo* domain. **ldPoint** – *Load Point? (Y/N)* - <integer> - Whether this is a load allocation control point. Uses the *eDomainBooleanYesNo* domain.
Subtypes	Not applicable
Relationships	Not applicable
Notes	

2.1.12 *MElectricSwitchingDevice*

MElectricSwitchingDevice is an abstract class which provides attributes for objects which will contain switching specific fields. *MElectricSwitchingDevice* provides attributes which are introduced in the MultiSpeak specification's *mspSwitchingDevice* abstract object class.

Class Name	***MElectricSwitchingDevice***
Type and Purpose	Abstract class
Inherits From	*MElectricObject*
Geometry	Not applicable
Attributes	**eaEquipment** – *EA Equipment* - <text, 255> - Description for the type of regulator in this section. **facilityID** – *Facility ID* - <text, 255> - Transformer Number (company number). **phase** – *Phase* - <text, 255> - The phase which the inheriting electric device is on. Uses the *eDomainPhase* domain. **devPosition** – *Position* - <text, 255> - The device position. Uses the *eDomainPosition* domain. ***Note**: "position" is an SQL reserved word and so it cannot be used as an attribute name, and so the name has been updated to "devPosition".* **ratedVolt** - *Rated Voltage* - <double> - Nominal system voltage. **operVolt** - *Operating Voltage* - <double> - Operating voltage. **maxContAmp** - *Maximum Continuous Amps* - <double> - Maximum continuous current rating in amps. **manufacturer** – *Manufacturer* - <text, 255> - Manufacturer. **mounting** – *Mounting* - <text, 255> - Description of the switching device's mounting. Uses the *eDomainDeviceMounting* domain.
Subtypes	Not applicable
Relationships	Not applicable
Notes	

2.1.13 *MLineFeature*

MLineFeature is an abstract class which provides attributes unique to line features. *MLineFeature* equates to the MultiSpeak specification's *mspLineObject*.

Class Name	***MLineFeature***
Type and Purpose	Abstract class
Inherits From	*MElectricObject*
Geometry	Not applicable
Attributes	***All attributes are inherited.***
Subtypes	Not applicable
Relationships	Not applicable
Notes	

2.1.14 *MPointFeature*

MPointFeature is an abstract class which provides attributes unique to point features. *MPointFeature* provides attributes which are introduced in the MultiSpeak specification's *mspPointObject* abstract object class.

Class Name	***MPointFeature***

Type and Purpose	Abstract class
Inherits From	*MElectricObject*
Geometry	Not applicable
Attributes	**gridLocation** - *Grid Location* - <text, 255> - This is a utility-assigned location identifier that may or may not be tied to any GIS coordinate system. Sometimes referred to by the utility as "map number" or "grid number". **rotation** – *Rotation Angle* - <text, 255> - Rotation angle for the physical feature, expressed in degrees measured clockwise. **facilityID** – *Facility ID* - <text, 255> - Utility string designation for the object.
Subtypes	Not applicable
Relationships	Not applicable
Notes	

2.1.15 *MUGElectricLine*

MUGElectricLine is an abstract class which provides attributes for UGPrimaryLine and UGSecondaryLine classes. There is no equivalent abstract class in the MultiSpeak specification, this class is provided for consistent attributes found in each MultiSpeak underground conductor class.

Class Name	***MUGElectricLine***
Type and Purpose	Abstract class
Inherits From	*MElectricLineFeature*
Geometry	Not applicable
Attributes	**isInConduit** – *Is In Conduit? (Y/N)* - <integer> - Whether this section is installed in conduit (Y/N). Uses the *eDomainBooleanYesNo* domain. **manufacturer** – *Manufacturer* - <text, 255> - Manufacturer of conductor in this section.
Subtypes	Not applicable
Relationships	Not applicable
Notes	

2.2 Conductors, Network Devices and Administration

This section provides information on feature and object classes for conductors, network devices and also administrative classes related to these objects.

2.2.1 Breaker

Breaker is an object class table that describes individual breaker units. It is related to the *overcurrentDeviceBank* feature class which provides the geographic location and participates in the geometric network.

Class Name	Breaker
Type	Object class
Inherits From	*MElectricOvercurrentDevice*
Alias	Breaker
Geometry	Not applicable
Attributes	***All attributes are inherited.***
Subtypes	Not applicable
Relationships	overcurrentDeviceBank
Notes	

2.2.2 Capacitor

Capacitor is an object class table that describes individual capacitor units. It is related to the *CapacitorBank* feature class which provides the geographic location and participates in the geometric network.

Class Name	Capacitor
Type	Object Class
Inherits From	*MDeviceObject*
Alias	Capacitor
Geometry	Not Applicable
Attributes	**phase** – *Phase* - <text, 255> - The phase of the capacitor. Uses the *eDomainSinglePhase* domain. **kvar** – *kVA Rating* - <double> - The kVA rating of the capacitor unit. **capacitorBankID** – *Capacitor Bank ID* - <GUID> - The foreign key to the related Capacitor Bank feature.
Subtypes	Capacitor Unit (subtypeCode=1)
Relationships	CapacitorBank
Notes	

2.2.3 CapacitorBank

CapacitorBank is a junction feature class that describes the capacitor bank.

CapacitorBank is related to the *Capacitor* object class for which individual capacitor units are associated with this bank feature class.

Class Name	CapacitorBank
Type	Simple Junction Feature Class
Inherits From	*MElectricBankFeature*
Alias	Capacitor Bank
Geometry	Point
Attributes	**connectionCd** – *Connection Code* - <text, 255> - The bank connection code. Uses the *eDomainConnectionType* domain. **swType** – *Switch Control Type* - <text, 255> - Type of capacitor switch control used in this section. Uses the *eDomainSwitchType* domain. **swStatus** – *Switch Status* - <text, 255> - Switch status of this capacitor. Uses the *eDomainSwitchStatus* domain. **swOn** - *Switch-on Value* - <double> - Capacitor switches ON at this value. **swOff** - *Switch-off Value* - <double> - Capacitor switches OFF at this value. **bankKvar** – *Bank kVar Value* - <double> - Total bank size, expressed in kVar. **volts** – *Volts* - <double> - L-G volts for wye-connected section; L-L volts for delta-connected section. Zero unless non-standard. Expressed in kV.
Subtypes	Fixed CapacitorBank (subtypeCode=1) Switched CapacitorBank (subtypeCode=2)
Relationships	Capacitor
Notes	

2.2.4 Customer

Customer is an object class table that provides information on individual customer accounts. *Customer* is related to the *ServiceLocation* feature class.

