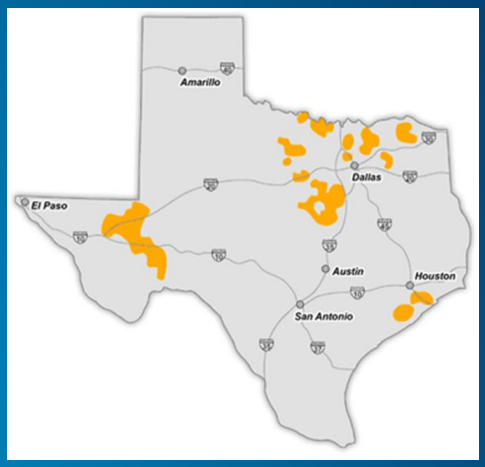
Using Small Data and Simple Apps for Smart Payoffs

Aaron Allen - PNM

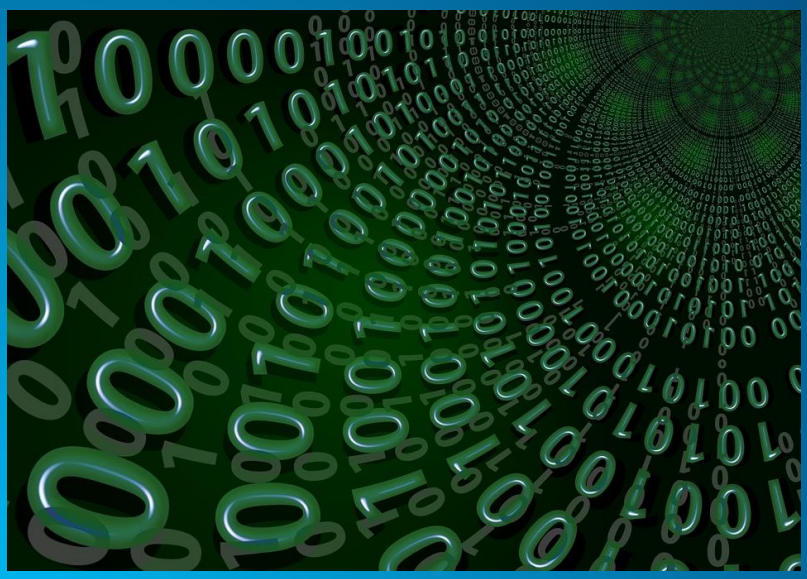
ESRI USER CONFERENCE

Title page

Distribution Service Areas for PNM & TNMP

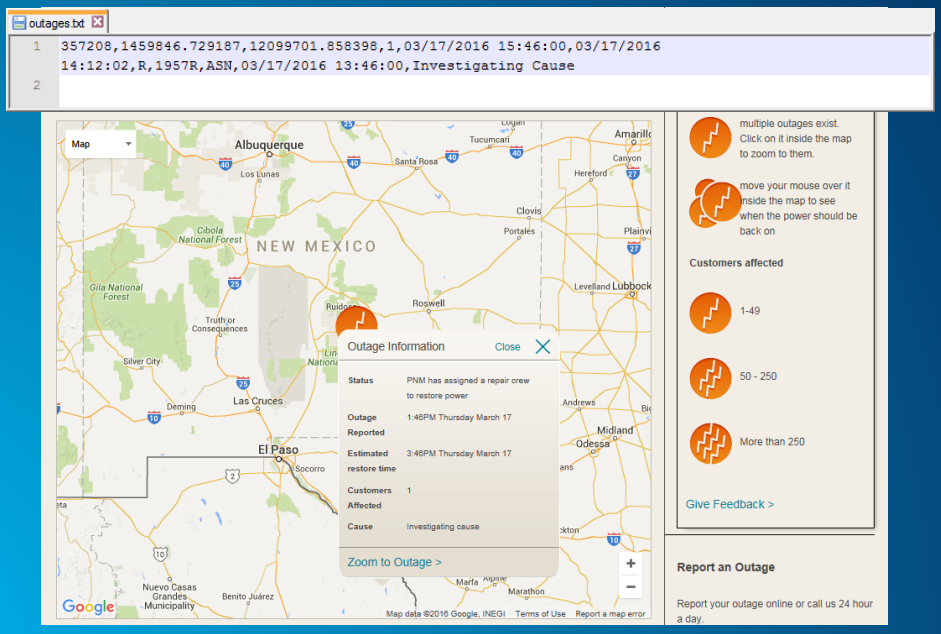


PNM is NM's largest electricity provider, serving more than 500,000 residential and business customers in the state. TNMP is an electricity transmission and distribution service provider serving many communities and more than 500,000 Texas residential and business customers. We have a good-sized GIS: 4 SDE GDBs with numerous feature datasets and inside those even more feature classes, 3 ArcGIS Server environments, a Portal Environment, and we use AGO. We have 10 dedicated GIS employees, several power users in other departments, and numerous employees who use GIS in some manner. In short, PNM has a significant amount of data and a GIS infrastructure available to leverage it. The point, however, is to **make it useful** to our employees, help them do their jobs and serve our customers.



