Polygons and Persistence

Mapping TxDOT's Right of Way One Segment at a Time

Texas Department of Transportation

11/12/2025


HELP MAKE TEXAS SAFER FOR EVERYONE
DRIVE like a TEXAN
Kind. Courteous. Safe.
DriveLikeATexan.com

Welcome!

Drive safely out there, you're our only hope…

The Purpose of this Presentation

The goal for this presentation is not to focus on the ROW GIS product, but rather the multiyear process to build it. There were two distinct phases to this project

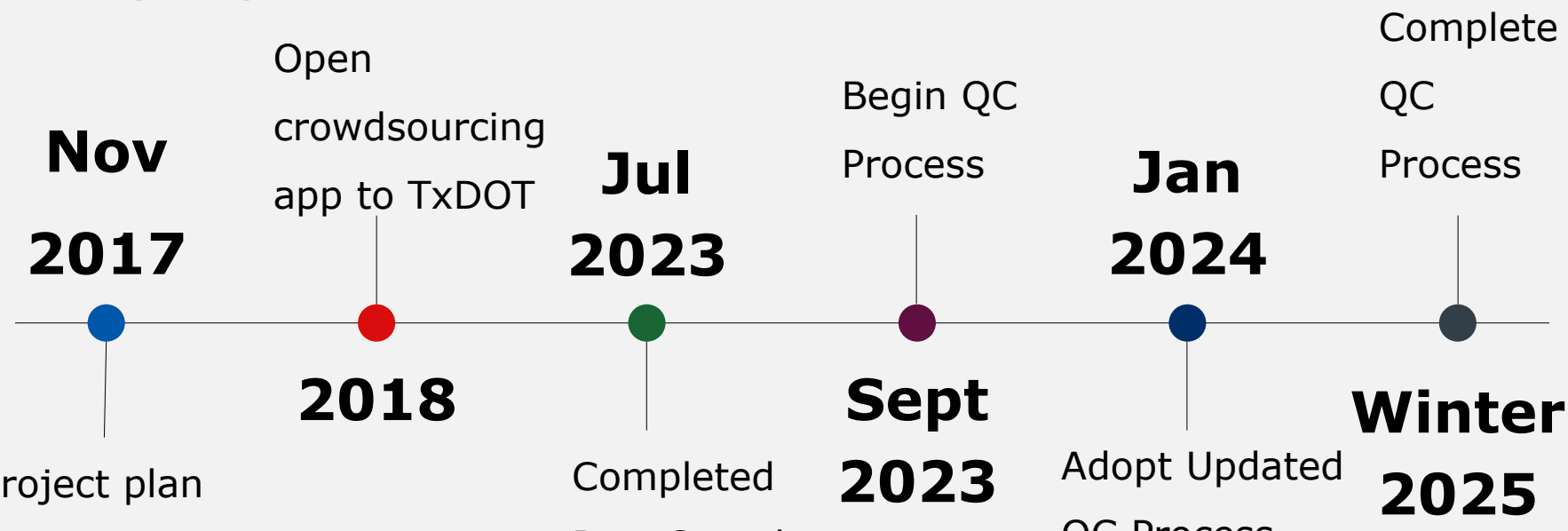
- Internal crowdsourcing app to approximate delineations for the ROW
- QC those delineations using reliable sources including TxDOT's existing survey grade ROW, Regrid, StratMap, aerial imagery, etc.

Start With Some Questions

How does TxDOT obtain a mapping grade delineation of the Right of Way (ROW) on its managed roadways?

- What GIS layers do we currently have?
  - Existing delineations where projects occurred
- Dataset will help answer questions related to approximate area, assets in the ROW, etc.
- ROW Division has the information, but not in an easy to translate GIS format

Timeline


Nov 2017
Project plan created for Row Crowd
2018
Open crowdsourcing app to TxDOT
Jul 2023
Completed Row Crowd Effort
Sept 2023
Begin QC Process
Jan 2024
Adopt Updated QC Process
Winter 2025
Complete QC Process

ROW Crowd

ROW Crowd – the beginning

The initial part of the project called for as many users as possible to delineate the approximate ROW boundaries. Participants would attended editing sessions to learn the process and ask questions. This was an internal effort to TxDOT.