Class Name	Customer
Type	Object Class
Inherits From	*MElectricObject*
Alias	Customer
Geometry	Not Applicable
Attributes	**lastName** – *Last Name* - <text, 255> - Last name of customer residing at the service address. **firstName** – *First Name* - <text, 255> - Last name of customer residing at the service address. **mName** – *Middle Name* - <text, 255> - Middle name of customer residing at the service address. **dBAName** – *DBA Name* - <text, 255> - DBA name for business customers. **billAddr1** – *Billing Address Line 1* - <text, 255> - Billing address line 1. **billAddr2** – *Billing Address Line 2* - <text, 255> - Billing address line 2. **billCity** – *City of Billing Address* - <text, 255> - City of billing address. **billState** – *State of Billing Address* - <text, 255> - State of billing address. **billZip** – *Zip Code of Billing Address* - <text, 255> - The nine-character zip code of the billing address.

Subtypes	Not applicable
Relationships	ServiceLocation
Notes	

2.2.5 Fuse

Fuse is an object class table that describes individual fuse units. It is related to the *overcurrentDeviceBank* feature class which provides the geographic location and participates in the geometric network.

Class Name	Fuse
Type	Object class
Inherits From	*MElectricOvercurrentDevice*
Alias	Fuse
Geometry	Not applicable
Attributes	**linkRtg** - *Fuse Link Rating* - <double> - The current rating of the fuse link.
Subtypes	Not applicable
Relationships	overcurrentDeviceBank
Notes	

2.2.6 Generator

Generator is a junction feature class that describes a generator, and participates in the geometric network.

Class Name	Generator
Type	Simple Junction Feature Class
Inherits From	*MElectricMotorGenerator*
Alias	Generator
Geometry	Point
Attributes	**model** – *Generator Model* - <integer> - The generator model code. Uses the *eDomainGeneratorModel* domain. **kwOut** – *Swing kVAR Power Output* - <double> - For Swing kVAR model, real power output. Expressed in kW. **kvarLead** – *Swing kVAR Max Leading Reactive Output* - <double> - For Swing kVAR model, maximum leading reactive power output. Expressed in kVAR. **kvarLag** – *Swing kVAR Max Lagging Reactive Output* - <double> - For Swing kVAR model, Maximum lagging reactive power output. Expressed in kVAR. **connected** – *Connection Type* - <text, 255> - Type of connection – wye or delta. Uses the *eDomainConnectionTypeWyeDelta* domain.
Subtypes	Not applicable
Relationships	overcurrentDeviceBank
Notes	

2.2.7 Meter

Meter is a concrete object class that describes meters that are located as individuals or multiple meters in a bank. The Meter object class is related to the MeterBase object class, which in turn is related to the ServiceLocation feature class which is the point where the service ends.

Class Name	Meter
Type	Object Class
Inherits From	*MDeviceObject*
Alias	Meter
Geometry	Not Applicable
Attributes	**meterNo** – *Meter Number* - <text, 255> - Meter number (utility-defined identifier usually called a Company Number). **serialNumber** – *Serial Number* - <text, 255> - Meter serial number. This is the original number assigned to the meter by the manufacturer. **meterType** – *Meter Type* - <text, 255> - Meter type/model. **manufacturer** – *Manufacturer* - <text, 255> - Meter manufacturer. **AMRDeviceType** – *AMR Device Type* – <text, 255> - Vendor-specific Device designation. **AMRVendor** – *AMR Vendor* - <text, 255> - Name of AMR vendor. **MeterBaseID** – *Meter Base ID* - <GUID> - The foreign key to the related MeterBase object.
Subtypes	Service Meter (subtypeCode=1) Primary Meter (subtypeCode=2)
Relationships	MeterBase
Notes	

2.2.8 MeterBase

The meter base is that portion of a metering installation that remains installed when a meter is removed or replaced. It may contain a socket and instrument transformers, among other items. The MeterBase object class describes meter base objects associated with a ServiceLocation feature point. A ServiceLocation can have multiple MeterBase features associated with it. The MeterBase has a one-to-one relationship with the Meter object class as well.

Class Name	MeterBase
Type	Object Class
Inherits From	*MDeviceObject*
Alias	Meter Base
Geometry	Not Applicable
Attributes	**meterNo** – *Meter Number* - <text, 255> - Meter number (utility-defined identifier usually called a Company Number). **form** – *Form Designation* - <text, 255> - The form designator for the type of meter that can be inserted into this meter base. **class** – *Meter Class* - <text, 255> - The maximum of the meter load rating in amperes for meters that could be used in this meter base. **baseType** – *Base Type* - <text, 255> - The maximum of the meter load rating in amperes for meters that could be used in this meter base. Uses the *eDomainMeterBaseType* domain.

	serviceLocationID – *Service Location ID* - <GUID> - The foreign key to the related ServiceLocation feature.
Subtypes	Meter Base (subtypeCode=1)
Relationships	MeterBase ServiceLocation
Notes	

2.2.9 Motor

Motor is a junction feature class that describes a generator, and participates in the geometric network.

Class Name	Motor
Type	Simple Junction Feature Class
Inherits From	*MElectricMotorGenerator*
Alias	Motor
Geometry	Point
Attributes	**status** – *Motor Status* - <integer> - Motor status code. Uses the *eDomainMotorStatus* domain. **hp** – *Horsepower* - <double> - The rated horsepower. **pf** – *Rated Power Factor* - <double> - Running power factor at full load. Expressed in per unit. **eff** – *Efficiency at Full Load* - <double> - Efficiency at full load. Expressed in per unit. **lgVolts** – *Line-to-ground Voltage* - <double> - The line-to-ground voltage, expressed in kV. **dropout** – *Dropout* - <double> - Motor drops out if voltage is below this fraction, expressed in per unit. Integer NEMA motor type code. **nemaTyp** – *NEMA Type* - <integer> - The integer NEMA motor type code. Uses the *eDomainNEMATypeCode* domain. **limit** – *Limit* - <double> - Motor will not start if below this fraction, expressed in per unit. **sftStTyp** – *Soft Start Type Code* - <integer> - Integer soft start type code. Uses the *eDomainSoftStartTypeCodes* domain. **sftStr** – *Soft Start Resistance* - <double> - For soft start, starting line resistance (real part of impedance), expressed in Ohms. **sftStTap** – *Soft Start Transformer* - <double> - For autotransformer starting. At start, the motor terminal voltage will be this fraction of the normal across-the-line starting voltage, expressed in “per unit”. **lrPf** – *Locked Rotor Power Factor* - <double> - Locked rotor power factor, expressed in per unit. **lrMult** – *Locked Rotor kVA* - <double> - Locked rotor kVA per horsepower.
Subtypes	Not applicable
Relationships	
Notes	

2.2.10 OHPrimaryLine

OHPrimaryLine is a concrete object class that describes overhead primary conductors.