- **If It’s Small**
- **Keep It Simple**
- **Make it Smart!**

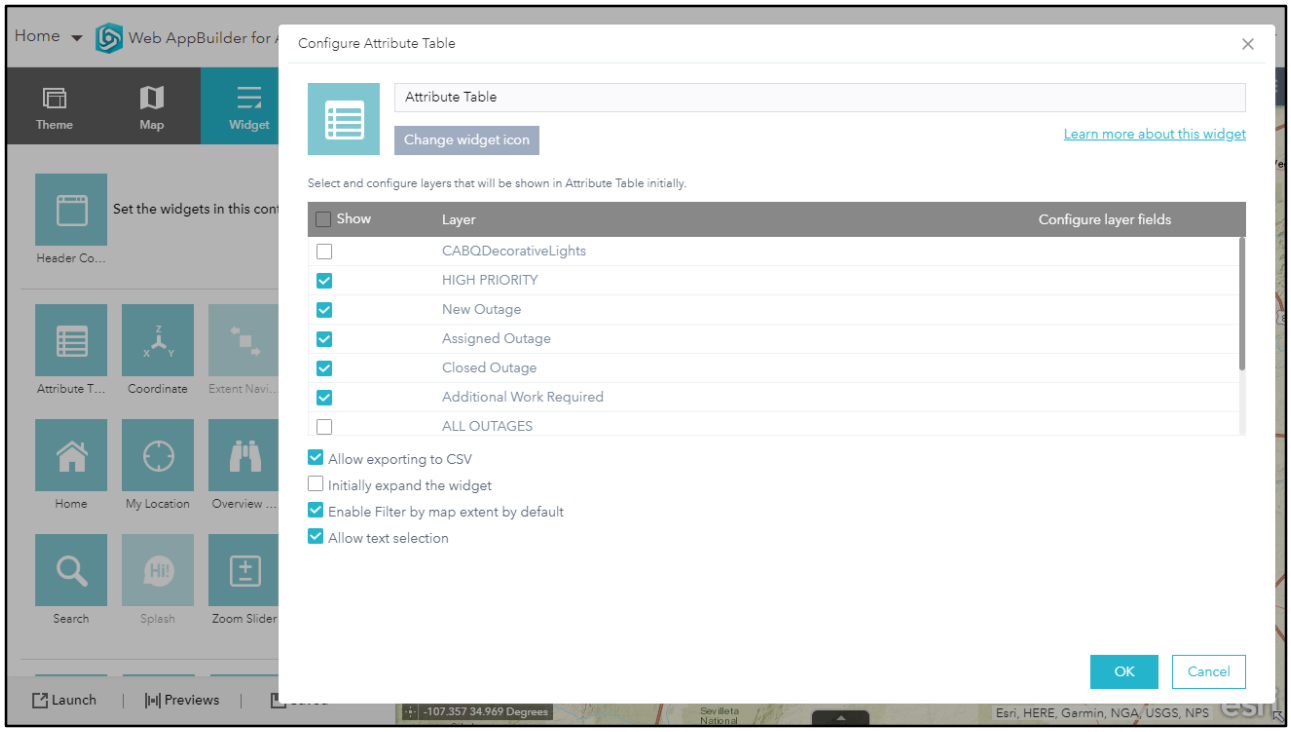
Audience question: How many of you work with feature classes or other data containing billions of records? Hundreds of millions? Big data, along with machine learning and AI, are currently in the media spotlight. GIS for big data is important and has its place but many/most of us work for organizations with more modest datasets (thousands or only hundreds of records), Things like roads, parcels, municipal boundaries, water lines, transmission lines, etc. And while GIS professionals (power users) perform many different and complicated tasks using GIS, our non-GIS colleagues usually don't need or want full blown GIS software. It's too much - and commonly an impediment to adopting GIS. They usually want to accomplish a simple or a limited set of tasks so having a straightforward, simple app or web map can be a great way to help them do their jobs. So if it’s small, keeping it simple often makes it smart.



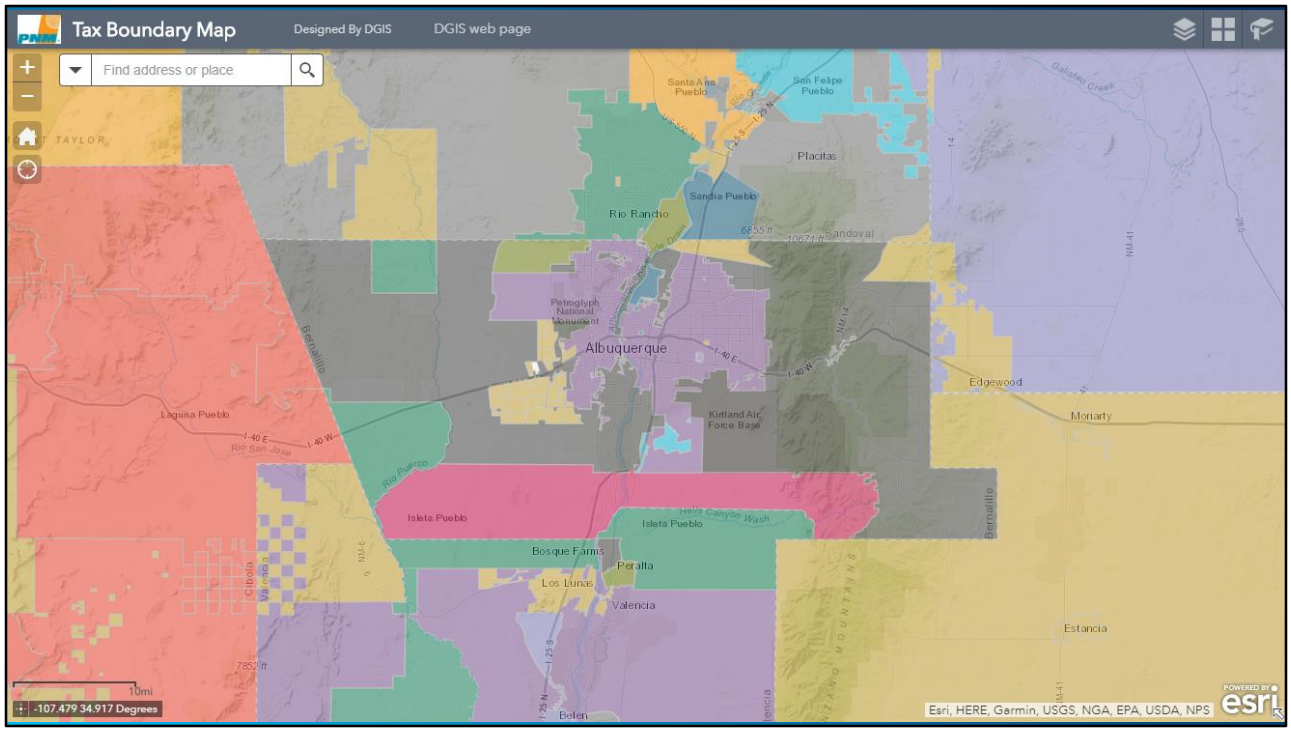
Outage Map: PNM has been using and building (and sometimes paying for) interactive, online maps for several years now. The outage map is an example of a simple web map we developed in-house. (Next slide…)

```
108         }
109         var markers = xml.documentElement.getElementsByTagName(outagesTag);
110         //alert("Total markers: " + markers.length);
111         for (var i = 0; i < markers.length; i++) {
112           var strCustomersAffected = markers[i].getElementsByTagName(customersAffectedTag)[0].firstChild.nodeValue;
113           var strEstTimePowerBackOn = markers[i].getElementsByTagName(estTimePowerBackOnTag)[0].firstChild.nodeValue;
114           var strOutageReported = markers[i].getElementsByTagName(outageReportedTag)[0].firstChild.nodeValue;
115           var strStatus = markers[i].getElementsByTagName(statusTag)[0].firstChild.nodeValue;
116           var strCause = markers[i].getElementsByTagName(causeTag)[0].firstChild.nodeValue;
117           outageLatLng = new google.maps.LatLng(
118             parseFloat(markers[i].getElementsByTagName(yGeoTag)[0].firstChild.nodeValue),
119             parseFloat(markers[i].getElementsByTagName(xGeoTag)[0].firstChild.nodeValue)
120           );
121
122           var outageIconImage;
123           if (parseInt(strCustomersAffected) == 0) { //zero outages
124             outageIconImage = new google.maps.MarkerImage("http://www.pnm.com/customers/outage_map/images/map/zero_outages.png", null, null, new
              google.maps.Point(52,52));
125           } else { //otherwise use icons based on number of outages
126             if (parseInt(strCustomersAffected) < 50) { //less than 50 customers
127               outageIconImage = new google.maps.MarkerImage("http://www.pnm.com/customers/outage_map/images/map/1_lightning_bolt.png", null, null, new
                google.maps.Point(14,14));
128             } else if (parseInt(strCustomersAffected) > 250) { //more than 250 customers
129               outageIconImage = new google.maps.MarkerImage("http://www.pnm.com/customers/outage_map/images/map/3_lightning_bolts.png", null, null, new
                  google.maps.Point(19,19));
130             } else { // 50 - 250 customers
131               outageIconImage = new google.maps.MarkerImage("http://www.pnm.com/customers/outage_map/images/map/2_lightning_bolts.png", null, null, new
                  google.maps.Point(16,16));
132             }
133           }
134           outageMarker = new google.maps.Marker({
135             map: map,
136             position: outageLatLng,
137             icon: outageIconImage
138           });
139           markersArray.push(outageMarker); //add each marker to the markersArray for clustering below
140           bindOutageMarkerInfoWindow(outageMarker, map, strCustomersAffected, strOutageReported, strEstTimePowerBackOn, strStatus, strCause);
141         }
142
143         var clusterStyles = [[{
144           url: '../images/map/multiple_outages.png',
145           height: 37,
146           width: 45,
```

