# Runestone Electric Association's (REA) AMI Implementation Using Collector for ArcGIS

Scott Krueger, Billing/IT Supervisor
For Runestone Electric Association

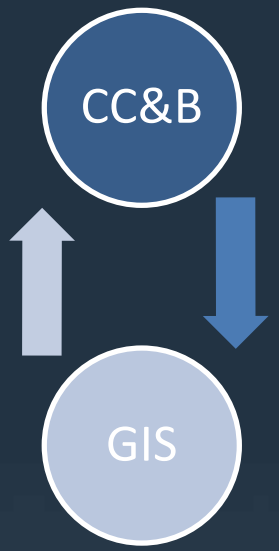
Hillary Bjorstrom, GIS Analyst
For STAR Energy Services LLC

Esri Certified Desktop Associate
ArcGIS Desktop Associate

MID - WEST
EUUG
ESRI Utility Users Group

# REA's AMI Implementation Needs

- REA started an AMI deployment in August 2015
  - 18,000 meters to be changed over 3-year period
  - Aclara TWACS® meters (Two-way Automatic Communication System)
- Integrated with Customer Care & Billing (CC&B) and GIS software
  - REA is a NISC Member and STAR Owner
- Migrate multipliers from collected meters to CC&B
- Something easy for Meter Techs and Office Staff to use




MID - WEST EUUG ESRI Utility Users Group

# REA's AMI Implementation Needs

- Cost-effective solution for meter exchanges
  - Currently not using NISC iVUE® AppSuite
  - Use Internal Staff/STAR for meter exchanges
  - Utilize current software
- Can be used on/off line
- Shows progress to Meter Tech and Office Staff on change outs
- Data current to reflect account changes
  - New accounts or critical customer changes

# AMI Implementation Overview

- Run a nightly scripts to manage data
- Meter Techs change meters
  - Take pictures of old/new meters
  - Document reading and comments
  - GPS location of meter(s)
- Collected data retrieved by syncing feature services in Collector app
- Exports created for batch meter exchange into CC&B
- Batch Meter Exchange imported and processed
  - MultiSpeak Interface to TWACS to search in new meters
  - Exception list created
- Meter exchange and usage reports reviewed for issues

# Using ArcGIS for Server

- ArcGIS for Server to host the data
  - Needed data to be easily updated at night
- Created different versions for office and field edits
- Feature services were set to create a version by user
  - We went with user instead of downloaded map since it is an ongoing project and version are easier to maintain
- Must reconcile and post versions nightly to ensure data changes are reflected from field and office edits and in Collector the next day
  - To do this we use python scripts on a nightly process

# Using ArcGIS Online and Collector App

- ArcGIS Online to create and host web maps and applications
  - We have 3 web maps
    - Office Viewer – No editing
    - Field Live Editing – Live editing
    - Field Editing – Syncing
  - Some hurdles we found with using Collector for ArcGIS:
    - Collector app does not support labeling
    - Limited information on Esri Resources discussing retrieving lost data on iPad

# Using Feature Services

- Creating a feature service not as easy as it seems
- Some issues we ran into that prevented services from being published or syncing from Collector:
  - Data is registered with database
  - Symbology/Labeling issues in .mxd
  - Joins, definition queries, duplicated data sources
  - Security on services in ArcGIS Manager are double locked
  - Security on Syncing utility service is locked
  - Security on arcgisserver folder needs to have ArcGIS user added with read/write rights

***Map and feature services have different requirements***

# Data Used for Meter Exchange Feature Service

- Data exported from REA's NISC CC&B (Must be updated nightly)
  - Active meter (table)
  - Customer (table)
- SQL database table views to create unique ID for joining
  - Note: Compressing the database ensures views function properly
- Data exported from GIS:
  - Location (point feature)
  - Background data (network and landbase)

# Data Used for Meter Exchange Feature Service

- Meter exchange feature - Meter data was spatially located using location point and additional collection fields added
  - Enabled to store attachments (pictures)
- Meter exchange feature must have a Unique ID for joining for data updates from CC&B
  - Used Location ID and Meter Position Number
    - Ex: 020304_1
      (020304=location id + 1=primary meter position)
- Only meters from substations with the Aclara technology were added to the meter exchange feature
  - Reduced unnecessary data updates as other meters were still being exchanged daily
- NISC Schema table created for importing collected features into NISC Interface

# Data Updating Process

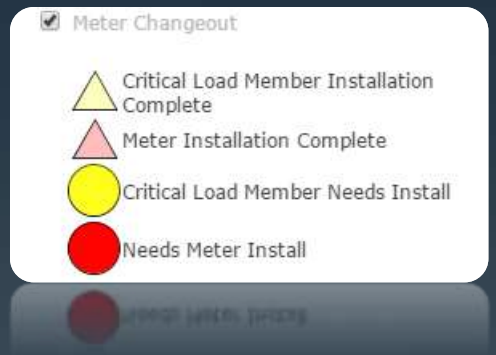
- REA exports data from CC&B nightly
- CC&B data is imported into a temporary database given unique id
  - Used to update meter exchange feature using joins and field calculations
- The meter exchange data is updated nightly using python scripts created from Esri Model Builder exports
- There are 18 models created to run the process at night
  - These models are converted to python and ran in a batch file

# Data Updating Process

- Model Examples: Updating XY, Compresses, Syncs, Reconciles, Date Conversion, Email Notification, Data Loading and Appending, Field Calculations, and Data Exports Including Attachments

# Data Updated in the Process

- Meter exchange feature data updated:
  - Account number
  - Update XY fields with converted coordinates
  - Convert Esri reading date/time to required NISC *yyyymmdd* format
  - Critical customer status
  - Meter number, form, multiplier (using Register Ratio)
  - Install status (for service order installs)
  - Electric background (replica sync one-way)
- Database is reconciled and posted on the SQL Server
- Collected features are imported to a NISC Schema and emails are sent to notify us of meters to import into the CC&B
- Collector app is opened the next day for collection



# Additional Hurdles

- Determining the python running order and workflow
- Schema locking
  - Used scripts to overwrite data
  - Used scripts to notify by email when a script fails
- Needed python scripts that would update data and sync and reconcile versions at night on the server
- Needed to add/remove meters from the meter exchange feature that were taken out or installed after the initial data was created

```
# Process: Table to Table (2)
    if arcpy.Exists("C:\\Mapping\\REA\\METERPROJECT\
        arcpy.Delete_management("C:\\Mapping\\REA\\M
    arcpy.TableToTable_conversion(xls, Temp_mdb, "ru
    print "New runestone meters table created"
```

# Where we are now

- Nightly processes are created and running
- Collector App is deployed and being used
- Web application is being used by office staff
  - Viewing pictures to confirm readings
- Email is sent nightly showing how many meters were submitted
- Continue to test and watch scripting process

# QUESTIONS?
# THANK YOU FOR YOUR TIME!