ROW Crowd – the beginning

An underlying point layer along the roadway drove the application. Along with limited map controls, it provided a focused app for the user.

- The user was presented points at random
- Users were taught to use landmarks and Google Maps
- No zoom or pan, just draw and go

The QC Process

The QC Process

The layer that came out of the ROW Crowd effort was messy and had to be organized.

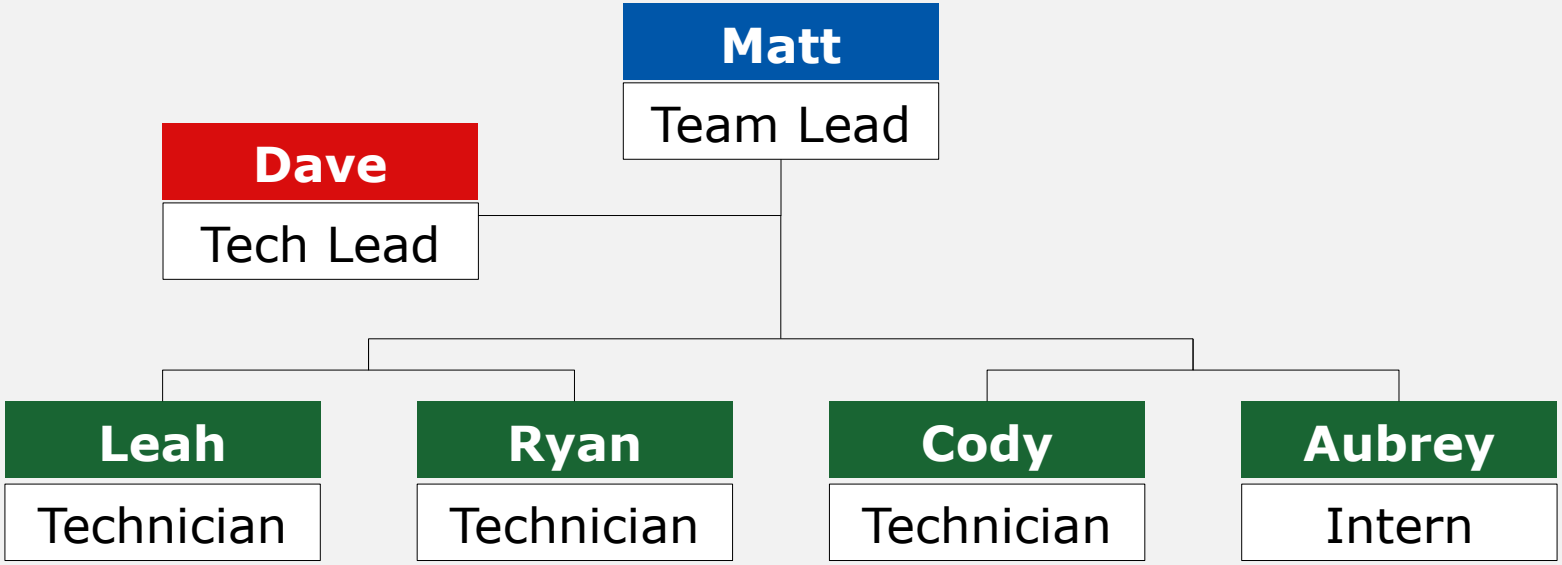
- The data had to be split
- A unique identifier had to be created
- A merge had to occur

The QC Process

After merges, dissolves, and clips, the ROW QC Edits layer was created.

- More manageable features to work with
- Created HWY_CNTY field which was the unique identifier
- Split by TxDOT Districts
- The layer was moved to an SDE

ROW QC Team



Previous Technicians: Kassandra & Easton

The QC Process – first go

The initial QC process involved working with replicas of the data outside of the Enterprise Geodatabase (SDE), then merging those edits back into the database.