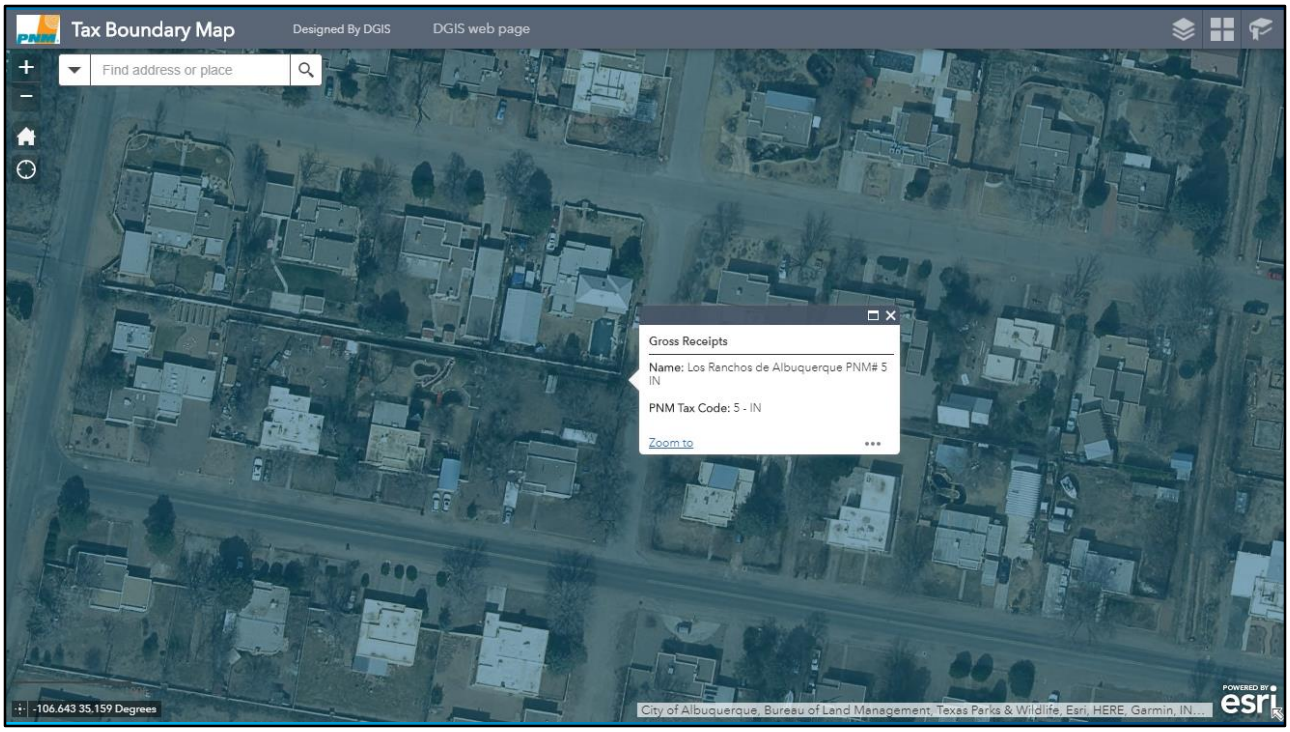
Outage Map (con't): But it required custom code to build (Google Maps JavaScript API) - not simple. Not everyone codes so app development necessarily limited. (Next slide…)