Class Name	OHPrimaryLine
Type	Complex Edge Feature Class
Inherits From	*MElectricConductor*
Alias	OH Primary Line
Geometry	Polyline
Attributes	***All attributes are inherited.***
Subtypes	Single Phase (subtypeCode=1) Two Phase (subtypeCode=2) Three Phase (subtypeCode=3) Bus Bar (subtypeCode=4)
Relationships	Not applicable
Notes	

2.2.11 OHSecondaryLine

OHSecondaryLine is a concrete object class that describes overhead secondary conductors.

Class Name	OHSecondaryLine
Type	Complex Edge Feature Class
Inherits From	*MElectricConductor*
Alias	OH Secondary Line
Geometry	Polyline
Attributes	**secondaryType** – *Secondary Type* - <text, 255> - Type of secondary line. Uses the *eDomainSecondaryType* domain.
Subtypes	Single Phase (subtypeCode=1) Three Phase (subtypeCode=2) Streetlight Connection (subtypeCode=3)
Relationships	Not applicable
Notes	

2.2.12 OvercurrentDeviceBank

OvercurrentDeviceBank is a junction feature class that describes a bank of overcurrent devices such as fuses, reclosers, sectionalizers, or breakers. *OvercurrentDeviceBank* is related to the *Fuse, Recloser, Sectionalizer* and *Breaker* object classes which represent the individual overcurrent device units associated with this bank feature class.

Class Name	OvercurrentDeviceBank
Type	Simple Junction Feature Class
Inherits From	*MElectricSwitchingBankFeature*
Alias	Overcurrent Device Bank

Geometry	Point
Attributes	***All attributes are inherited.***
Subtypes	Fuse Bank (subtypeCode=1) Recloser Bank (subtypeCode=2) Sectionalizer Bank (subtypeCode=3) Breaker Bank (subtypeCode=4) Combined Overcurrent Bank (subtypeCode=5)
Relationships	Fuse Recloser Sectionalizer Breaker
Notes	

2.2.13 Recloser

Recloser is an object class table that describes individual recloser units. It is related to the *overcurrentDeviceBank* feature class which provides the geographic location and participates in the geometric network.

Class Name	Recloser
Type	Object class
Inherits From	*MElectricOvercurrentDevice*
Alias	Recloser
Geometry	Not applicable
Attributes	***All attributes are inherited.***
Subtypes	Not applicable
Relationships	overcurrentDeviceBank
Notes	

2.2.14 Regulator

Regulator is an object class table that describes individual regulator units. It is related to the *RegulatorBank* feature class which provides the geographic location and participates in the geometric network.

Class Name	Regulator
Type	Object Class
Inherits From	*MDeviceObject*
Alias	Regulator
Geometry	Not Applicable
Attributes	**eaEquipment** – *EA Equipment* - <text, 255> - Description for the type of regulator in this section. **vOut** – *Output Voltage* - <double> - The target output volts. **ldcR** – *Line Drop Compensation Resistance* - <double> - The line drop compensation resistance. **ldcX** – *Line Drop Compensation Reactance* - <double> - The line drop compensation reactance.

	fhHi – *First House High Protection* - <double> - The first house high protection setting, in per unit volts. **lhHi** – *First House Low Protection* - <double> - The first house low protection setting, in per unit volts **phase** – *Phase* - <text, 255> - The phase of the regulator. Uses the *eDomainSinglePhase* domain. **kva** – *kVA Rating* - <double> - The kVA rating of the regulator unit. **regulatorBankID** – *Regulator Bank ID* - <GUID> - The foreign key to the related Regulator Bank feature.
Subtypes	Regulator Unit (subtypeCode=1)
Relationships	RegulatorBank
Notes	

2.2.15 RegulatorBank

RegulatorBank is a junction feature class that describes the regulator bank. *RegulatorBank* is related to the *Regulator* object class for which individual regulator units are associated with this bank feature class.

Class Name	RegulatorBank
Type	Simple Junction Feature Class
Inherits From	*MElectricBankFeature*
Alias	Regulator Bank
Geometry	Point
Attributes	**regType** – *Regulator Type* - <text, 255> - Code for type of regulator in this section. **ctrlPhase** – *Control Phase* - <text, 255> - Code for phase used to control regulator in this section. Uses the *eDomainSinglePhase* domain. **wdgType** – *Winding Type* - <text, 255> - Code for winding type of regulator in this section.
Subtypes	Regulator Bank (subtypeCode=1)
Relationships	Regulator
Notes	

2.2.16 Sectionalizer

Sectionalizer is an object class table that describes individual recloser units. It is related to the *overcurrentDeviceBank* feature class which provides the geographic location and participates in the geometric network.

Class Name	Sectionalizer
Type	Object class
Inherits From	*MElectricOvercurrentDevice*
Alias	
Geometry	Not applicable
Attributes	***All attributes are inherited.***
Subtypes	Not applicable

Relationships	overcurrentDeviceBank
Notes	

2.2.17 ServiceLocation

ServiceLocation is a junction feature class that describes the location where service is provided to a facility or structure. ServiceLocation feature class is related to the MeterBase object class, which in turn is associated with the Meter object class.