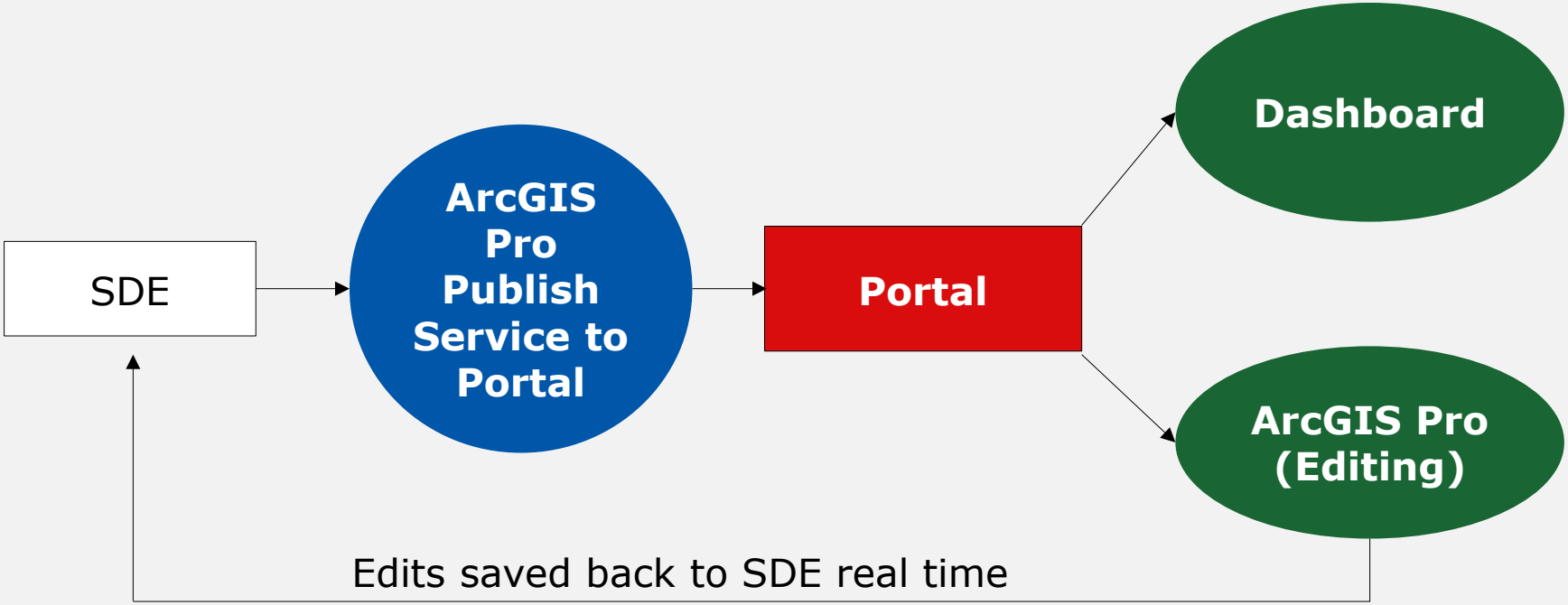
- Proposed due to latency issues working in the SDE
- Was not ideal due to copying of GDB's and relying on one individual to merge the data back into the SDE
- This process lasted for ~2 months

The QC Process – figured it out

While the initial QC process allowed the project to start, a better solution was needed. Everything changed when we switched to a custom datastore and published from our SDE to our Portal. We also setup archiving on the SDE.

- Technicians could now edit a service, instead of needing to create replicas
- This was a great workaround to the latency issues faced when working straight out of the SDE
- We initially were working out of our test Portal, but migrated to production after approval