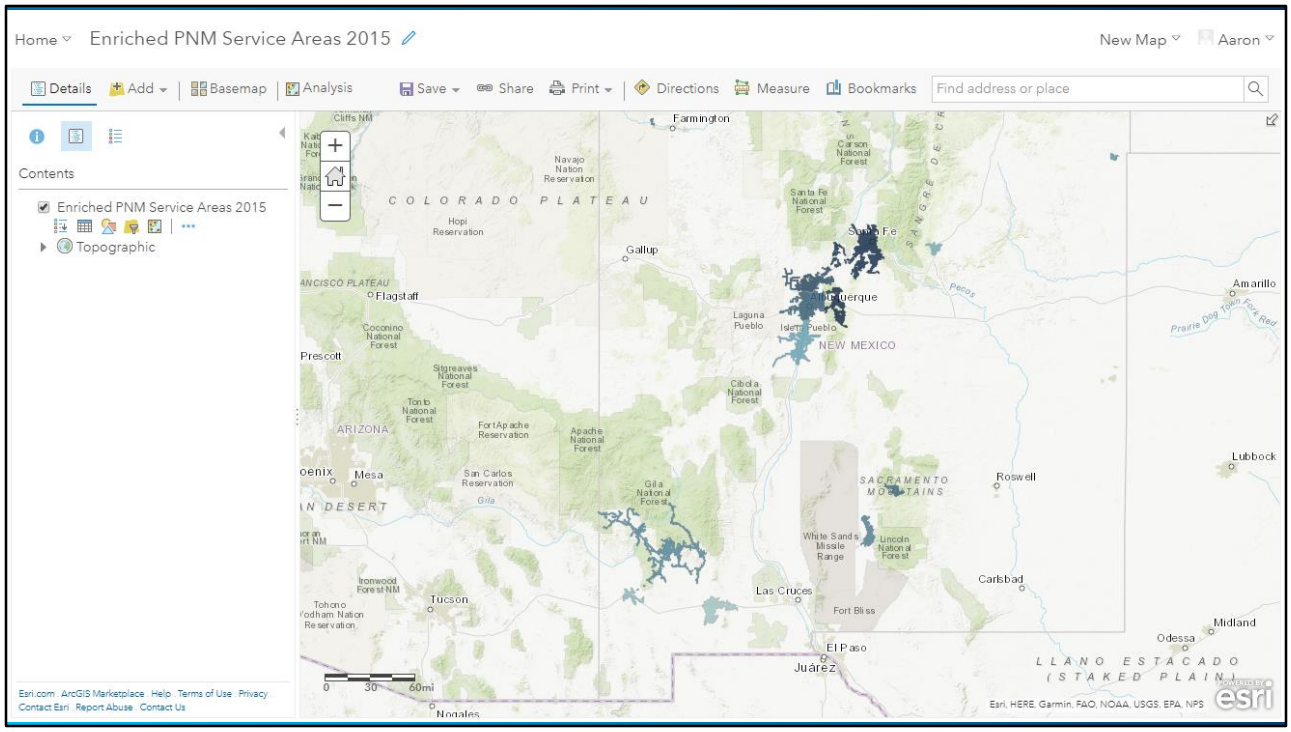
Outage Map (con't): Enter AGO/Portal. Templates emphasize configuration, which makes it much easier to deploy apps. Many basic tools/functionality are available although there are limitations (option to code if you want).



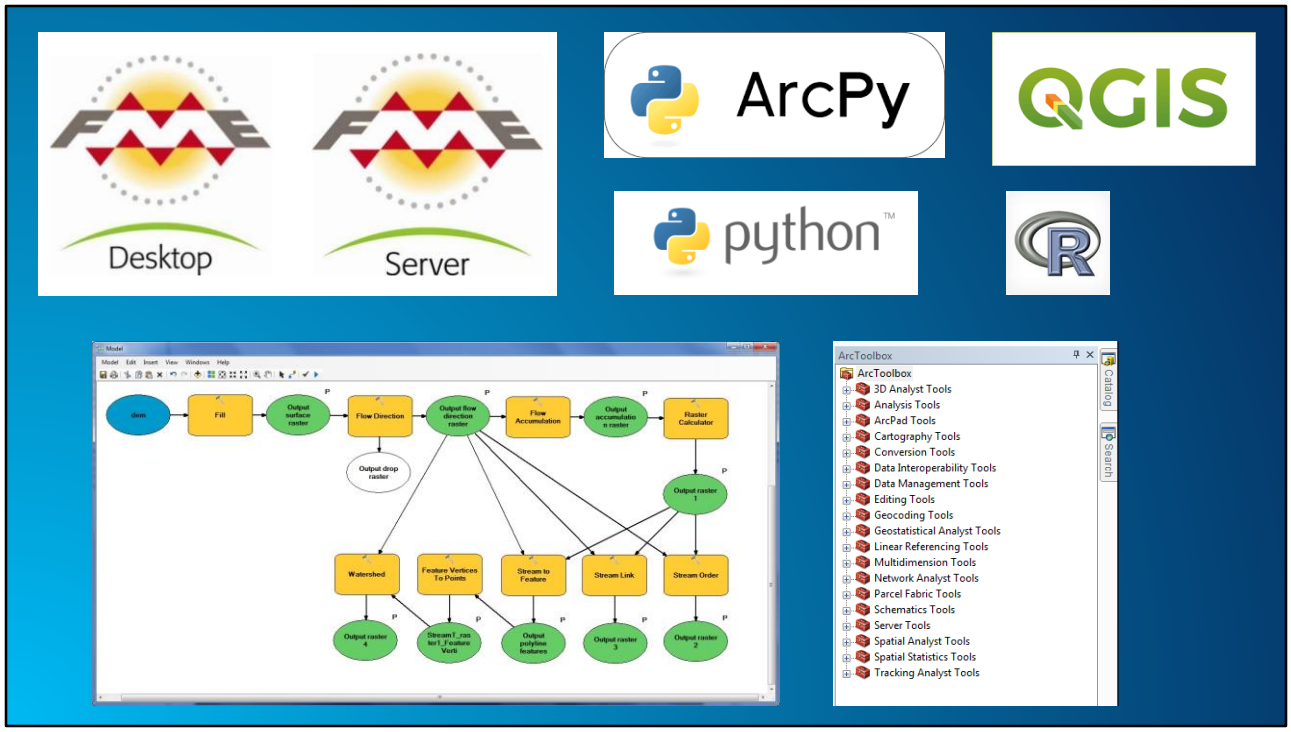
Tax Boundary Map (on Portal): PNM has a number of web maps/apps but sometimes they are more than someone needs to accomplish a task. For instance, we have a "web viewer" of our distribution system that has a ton of layers and lots of tools and functionality. It's useful to people who work extensively and in different ways with distribution data but it can be overwhelming for people don't use GIS much. That's why we built the Tax Boundary Map. (Next slide…)