Class Name	ServiceLocation
Type	Simple Junction Feature Class
Inherits From	*MElectricPointFeature*
Alias	Service Location
Geometry	Point
Attributes	**custID** – *Customer ID* - <text, 255> - Customer's identification number (Customer ID). **accountNumber** – *Account Number* - <text, 255> - Account number for this service location. **facilityName** – *Facility Name* - <text, 255> - Name of facility for multi-site business customer. **siteID** – *Site ID* - <text, 255> - Identifier for site for a multi-site business customer. **servAddr1** – *Service Address Line 1* - <text, 255> - First address line of service address. **servAddr2** – *Service Address Line 2* - <text, 255> - Second address line of service address. **servCity** – *City* - <text, 255> - City of service address. **servState** – *State* - <text, 255> - State of service address. **servZip** – *Zip Code* - <text, 255> - The nine-character zip code of service address. **servType** – *Service Type* - <text, 255> - Service type (Rate code). **servStatus** – *Service Status* - <text, 255> - Service status code. **outageStatus** – *Outage Status* - <text, 255> - Outage status description. Uses the *eDomainPowerStatus* domain. **specialNeeds** – *Special Needs* - <text, 255> - Life support, or other critical customer special needs code. **customerID** – *Customer ID* - <GUID> - The foreign key to the related Customer record.
Subtypes	Service Point (subtypeCode=1) Primary Meter (subtypeCode=2)
Relationships	Customer MeterBase
Notes	

2.2.18 Streetlight

Streetlight is a junction feature class that describes a streetlight, and participates in the geometric network.

Class Name	Streetlight
Type	Simple Junction Feature Class
Inherits From	*MPointFeature*
Alias	Streetlight
Geometry	Point
Attributes	**lightType** – *Light Type* - <text, 255> - Type of fixture (e.g., high pressure sodium, mercury, etc.) **wattage** – *Light Wattage* - <text, 255> - The lamp wattage. **brightness** – *Brightness* - <double> - Lamp brightness, expressed in lumens. **transformerBankID** – *Transformer Bank ID* - <GUID> - The foreign key to the related Transformer Bank feature.
Subtypes	Not applicable
Relationships	TransformerBank
Notes	

2.2.19 Substation

Substation is a point feature class that describes a substation location.

Class Name	Substation
Type	Simple Junction Feature Class
Inherits From	*MElectricPointFeature*
Alias	Substation
Geometry	Not applicable
Attributes	**zMin** – *Minimum Source Impedance* - <text, 255> - Name of minimum source impedance. **zMax** – *Maximum Source Impedance* - <text, 255> - Name of maximum source impedance. **units** – *Impedance Units* - <text, 255> - Unit used to express impedances. Uses the *eDomainImpedanceUnits* domain. **busVolts** – *Bus Voltage Multiplier* - <double> - The bus voltage multiplier, expressed in per unit. **ohGndZ** – *Overhead Ground Impedance* - <double> - Ground impedance for a fault on an overhead section, expressed in ohms. **ugGndZ** – *Underground Ground Impedance* - <double> - Ground impedance for a fault on an underground section, expressed in ohms. **nomVolts** – *Nominal Voltage* - <double> - Nominal voltage for section. **ldAolloc** – *Load Connection Control Indicator* - <integer> - Whether the substation a load connection control point. Uses the *eDomainBooleanYesNo* domain. **ldCon** – *Load Connection* - <text, 255> - Load connection type. Uses the *eDomainConnectionTypeWyeDelta* domain. **isRegulated** – *Regulated Indicator* - <integer> - Whether the substation source is regulated. Uses the *eDomainBooleanYesNo* domain. **name** – *Substation Name* - <text, 255> - The name of the substation.
Subtypes	Not applicable
Relationships	
Notes	

2.2.20 Switch

Switch is an object class table that describes individual capacitor units. It is related to the *SwitchDeviceBank* feature class which provides the geographic location and participates in the geometric network.

Class Name	Switch
Type	Object class
Inherits From	*MElectricSwitchingDevice*
Alias	Switch
Geometry	Not applicable
Attributes	**switchDeviceBankID** – *Switch Device Bank ID* - <GUID> - The foreign key to the related SwitchDevice Bank feature.
Subtypes	Switch Unit (subtypeCode=1)
Relationships	SwitchDeviceBank
Notes	

2.2.21 SwitchDeviceBank

SwitchDeviceBank is a junction feature class that describes a bank of switches. *SwitchDeviceBank* is related to the *Switch* object class for which individual switch units are associated with this bank feature class.

Class Name	SwitchDeviceBank
Type	Simple Junction Feature Class
Inherits From	*MElectricSwitchingBankFeature*
Alias	Switch Device Bank
Geometry	Point
Attributes	***All attributes are inherited.***
Subtypes	Overhead SwitchDeviceBank (subtypeCode=1) Underground SwitchDeviceBank (subtypeCode=2)
Relationships	Switch
Notes	Note: As SwitchDeviceBank is also a protective device bank, it could potentially be associated with protective devices beyond switches such as: fuse, recloser, sectionalizer and breaker. Although the relationships are not included in the template, this is possible implementation.

2.2.22 Transformer

Transformer is an object class table that describes individual transformer units. It is related to the *TransformerBank* feature class which provides the geographic location and participates in the geometric network.

Class Name	Transformer
Type	Object Class
Inherits From	*MDeviceObject*

Alias	Transformer
Geometry	Not Applicable
Attributes	**manufacturer** – *Manufacturer* - <text, 255> - Transformer manufacturer. **serialNumber** – *Serial Number* **–** <text, 255> - Transformer serial number. **kva** – *kVA Rating* – <double> - Transformer size (in kVA). **xfmrType** – *Transformer Type* – <text, 255> - Transformer type code. **status** – *Status Code* – <text, 255> - Transformer status code. **phases** - *Phases* - <integer> - The number of transformer phases. **phase** – *Phase* - <text, 255> - The phase of the transformer. Uses the *eDomainSinglePhase* domain. **impedance** – *Impedance* - <double> - Transformer impedance. **nLLoss** - *No Load Loss* - <double> - No load loss value. **fLLoss** - *Full Load Loss* - <double> - Full load loss value. **price** – *Price* - <double> - Purchase price. **priVoltsLo** - *Primary Voltage-Low* - <double> - Transformer primary voltage (lower if dual rated). **priVoltsHi** - *Primary Voltage-High* - <double> - Transformer primary voltage (higher if dual rated). **secVoltsLo** - *Secondary Voltage-Low* - <double> - Transformer secondary voltage (lower if dual rated). **secVoltsHi** - *Secondary Voltage-High* - <double> - Transformer secondary voltage (higher if dual rated). **mfgDate** – *Date Manufactured* - <date> - Transformer manufacture date. **pcb –** PCB Code - <text, 255> - Transformer PCB code. **eaEquipment** – *EA Equipment*- <text, 255> - Description for the type of regulator in this section. **transformerBankID** – *Transformer Bank ID* - <GUID> - The foreign key to the related Transformer Bank feature.
Subtypes	Transformer Unit (subtypeCode=1)
Relationships	TransformerBank
Notes	

2.2.23 TransformerBank

TransformerBank is a junction feature class that describes the transformer bank. *TransformerBank* is related to the *Transformer* object class for which individual transformer units are associated with this bank feature class.