ROW QC Data Flow Chart



The QC Process - strategy

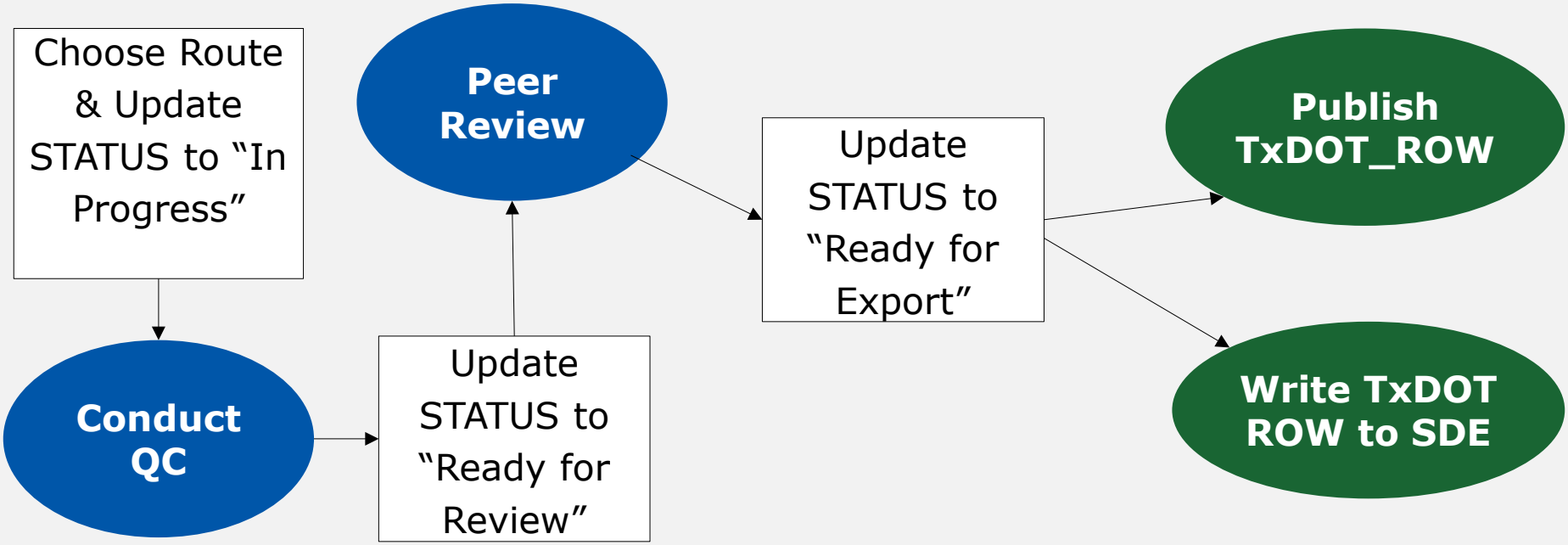
We developed a strategy and best practices for editing.

- Each technician was assigned a district
- Use existing ROW boundaries, published boundary data (StratMap, Regrid), aerial imagery
- Follow the routes hierarchy, working starting at the top