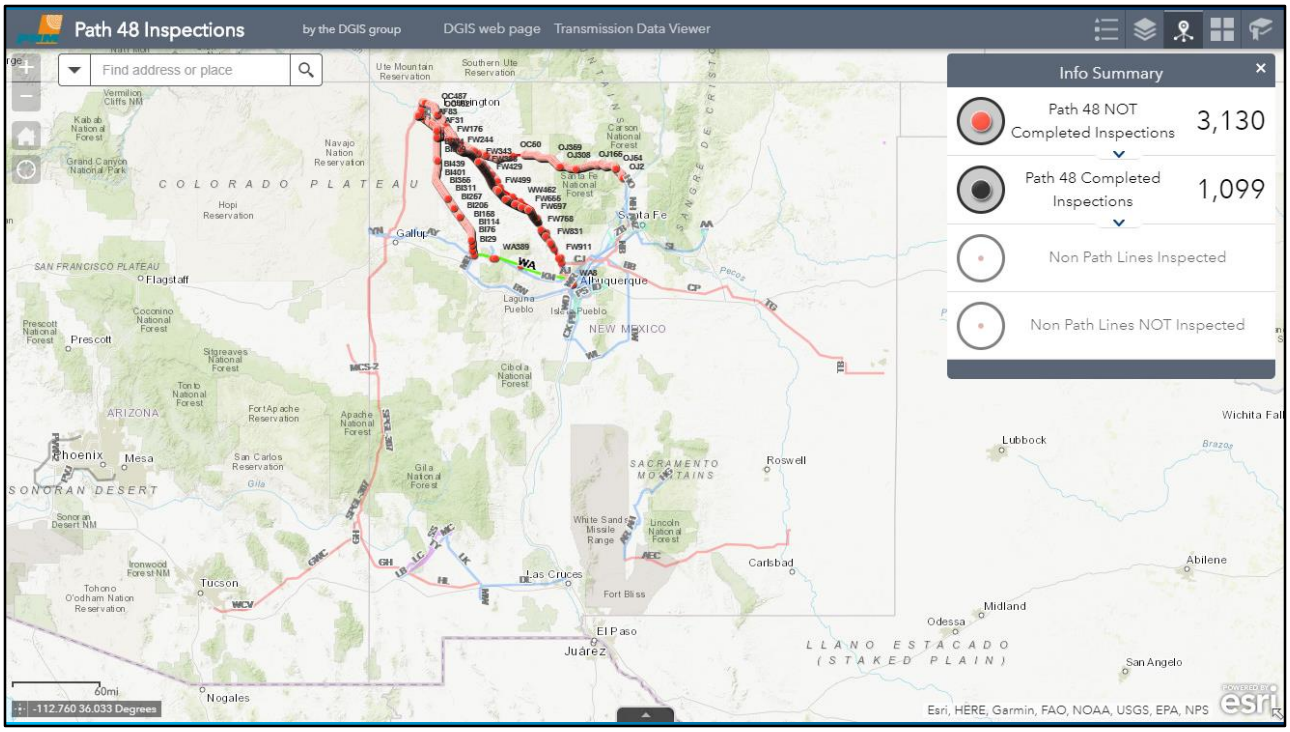
Tax Boundary Map (con't): It does one simple but useful task - clicking the map displays the tax code and in/out indicator at a specific location. It's an important tool for getting accurate gross receipts tax info for new customers. It's helped reduce billing errors thereby saving PNM money and making customers happier (or less ticked off). (small, very simple, but smart)



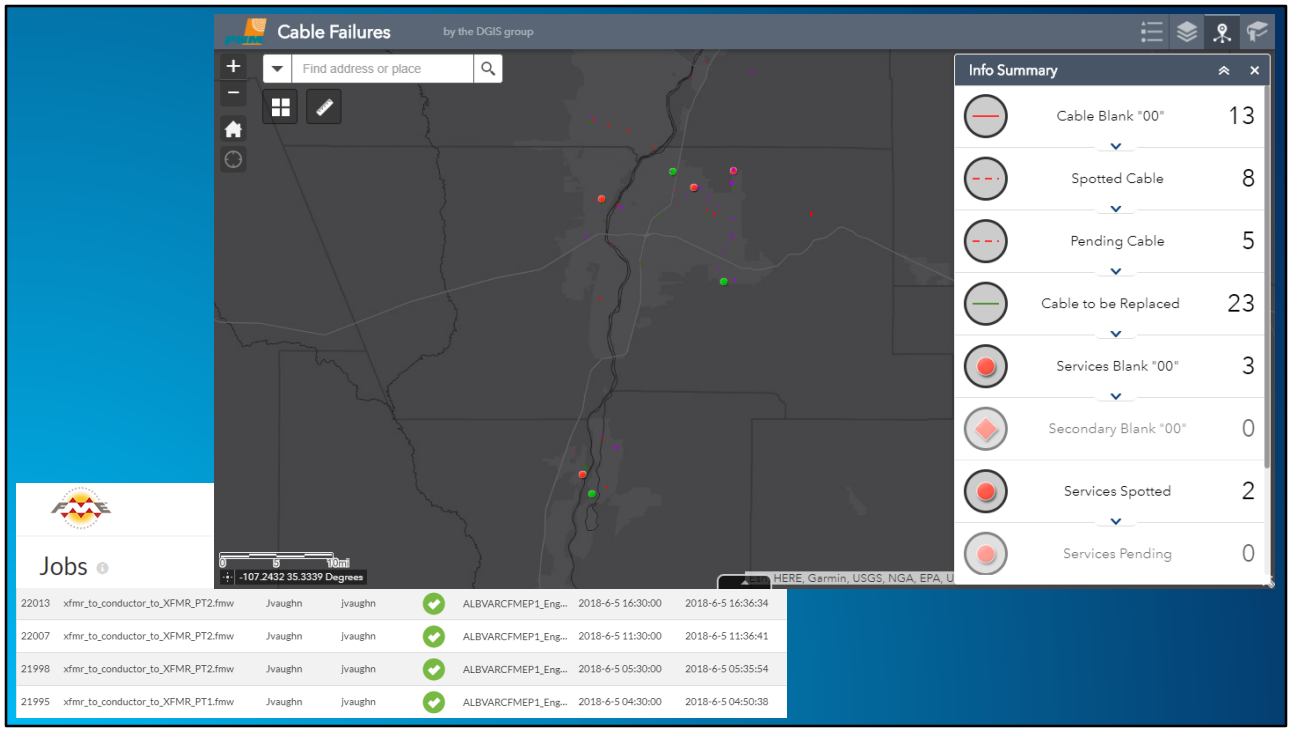
Enriched PNM Service Areas: An early, simple web map created to illustrate economic differences between the regions PNM serves. ESRI's Enrich Layer Tool was used to add economic data to the regions. The goal was to help the marketing department better understand the communities PNM serves. The map corrected an assumption about which areas were poorest (assumed to be Alamogordo but is not) and therefore helped better direct services such as budget billing to those customers. (small, simple, and smart)