Class Name	TransformerBank
Type	Simple Junction Feature Class
Inherits From	*MElectricBankFeature*
Alias	Transformer Bank
Geometry	Point
Attributes	**wdgCode** – *Winding Code* - <text, 255> - Transformer winding connection code. Uses the *eDomainTransformerWindingCode* domain. **transDescr** – *Transformer Description* - <text, 255> - Description of transformer. **mounting** – *Mounting* - <text, 255> - Type of mounting for this bank. Uses the *eDomainDeviceMounting* domain. **vInput** – *Input Voltage* - <double> - Transformer input voltages values are in kV. Input voltage values are stored L-G (Wye connected) unless the winding code is D-Y or D-D and then values are stored L-L (Delta

	connected). **vOut** – *Output Voltage* - <double> - Transformer output voltage in kV. Output voltage is L-G unless winding code is Grounded Y-D, Ungrounded Y-D, Open Y-D or D-D. **sourceSideConfig**– *Source Side Configuration* - <text, 255> - Type of source side configuration. Uses the *eDomainSourceSideConfig* domain. **tertVolts** - *Tertiary Voltage* - <double> - Tertiary voltage in kV. **vOutNom** - *Nominal Output Voltage* - <double> - Nominal output voltage. **vOutNomTertiary** - *Nominal Output Voltage - Tertiary*. - <double> - Nominal output voltage on the tertiary. **secProt** - *Secondary Protection* - <text, 255> - Description of secondary protection. **secVolt** - *Secondary Voltage* - <double> - Secondary voltage. **ratedVolt** - *Rated Voltage* - <double> - Rated voltage. **priProt** - *Primary Protection* - <text, 255> - Description of primary protection.
Subtypes	Overhead TransformerBank (subtypeCode=1) Underground TransformerBank (subtypeCode=2) Step TransformerBank (subtypeCode=3)
Relationships	Transformer
Notes	

2.2.24 UGPrimaryLine

UGPrimaryLine is a concrete object class that describes underground primary conductors.

Class Name	UGPrimaryLine
Type	Complex Edge Feature Class
Inherits From	*MUGElectricLine*
Alias	UG Primary Line
Geometry	Polyline
Attributes	***All attributes are inherited.***
Subtypes	Single Phase (subtypeCode=1) Two Phase (subtypeCode=2) Three Phase (subtypeCode=3) Bus Bar (subtypeCode=4)
Relationships	Not applicable
Notes	

2.2.25 UGSecondaryLine

UGSecondaryLine is a concrete object class that describes underground secondary conductors.

Class Name	UGSecondaryLine
Type	Complex Edge Feature Class
Inherits From	*MUGElectricLine*

Alias	UG Secondary Line
Geometry	Polyline
Attributes	**uGSecType** – *Secondary Type* - <text, 255> - Type of secondary. Uses the *eDomainSecondaryType* domain.
Subtypes	Single Phase (subtypeCode=1) Three Phase (subtypeCode=2) Streetlight Connection (subtypeCode=3)
Relationships	Not applicable
Notes	

2.3 Surface and Overhead Structures

The Overhead Structures section contains non-network features which support conductors and devices.

2.3.1 Assembly

Assembly is an object class table that describes individual pole-top assemblies on a *Pole* feature.

Class Name	Assembly
Type	Object Class
Inherits From	*MElectricObject*
Alias	Assembly
Geometry	Not Applicable
Attributes	**unitCode** – *Assembly Unit Code* - <text, 255> - The assembly unit code. **unitActn** – *Assembly Action* - <text, 255> - The action to be taken with the assembly. Uses the *eDomainAssemblyAction* domain. **spanTyp** – *Span Type* - <text, 255> - The span type on the assembly. Uses the *eDomainAssemblySpanType* domain. **unitLength** – *Unit Length* - <double> - The assembly unit length. poleVar **poleVar** – *Pole Variant* - Ref Code 1 – <text, 255> - Reference code 1 – pole variant. Used in some systems. **wireVar** – *Wire Variant* - Ref Code 2 – <text, 255> - Reference code w – wire variant. Used in some systems. **neutVar** – *Neutral Variant* - Ref Code 3 – <text, 255> - Reference code 1 – pole variant. Used in some systems. **miscVar** – *Misc Variant* - Ref Code 4 - <text, 255> - Reference code 4 – miscellaneous variant. Used in some systems. **quantity** – *Quantity* - <double> - The quantity of unit required. **poleHeight** – *Pole Height* - <integer> - The pole height. **poleClass** – *Pole Class* - <text, 255> - The pole class. Uses the *eDomainPoleClass* domain. **poleType** – *Pole Type* - <text, 255> - The pole type. Uses the *eDomainPoleType* domain. **poleID** – *Pole ID* - <GUID> - The foreign key to the related Pole feature.
Subtypes	Not applicable

Relationships	Pole
Notes	

2.3.2 Guy

Guy is a point feature class that describes a guy attachment for a pole.

Class Name	Guy
Type	Point feature class
Inherits From	*MPointFeature*
Alias	Guy
Geometry	Not applicable
Attributes	**status** – *Status* - <text, 255> - The status of the guy assembly. Uses the *eDomainGuyStatus* domain. **guyType** – *Guy Type* - <text, 255> - The type of guy assembly. Uses the *eDomainGuyType* domain. **guyAgainst** – *Guy Set Against* - <text, 255> - The configuration the guy assembly is set against. Uses the *eDomainGuyAgainst* domain. **attachmentHeight** – *Attachment Height* - <double> - The height of the guy assembly.
Subtypes	Not applicable
Relationships	
Notes	

2.3.3 JointUse

JointUse is an object class table that describes individual joint use attachments on a *Pole*. *JointUse* is related to the *Pole* feature class which provides the pole location.

Class Name	JointUse
Type	Object Class
Inherits From	*MElectricObject*
Alias	Joint Use
Geometry	Not Applicable
Attributes	**companyID** – *Company ID* - <text, 255> - The company ID that owns the attachment. **jUseType** – *Joint Use Type* - <text, 255> - The joint use type. **jUseHeight** – *Joint Use Height* - <text, 255> - The joint use type. **isGuyed** – *Is Guyed? (Y/N)* - <integer> - Whether the attachment is guyed. Uses the *eDomainBooleanYesNo* domain. **poleID** – *Pole ID* - <GUID> - The foreign key to the related Pole feature.
Subtypes	Not applicable
Relationships	Pole
Notes	

2.3.4 Pole

Pole is a point feature class that describes a pole support structure. It supports electrical network features, and it does not participate in the electrical geometric network.