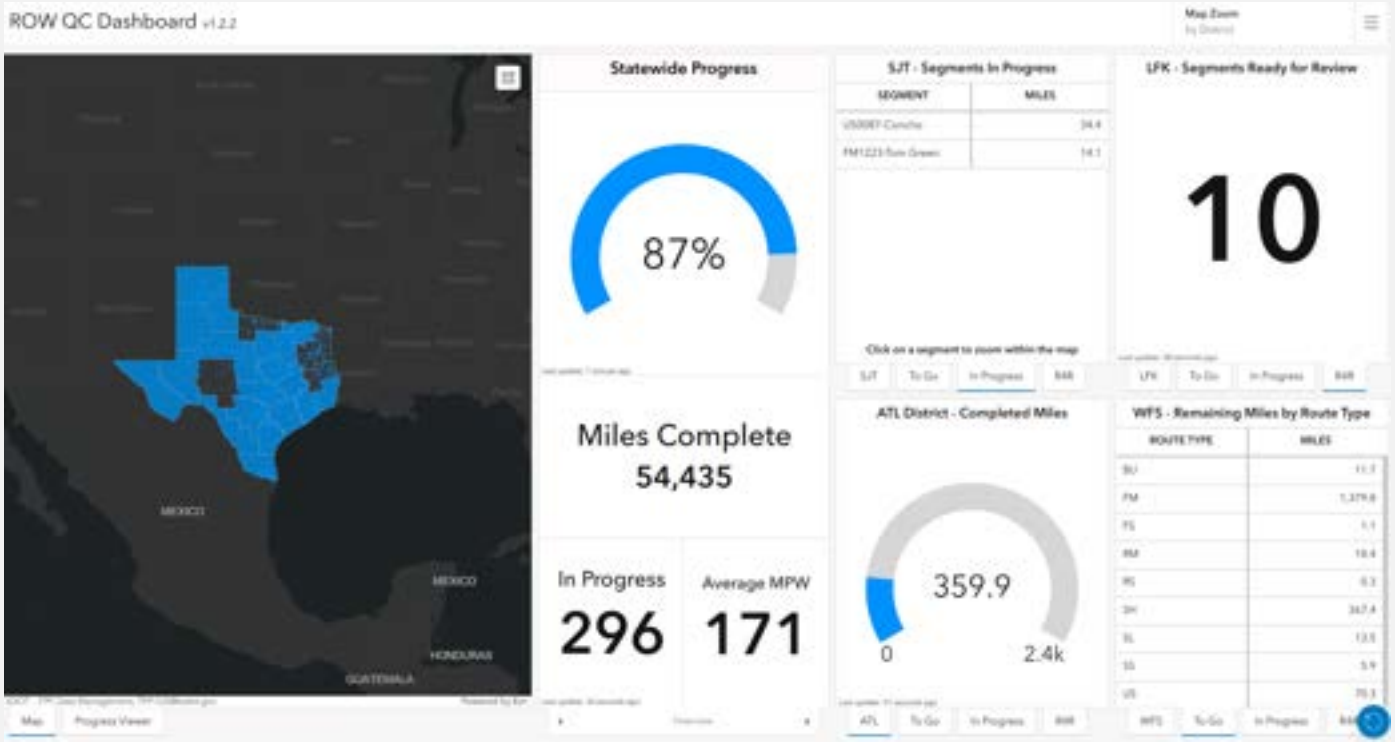
The QC Process - strategy

- Use a STATUS field with domain values (Not Started, Ready for Review, etc.)
- Use available editing tools in ArcGIS Pro (Generalize, Edit Vertices, Reshape, Clip)
- Bi-weekly (switched to weekly) edit session meetings to discuss editing techniques that worked, issues, or show troublesome areas
- Develop an operations dashboard to show progress

ROW QC Workflow Chart



The QC Process – ROW QC Dashboard



FME

Can We Use FME to help QC?

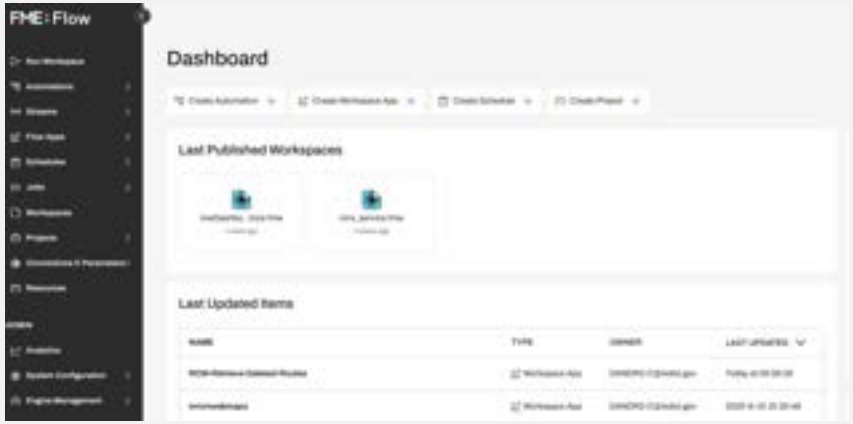
YES - four reasons why:

- Able to host on FME Flow
- Setup schedules for consistency
- Sprinkle in Python for some complicated solutions
- FME Flow apps (because I take vacations)

FME Flow Apps

FME Flow checked all the boxes:

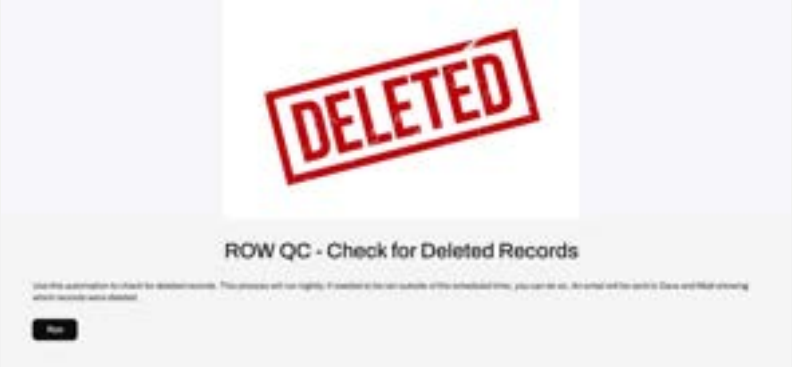
- Easy to work with
- Able to read / write to an SDE & Portal
- Reliable scheduling task
- Ability to build Flow Apps to allow on-demand automations to be ran



FME Flow Apps

The Flow Apps we used:

- Retrieve deleted records
  - Enter the last known time the record existed
  - Enter the HWY_CNTY
- Identity deleted records
  - No parameters, simply sends email noting HWY_CNTY of record removed



FME Flow Automations

We used automations to:

- Capture progress metrics
- Assign reviewers to routes
- Log edits
- Update the Portal service with completed routes