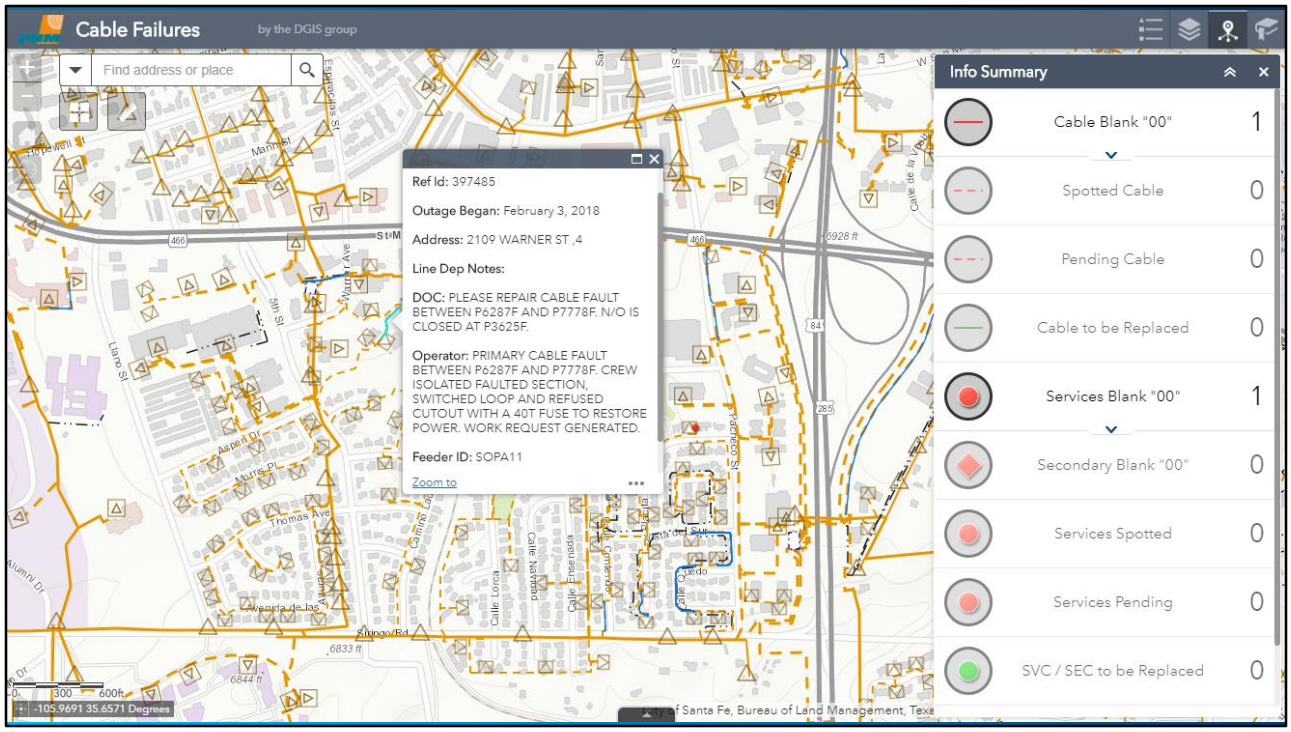
Having a simple app or web map powered by "small data" does not necessarily mean you have simple data or processes behind it. We all know that getting good quality data takes effort. And, a map is relatively worthless without the right data behind it. The GIS professional often needs to convert, clean up, transform, update, and and/or QA/QC data to "get it right". In addition, our data is sometimes tied to multiple systems and/or complex processes, and these relationships need to be accounted for when creating, maintaining, and updating data. At PNM, we often use Safe's FME Desktop and FME Server to "get our data right." We've found it to be a great product. They are an ESRI partner BTW. ESRI's Data Interoperability Extension is basically FME. There are other options too such as ESRI's Model Builder, Python (ArcGIS, QGIS), R, GDAL, etc. Use what works for you.



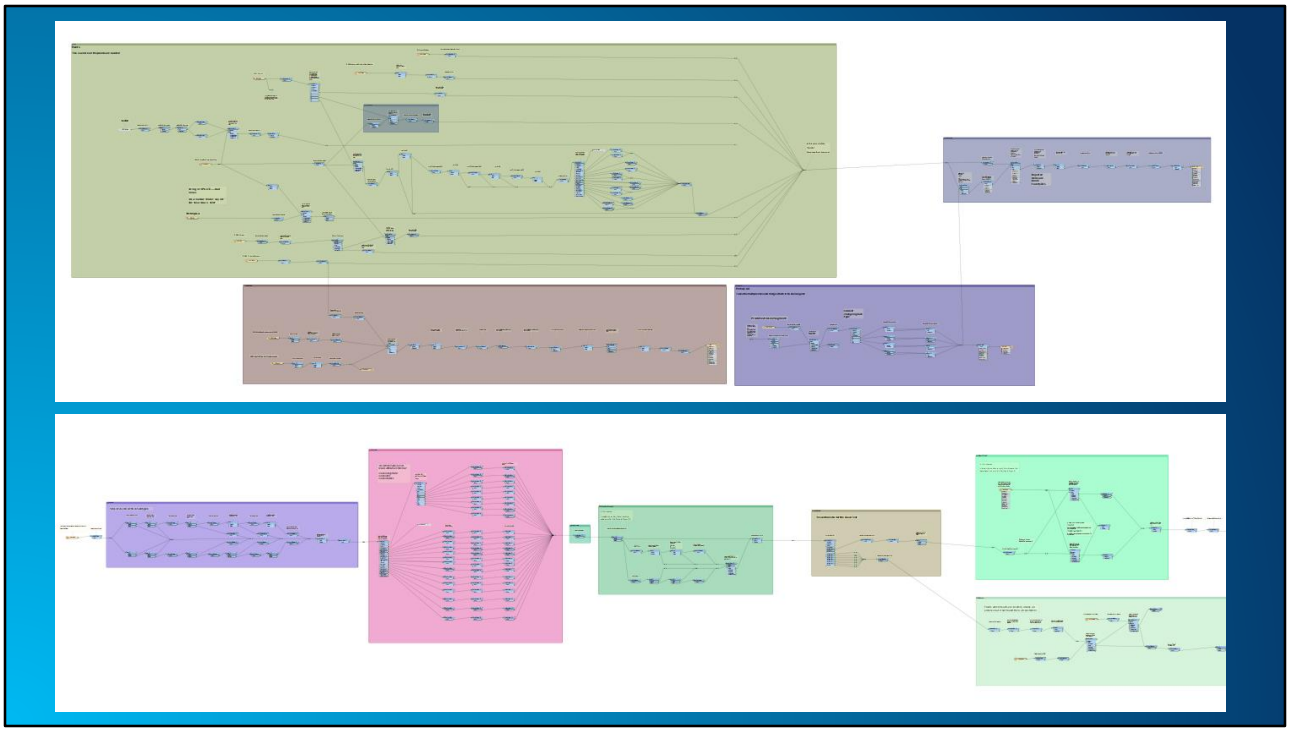
Path 48 Inspections (for NERC and FERC reporting): An inspections table is joined to transmission structures using FME. It's updated daily and pushed to Portal. Completed inspections are displayed on a map, a visual making it much easier to see what's going on rather than poring over a table. The app illustrates progress, promotes syncing (getting field guys to sync laptops), and facilitates directing work where needed. It's a simple app with GIS complexity (the smart part) transparent to the end user. In other words, FME and the GIS professional do the work so the end user doesn't have to. (NERC – North American Electric Reliability Corporation; FERC – Federal Energy Regulatory Commission)