Class Name	Pole
Type	Point feature class
Inherits From	*MPointFeature*
Alias	Pole
Geometry	Point
Attributes	**poleHeight** – *Pole Height* - <integer> - The pole height. **poleClass** – *Pole Class* - <text, 255> - The pole class. Uses the *eDomainPoleClass* domain. **poleType** – *Pole Type* - <text, 255> - The pole type. Uses the *eDomainPoleType* domain. **poleNumber** – *Pole Number* - <text, 255> - The pole number. **manufacturer** – *Manufacturer* - <text, 255> - The pole manufacturer. **mfgDate** – *Date Manufactured* - <date> - The pole manufacture date. **owner** – *Owner* - <text, 255> - The pole manufacture date. **treatment** – *Treatment Type* - <text, 255> - The pole treatment type. Uses the *eDomainPoleTreatmentType* domain. **inspDate** – *Inspection Date* - <date> - The date of last inspection. **poleUse** – *Pole Use* - <text, 255> - The pole use type. Uses the *eDomainPoleUseType* domain.
Subtypes	Not applicable
Relationships	Assembly JointUse
Notes	

2.3.5 PrimaryCabinet

PrimaryCabinet is a point feature class that describes a primary voltage pad-mounted or vault mounted enclosure.

Class Name	PrimaryCabinet
Type	Point feature class
Inherits From	*MPointFeature*
Alias	Primary Cabinet
Geometry	Not applicable
Attributes	**structType** – *Structure Type* - <text, 255> - The description for the type of primaryCabinet.
Subtypes	Switch Cabinet (subtypeCode=1) Junction Box (subtypeCode=2) Pedestal (subtypeCode=3) Manhole (subtypeCode=4) Handhole (subtypeCode=5) Vault (subtypeCode=6)
Relationships	Not applicable
Notes	

2.3.6 Riser

Riser is a point feature class that describes a riser attached to a pole.

Class Name	Riser
Type	Point feature class
Inherits From	*MPointFeature*
Alias	Riser
Geometry	Not applicable
Attributes	**constType** – *Construction Type* - <text, 255> - The riser construction type. **riserHeight** – *Riser Height* - <integer> - Mounting height of riser on the pole. **material** – *Material Type* - <text, 255> - The description of the type of material used for the riser.
Subtypes	Not applicable
Relationships	
Notes	

2.3.7 SecondaryJunctionBox

SecondaryJunctionBox is a point feature class that describes a secondary junction box.

Class Name	SecondaryJunctionBox
Type	Point feature class
Inherits From	*MPointFeature*
Alias	Secondary Junction Box
Geometry	Not applicable
Attributes	**junctionBoxType** – Junction Box Type - <text, 255> - The description for the type of junction box.
Subtypes	Secondary Junction Box (subtypeCode=1)
Relationships	Not applicable
Notes	

2.3.8 SpanGuy

SpanGuy is a simple line feature class that describes a span guy line attachment for a pole.

Class Name	SpanGuy
Type	Line feature class
Inherits From	*MLineFeature*
Alias	Span Guy
Geometry	Not applicable
Attributes	**attachmentHeight** – Attachment Height - <double> - The height of the guy assembly.
Subtypes	Not applicable
Relationships	
Notes	

3.0 Domains and Relationships

3.1 Domains

This section is comprised of the data model domains. Domains offer a way to define a range of values that can be used for multiple attribute fields. Using domains helps ensure data integrity by limiting the choice of values for a particular field. Validation for coded value domains is accomplished by restricting the user to choose field values from a dropdown list. Range domains are automatically validated during editing.

3.1.1 eDomainAssemblyAction

Provides options for an action to be performed on the pole assembly.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Create	C
Replace	R
Existing	E

3.1.2 eDomainAssemblySpanType

Provides options for the type of secondary line.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Primary	primary
Secondary	secondary
Service	service
Underbuild	underbuild
None	none

3.1.3 eDomainBooleanYesNo

Provides options for “Yes” and “No” values for an integer field.

Domain Type	Coded Values
Field Type	Integer
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Yes	1
No	0

3.1.4 eDomainConnectionType

Provides options for the device’s connection type.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
ShuntWye	ShuntWye
ShuntDelta	ShuntDelta
ShuntSameAsParent	ShuntSameAsParent
Series	Series

3.1.5 eDomainConnectionTypeWyeDelta

Provides options for the connection type.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Wye	W
Delta	D

3.1.6 eDomainDeviceMounting

Provides options for the device mounting.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Bus-mounted	Bus-mounted
Pole-mounted	Pole-mounted
Pad-mounted	Pad-mounted
Vault-mounted	Vault-mounted
Other	Other

3.1.7 eDomainGeneratorModel

Provides options for the generator model.

Domain Type	Coded Values
Field Type	Integer
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Negative load	0
Swing generator (unlimited)	1
Swing kVA	2
Swing kVAR	3

3.1.8 eDomainGuyAgainst

Provides options for how the guy assembly is set against.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**

Dead-end	Dead-end
Angle	Angle
Storm-right	Storm-right
Storm-left	Storm-left
Unknown	Unknown
Other	Other

3.1.9 eDomainGuyStatus

Provides options for the guy status.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
New	N
Existing	E
Retire	R

3.1.10 eDomainGuyType

Provides options for the guy type.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Down	Down
Double down	Double down
Other	Other
Unknown	Unknown

3.1.11 eDomainImpedanceUnits

Provides options for the impedance units.

Domain Type	Coded Values
Field Type	Text

Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Ohms	Ohms
Percent	Percent
Per Unit	Per Unit
Total	Total

3.1.12 eDomainMeterBaseType

Provides options for the type of meter base.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Base Type 1	Base Type 1
S-Base	S-base
A-Base	A-Base
B-Base	B-Base
K-Base	K-Base
P-Base	P-Base
IEC Bottom Connect	IECBottomConnect
Switchboard	switchboard
Rack-mount	rackmount

3.1.13 eDomainMotorStatus

Provides options for the motor status type.