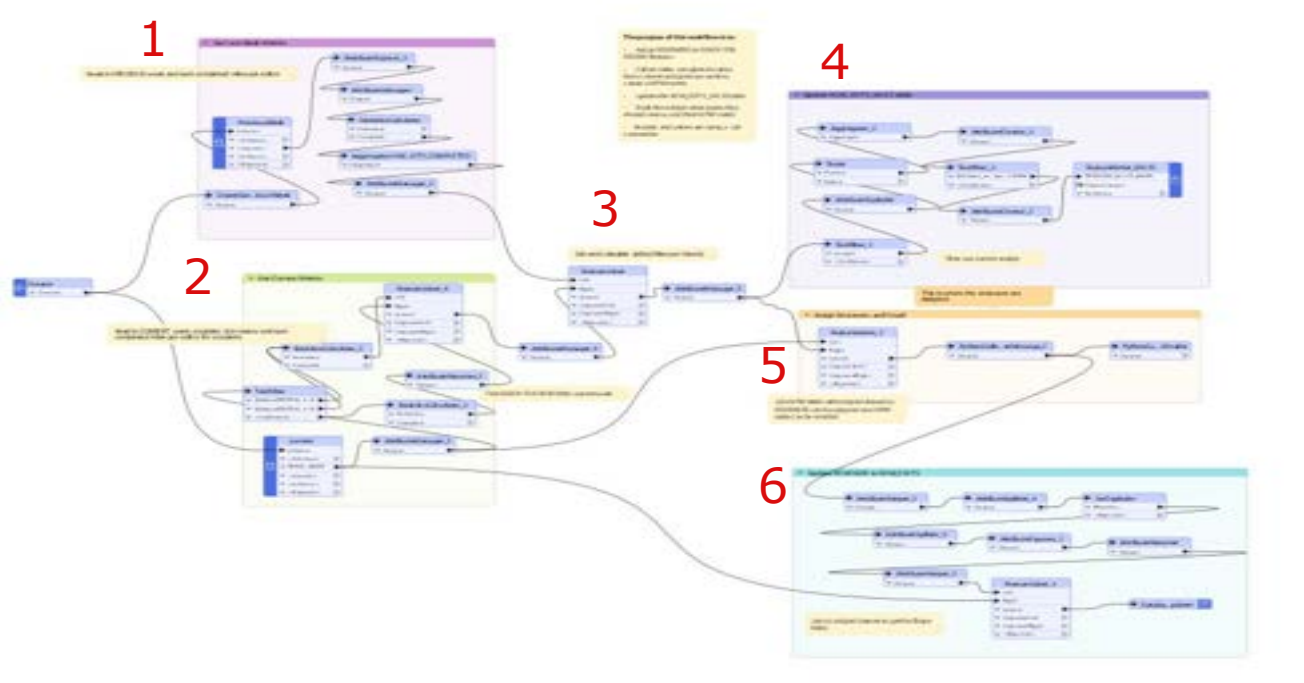
FME Flow Automations - Metrics

There was one automation that stood above the rest…

- Capture progress metrics, comparing previous week to current week
- Auto assigned reviewers for each route marked as ready for review
- Emailed results of progress and assigned routes to QC to the team

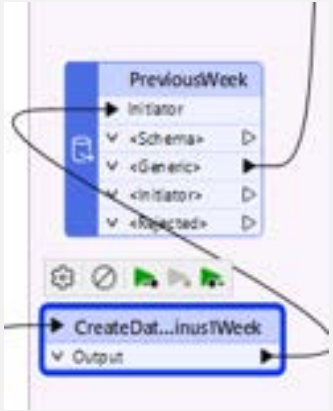
FME Flow Automations - Metrics

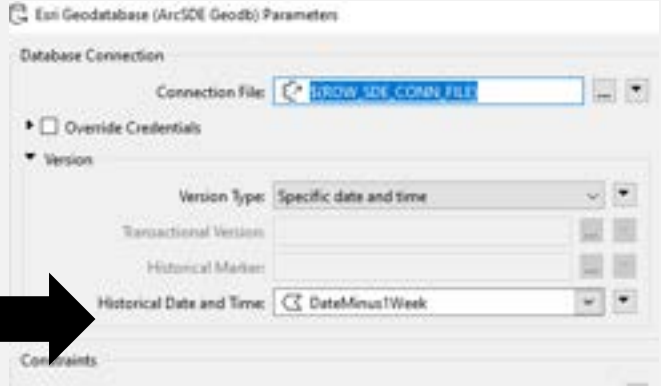
Mandatory FME Workbench screenshot…



FME Flow Automations - Metrics

We wanted to compare previous week's miles with the current week's miles. This was accomplished by creating a parameter and using it in the reader.






```
@DateTimeFormat(@DateTimeAdd(@DateTimeNow(),-P7D),%Y-%m-%d 00:00:00 )
```

FME Flow Automations - Metrics

The ability to compare last week's miles with this week's miles allowed us to show weekly progress to the technicians

CB - Current Progress. Miles marked as complete for the week of 09/28/2025

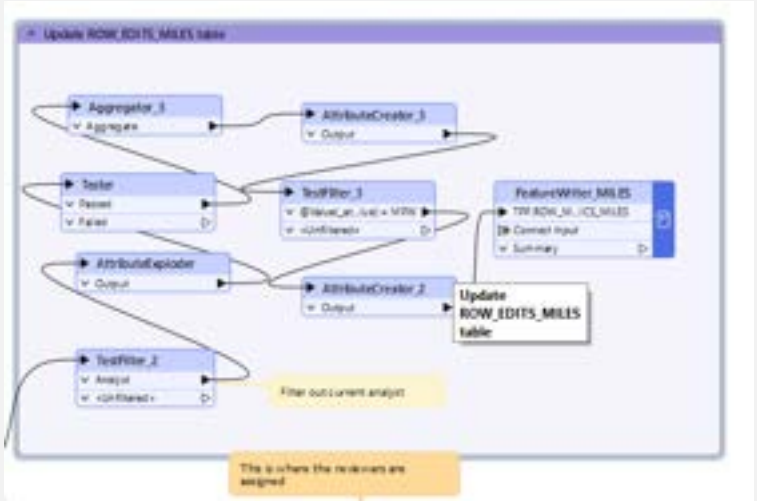
USER	PREV_COMPLETED_MIL	CURR_COMPLETED_MIL	MPW	CURR_MIL_READY_FOR_REVIEW
CB	17018	17141	124	268

LL - Current Progress. Miles marked as complete for the week of 09/28/2025

USER	PREV_COMP LETED_MIL	CURR_COMPLETED_MIL	MPW	CURR_MIL_READY_FOR_REVIEW
LL	16449	16643	194	237

FME Flow Automations - Metrics

We wanted to capture progress metrics to better help predict how long the project would take.