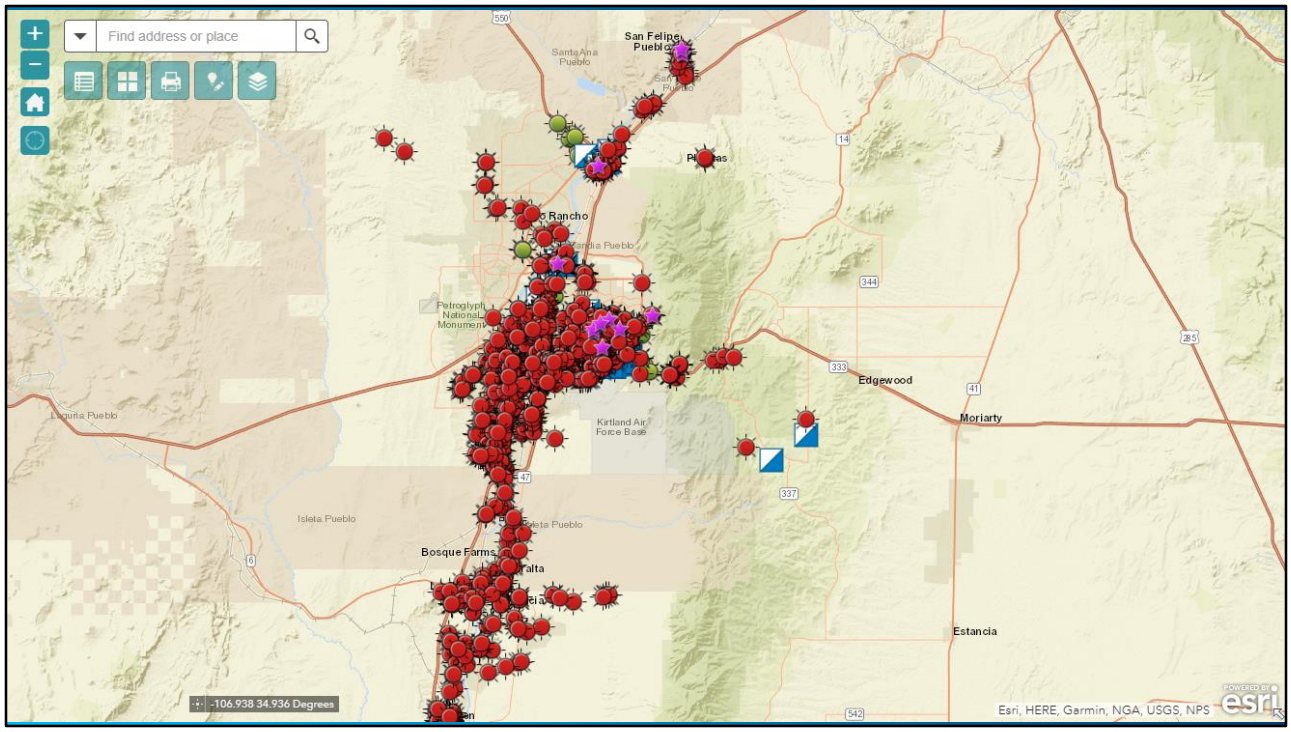
Cable Faults: An easy to use app for viewing cable faults but a complex process to assemble the data. Source data, tabular data is read from an Access DB and FME cleverly used to create segments or, if not possible, points with relevant attributes for display. The GIS data is assembled once and cable faults updated 3x/day on FME Server. (Next slide…)



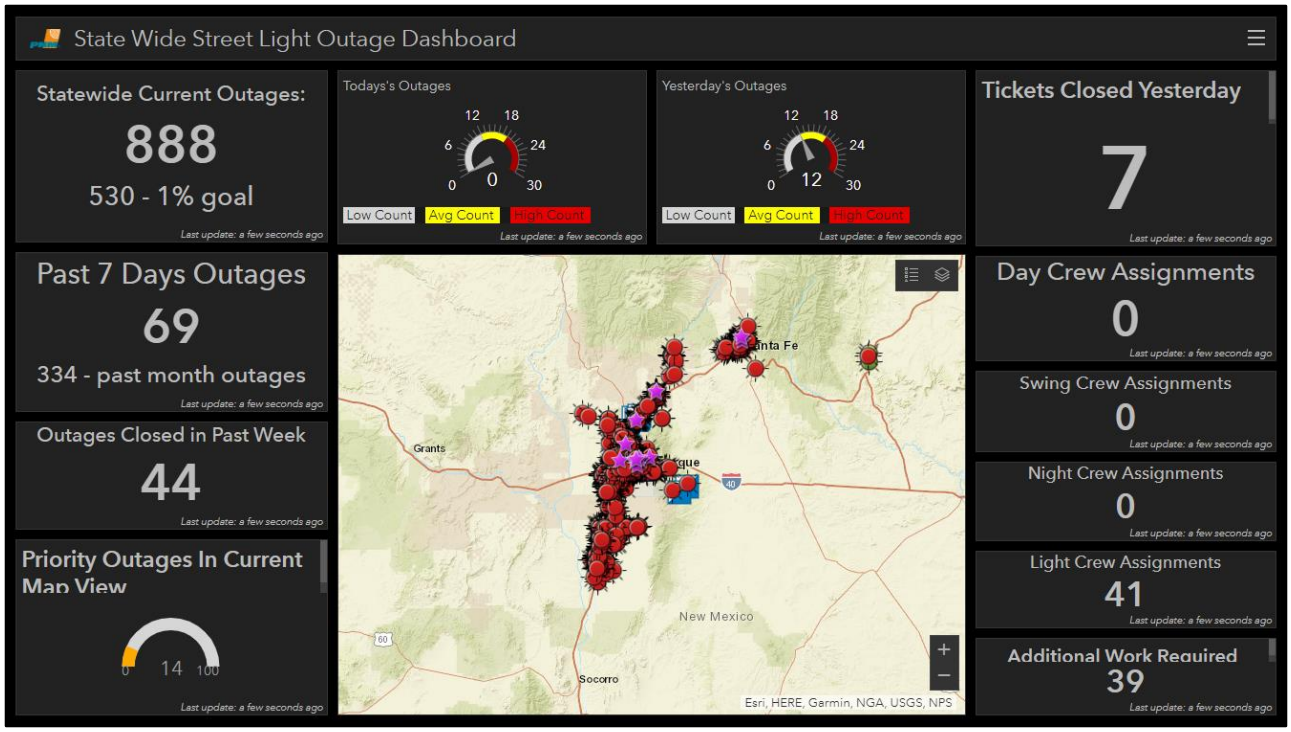
Cable Faults (con't): The app is used by a manager and her field crews to better direct and prioritize work. Again, simple but smart app with GIS complexity transparent to the end user. (Next slide…)



Cable Faults (con’t): And this is what’s transparent to the end user. FME and the GIS professional do the work so the end user doesn't have to (or even can’t). Common end user question: “Can you…?”