Domain Type	Coded Values
Field Type	Integer
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Disconnected	0
Off	1

Locked rotor	2
Soft start	3
Running	4

3.1.14 eDomainMSPObjectAction

Provides options for the type of action for MultiSpeak messaging.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
New	New
Change	Change
Delete	Delete
Replace	Replace
Link	Link
Unlink	Unlink

3.1.15 eDomainNEMATypeCode

Provides options for the NEMA type code.

Domain Type	Coded Values
Field Type	Integer
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
None	0
A	1
B	2
C	3
D	4
E	5
F	6
G	7
H	8
J	9
K	10

L	11
M	12
N	13
P	14
R	15
S	16
T	17
U	18

3.1.16 eDomainPhase

Provides options for the phase.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
A	A
B	B
C	C
AB	AB
AC	AC
BC	BC
ABC	ABC
Unknown	Unknown

3.1.17 eDomainPoleClass

Provides options for the pole class.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Class 1	Class 1
Class 2	Class 2
Class 3	Class 3

Class 4	Class 4
Class 5	Class 5
Class 6	Class 6
Class 7	Class 7
Class 8	Class 8
Class 9	Class 9
Class 10	Class 10
H1	H1
H2	H2
H3	H3
H4	H4
H5	H5
H6	H6
H7	H7
Other	Other
Unknown	Unknown

3.1.18 eDomainPoleTreatmentType

Provides options for the pole treatment type.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Butt	Butt
Natural	Natural
Penta	Penta
Creosote	Creosote
CCA	CCA
Chemonite	Chemonite
Napthena	Napthena
Cellon	Cellon
Other	Other
Unknown	Unknown

3.1.19 eDomainPoleType

Provides options for the pole type.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Aluminum	Aluminum
Concrete	Concrete
Fiberglass	Fiberglass
Galvanized	Galvanized
Steel	Steel
Wood	Wood
Wood - Treated	WoodTreated
Wood - Untreated	WoodUntreated
Other	Other
Unknown	Unknown

3.1.20 eDomainPoleUseType

Provides options for the pole use type.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Distribution	Distribution
Transmission	Transmission
Service	Service
Other	Other
Unknown	Unknown

3.1.21 eDomainPosition

Provides options for the device position.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate

Domain Members:	
Name	**Value**
Normally Open	NO
Normally Closed	NC
Unknown	Unknown

3.1.22 eDomainPowerStatus

Provides options for the power status.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Power On	PowerOn
Power Off	PowerOff
Status Unknown	StatusUnknown

3.1.23 eDomainSecondaryType

Provides options for the type of secondary line.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Underbuild	Underbuild
Secondary	Secondary
Service	Service
Other	Other
Unknown	Unknown

3.1.24 eDomainSinglePhase

Provides options for the phase for an individual device.

Domain Type	Coded Values
Field Type	Text

Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
A	A
B	B
C	C
Unknown	Unknown

3.1.25 eDomainSoftStartTypeCodes

Provides options for the soft start type code.

Domain Type	Coded Values
Field Type	Integer
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Starting across the line	0
Impedance soft start	1
Auto transformer start	2
Capacitor assisted start	3
Partial winding start	4
Delta-Wye soft start	5

3.1.26 eDomainSourceSideConfig

Provides options for the device's source side configuration.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
AB	AB
AC	AC
BC	BC

3.1.27 eDomainSwitchStatus

Provides options for the switch status.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Disconnected	Disconnected
On	On
Off	Off

3.1.28 eDomainSwitchType

Provides options for the switch type.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Manual	Manual
Voltage	Voltage
Current	Current
ReactiveAmps	ReactiveAmps
Time	Time
Temperature	Temperature

3.1.29 eDomainTransformerWindingCode

Provides options for the winding code of the transformer.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Y-Y	Y-Y
D-Y	D-Y
Grounded Y-D	Grounded Y-D

Ungrounded Y-D	Ungrounded Y-D
Open Y-D	Open Y-D
D-D	D-D
Y-Y Grounded Imp	Y-Y Grounded Imp
Y-Y 3ph core type	Y-Y 3ph core type
D-D one	D-D one
D-D open	D-D open
Y-Y-D Ground	Y-Y-D Ground
Y-D one	Y-D one
D-Y open	D-Y open

3.1.30 eDomainYesNo

Provides options for “Yes” and “No” values for a text field.

Domain Type	Coded Values
Field Type	Text
Merge Policy	Default Value
Split Policy	Duplicate
Domain Members:	
Name	**Value**
Yes	Yes
No	No

3.2 Relationships

This section is comprised of the relationship classes within the data model. Relationship classes provide an association or link between two objects in a geodatabase.

3.2.1 CapacitorBankHasCapacitor

Composite	Yes	
Cardinality	One To Many	
Notification	Forward	
Attributed	No	
	Origin	**Destination**
Object Class	CapacitorBank	Capacitor
Key	objectGUID (Primary Key)	capacitorBankID (Foreign Key)
Labels	Capacitor Bank	Capacitor

3.2.2 CustomerHasServiceLocation

Composite	No	
Cardinality	One To Many	
Notification	None	
Attributed	No	
	Origin	**Destination**
Object Class	Customer	ServiceLocation
Key	objectGUID (Primary Key)	customerID (Foreign Key)
Labels	Customer	ServiceLocation

3.2.3 MeterBaseHasMeter

Composite	Yes	
Cardinality	One To One	
Notification	Forward	
Attributed	No	
	Origin	**Destination**
Object Class	MeterBase	Meter
Key	objectGUID (Primary Key)	meterBaseID (Foreign Key)
Labels	Meter Base	Meter

3.2.4 OvercurrentDeviceBankHasBreaker

Composite	Yes	
Cardinality	One To Many	
Notification	Forward	
Attributed	No	
	Origin	**Destination**
Object Class	OvercurrentDeviceBank	Breaker
Key	objectGUID (Primary Key)	overcurrentDeviceBankID (Foreign Key)
Labels	Overcurrent Device Bank	Breaker

3.2.5 OvercurrentDeviceBankHasFuse

Composite	Yes	
Cardinality	One To Many	
Notification	Forward	
Attributed	No	
	Origin	**Destination**
Object Class	OvercurrentDeviceBank	Fuse
Key	objectGUID (Primary Key)	overcurrentDeviceBankID (Foreign Key)
Labels	Overcurrent Device Bank	Fuse

3.2.6 OvercurrentDeviceBankHasRecloser

Composite	Yes	
Cardinality	One To Many	
Notification	Forward	
Attributed	No	
	Origin	**Destination**
Object Class	OvercurrentDeviceBank	Recloser
Key	objectGUID (Primary Key)	overcurrentDeviceBankID (Foreign Key)
Labels	Overcurrent Device Bank	Recloser

3.2.7 OvercurrentDeviceBankHasSectionalizer

Composite	Yes
Cardinality	One To Many

Notification	Forward	
Attributed	No	
	Origin	**Destination**
Object Class	OvercurrentDeviceBank	Sectionalizer
Key	objectGUID (Primary Key)	overcurrentDeviceBankID (Foreign Key)
Labels	Overcurrent Device Bank	Sectionalizer

3.2.8 PoleHasAssembly

Composite	Yes	
Cardinality	One To Many	
Notification	Forward	
Attributed	No	
	Origin	**Destination**
Object Class	Pole	Assembly
Key	objectGUID (Primary Key)	poleID (Foreign Key)
Labels	Pole	Assembly

3.2.9 PoleHasJointUse

Composite	Yes	
Cardinality	One To Many	
Notification	Forward	
Attributed	No	
	Origin	**Destination**
Object Class	Pole	JointUse
Key	objectGUID (Primary Key)	poleID (Foreign Key)
Labels	Pole	Joint Use

3.2.10 RegulatorBankHasRegulator

Composite	Yes	
Cardinality	One To Many	
Notification	Forward	
Attributed	No	
	Origin	**Destination**
Object Class	RegulatorBank	Regulator
Key	objectGUID (Primary Key)	regulatorBankID (Foreign Key)

Labels	Regulator Bank	Regulator

3.2.11 ServiceLocationHasMeterBase

Composite	Yes	
Cardinality	One To Many	
Notification	Forward	
Attributed	No	
	Origin	**Destination**
Object Class	ServiceLocation	MeterBase
Key	objectGUID (Primary Key)	serviceLocationID (Foreign Key)
Labels	Service Location	Meter Base

3.2.12 SwitchDeviceBankHasSwitch

Composite	Yes	
Cardinality	One To Many	
Notification	Forward	
Attributed	No	
	Origin	**Destination**
Object Class	SwitchDeviceBank	Switch
Key	objectGUID (Primary Key)	switchDeviceBankID (Foreign Key)
Labels	Switch Device Bank	Switch

3.2.13 TransformerBankHasTransformer

Composite	Yes	
Cardinality	One To Many	
Notification	Forward	
Attributed	No	
	Origin	**Destination**
Object Class	TransformerBank	Transformer
Key	objectGUID (Primary Key)	transformerBankID (Foreign Key)
Labels	Transformer Bank	Transformer

4.0 Review and Testing Notes

The following are additional notes provided to assist users in their initial review and testing of the model template.

- As found on the model template's web page, there are a number of tips and tricks documents and important reference links available at the following sites, login may be required using your ESRI global account login. It is recommended that new users review these documents when getting started with the model template.
    - Building Geodatabases with CASE Tools http://downloads2.esri.com/resources/datamodels/Introduction_to_CASE_Tools.pdf
    - Additional Data Model Tips and Tricks: http://resources.arcgis.com/content/data-models-tips-and-tricks

- The geometric network which has been built within the template contains the following feature classes below. The geometric network is core geodatabase functionality which is particularly useful for electric utilities. It should be noted that the participating classes in the network were based upon information in the MultiSpeak 3.0 specification for electrical network classes, although not clearly defined explicitly as geometric network participants with established network roles. Users of this model template may determine that their implementation may use just some or all of the template's settings.

  Feature classes participating in the template's geometric network, "ElectricNetwork", with their network role:

Feature Class	Network Role
CapacitorBank	Simple Junction
ElectricNetwork_Junctions	Simple Junction
Generator	Simple Junction
Motor	Simple Junction
OHPrimaryLine	Complex Edge
OHSecondaryLine	Complex Edge
OvercurrentDeviceBank	Simple Junction
RegulatorBank	Simple Junction
ServiceLocation	Simple Junction
Streetlight	Simple Junction
Substation	Simple Junction
SwitchDeviceBank	Simple Junction
TransformerBank	Simple Junction
UGPrimaryLine	Complex Edge
UGSecondaryLine	Complex Edge

A key consideration that users will need to address in establishing flow within their geometric network is the establishing of a source feature. One option is the OvercurrentDeviceBank feature class with "Breaker Bank" subtype, which may be considered as a potential choice for this purpose, and users can set the "AncillaryRole" attribute to the "source" value to establish flow.

More information can be found on geometric networks in the ArcGIS Desktop web-based Help, including tracing analysis and other functions. Here are a few links to get started:

- What is a Geometric Network?
  http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?topicname=what_is_a_geometric_network?

- About creating Geometric Networks
  http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=About_creating_geometric_networks

- Editing Geometric Network Features
  http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=Editing_geometric_network_features

Appendix A – ArcGIS Data Types

The following is a listing of standard, non-geometry data types supported by ArcGIS, as referenced in the ArcGIS help documentation. Note: When you create a table or add a field to a table in the geodatabase, fields are created as a specific data type. Data types are classifications that identify possible values for and operations that can be done on the data, as well as the way the data in that field is stored in the database.

When you import data of one type into a field of another data type, you will need to understand what the equivalent data types are between ArcSDE and your database management system (DBMS) because it can impact data content. Also, when creating new datasets in ArcGIS, it is helpful to know the equivalent data types between ArcGIS and your DBMS. For example, if you add a floating point (float) column to an existing feature class, that ArcGIS data type equates to a numeric data type column in a SQL Server database.

More information can be found in the web-based help here:

- Geodatabase field data types: http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?id=2442&pid=2438&topicname=Geodatabase_field_data_types

- Data types in the DBMS: http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=Data_types_in_the_DBMS

Name	Specific range, length, or format	Size (Bytes)	Applications
Short integer	-32,768 to 32,767	2	numeric values without fractional values within specific range; coded values
Long integer	-2,147,483,648 to 2,147,483,647	4	numeric values without fractional values within specific range
Single precision floating point number (Float)	approximately -3.4E38 to 1.2E38	4	numeric values with fractional values within specific range
Double precision floating point number (Double)	approximately -2.2E308 to 1.8E308	8	numeric values with fractional values within specific range
Text	up to 64,000 characters	varies	names or other textual qualities
Date	mm/dd/yyyy	8	date and/or time

	hh:mm:ss A/PM		
BLOB	varies	varies	images or other multimedia
GUID	36 characters enclosed in curly brackets	16 or 38	applications requiring global identifiers