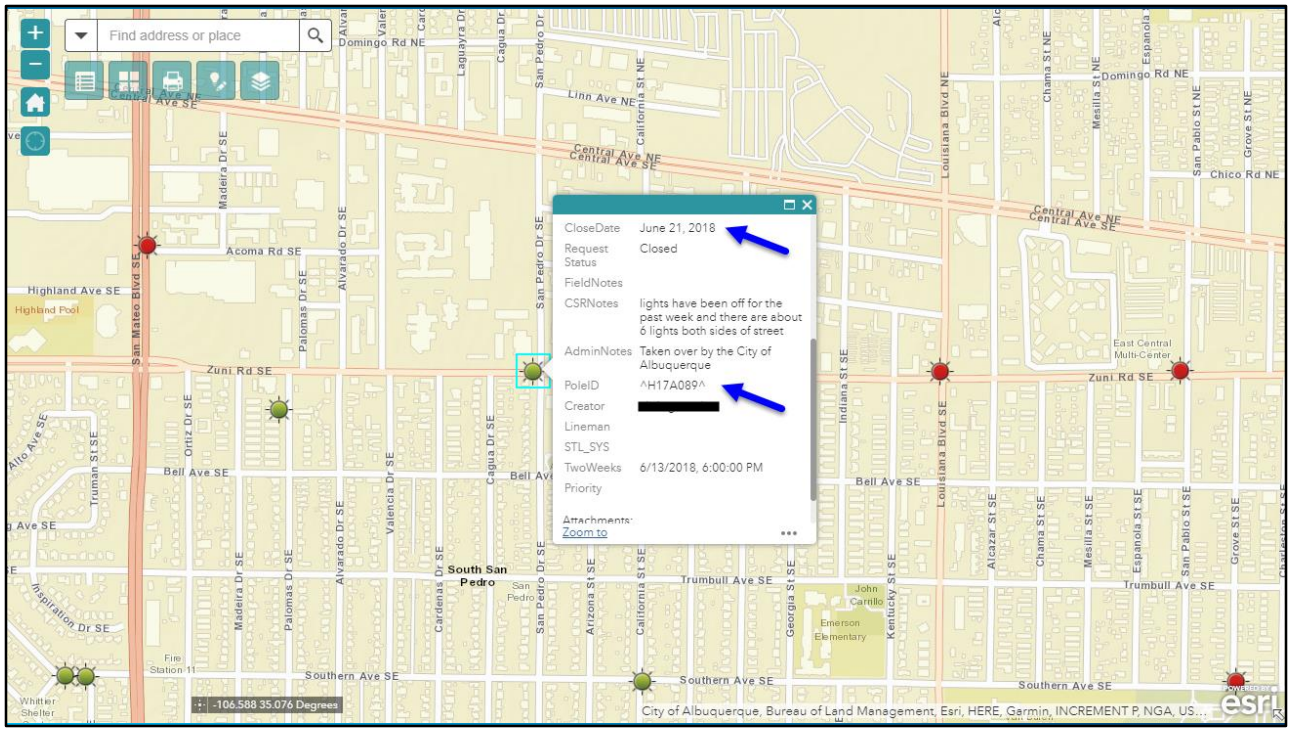
Street Outage Reporting Tool (SORT): Example of small data (a single feature layer) that has been leveraged in multiple ways. It's been so successful (smart!) it has become the system of record for streetlight outages and there is now a call to tie it to other systems, including billing and updating streetlight data (not just outages). (Next slide…)



SORT (con't): Streetlight outages feature layer may be one simple layer but it powers 3 web apps (one of which was on the previous slide) and a dashboard. In other words, the data is leveraged to be useful in a variety of ways – recording outages (CSRs), scheduling repairs (manager), fixing outages (field crews), and monitoring outages (managers). (Next slide…)

PowerClerk To ArcFM	Aallen	2018-03-12 04:45:00	N/A	0 45 04 ? * MON-FRI	PowerClerkToArcFM.fmw	
Streetlight Outage Report	Streetlights	2017-06-19 05:00:00	N/A	WEEK * 1	STREETLIGHT_REPORT_4_Weeks_FME_2016.fmw	
Streetlights - Backup AGO Streetlight Request Data	Streetlights	2016-09-01 02:00:00	N/A	DAY * 1	Backup_AGO_Streetlight_Request_Data.fmw	
Streetlights - Calculate TwoWeeks Field on AGO	Streetlights	2016-08-25 02:30:00	N/A	DAY * 1	Calc_TwoWeeksAgo_AGO.fmw	
Streetlights - Create THOR Files from AGO	Streetlights	2016-10-04 04:30:00	N/A	DAY * 1	THOR REPLACEMENT FILE.fmw	
Streetlights - Get PoleID and Set CloseDate on AGO	Streetlights	2016-09-01 00:05:00	N/A	HOUR * 1	Get_PoleID_and_Set_CloseDate_AGO.fmw	
Streetlights - incremental updates	Streetlights	2016-08-22 21:00:00	N/A	DAY * 1	Incremental_updates_to_AGO_STREETLIGHTS.fmw	

SORT (con't): Behind the scenes there are several (6) processes on FME Server that support it (again, complexity is behind the scenes, left to GIS professionals). (Next slide…)



SORT (con't): For instance, we use FME to automatically find the closest pole to an outage when a new streetlight outage is added to the map. This makes it easier to find poles in the field and know what materials are needed to fix a streetlight that is out. FME is also used to automatically add the Close Date when a ticket is closed. Note that FME reads from and writes directly to AGOL and Portal.

We use AGO/Portal to support short term projects but increasingly/more so institutional, day-to-day work processes. After experimenting with it for a couple years, things are really taking off this year. It's becoming a great way to serve our colleagues and customers (i.e. - making GIS data <u>increasingly</u> useful). And behind the scenes, we often use FME to do the heavy lifting of data analysis, assembly, and conversion. I should also note that not only do we run FME manually on desktops as requested or needed, we also use FME Server to automate processes (currently 14 processes running on FME Server hourly to weekly).

While small and simple isn't appropriate everywhere, there are countless possibilities where it can help your (often non-GIS) colleagues accomplish goals and/or provide additional value (smart!).

Questions?