CB	LL	KM	WK_SUM	WK_AVG	Week Ending
256	375	145	776	258.7	9/1/2024
129	290	113	532	177.3	8/25/2024
166	125	230	521	173.7	8/18/2024
339	264	296	899	299.7	8/11/2024
118	92	92	302	100.7	8/4/2024
235	105	107	447	149	7/28/2024
144	130	242	516	172	7/21/2024
79	183	133	395	131.7	7/14/2024
229	132	277	638	212.7	7/8/2024
176	101	275	552	184	6/30/2024

FME Flow Automations - Metrics

We wanted to auto assign reviewers for the routes marked as ready for review.
Below is an example table of what each technician would see emailed to them.

CB - Assigned to QC

USER	OID	HWY_CNTY
LL	1373	FM0765-Concho
LL	4718	RM0385-Kimble
RS	4598	IH0030-Titus
RS	4660	IH0369-Bowie
RS	6193	US0059-Harrison
RS	6230	US0067-Bowie
RS	6233	US0067-Cass
RS	6245	US0067-Morris

LL - Assigned to QC

USER	OID	HWY_CNTY
CB	6201	US0059-Polk
CB	6378	US0084-Shelby
RS	6188	US0059-Bowie
RS	6189	US0059-Cass
RS	6198	US0059-Marion
RS	6200	US0059-Panola
RS	6251	US0067-Titus
RS	6274	US0071-Bowie

RS - Assigned to QC

USER	OID	HWY_CNTY
AS	6574	US0287-Wichita
CB	6186	US0059-Angelina
CB	6199	US0059-Nacogdoches
CB	6203	US0059-San Jacinto
CB	6204	US0059-Shelby
LL	4717	RM0337-Real
LL	4738	RM0674-Edwards
LL	4835	RM2083-Croc kett

FME Flow Automations - Metrics

The reviewer assignments were made in FME using a Python Caller.

```
def distribute_work(self):
    for i in self.feature_list_shuffled:
        if i["STATUS"] == '3':
            user = i["USER"]
            analyst = self.get_next_analyst(user)

            if analyst is not None:
                self.analyst_dict[analyst].append(i)

def get_next_analyst(self, user):
    # shuffle analysts for random order of assignment
    analysts = list(self.analyst_dict.keys())
    random.shuffle(analysts)

    # filter out the user's own assignment
    remaining_analysts = [analyst for analyst in analysts if analyst != user]

    if not remaining_analysts:
        return None

    # select an analyst with the fewest tasks from the remaining analysts
    min_tasks_analyst = min(remaining_analysts, key=lambda x: len(self.analyst_dict[x]))

    return min_tasks_analyst
```

FME Flow Automations - Metrics

Constructing HTML tables to hold the review assignment metrics.

```
email_message_progress = f'<br><div style="margin-left: 15px; font-weight: bold;">{user_current_progress} - Current Progress. Miles marked as complete for the week of {datetime.now().strftime("%
                         '<table style="margin-left: 15px; border-collapse: collapse; border: 1px solid #aaa;">' + \
                         '<tr><td style="border: 1px solid #aaa; padding: 5px; font-weight: bold;">USER</td>' + \
                         '<td style="border: 1px solid #aaa; padding: 5px; font-weight: bold;">PREV_COMPLETED_MIL</td>' + \
                         '<td style="border: 1px solid #aaa; padding: 5px; font-weight: bold;">CURR_COMPLETED_MIL</td>' + \
                         '<td style="border: 1px solid #aaa; padding: 5px; font-weight: bold;">MPW</td>' + \
                         '<td style="border: 1px solid #aaa; padding: 5px; font-weight: bold;">CURR_MIL_READY_FOR_REVIEW</td></tr>'
```

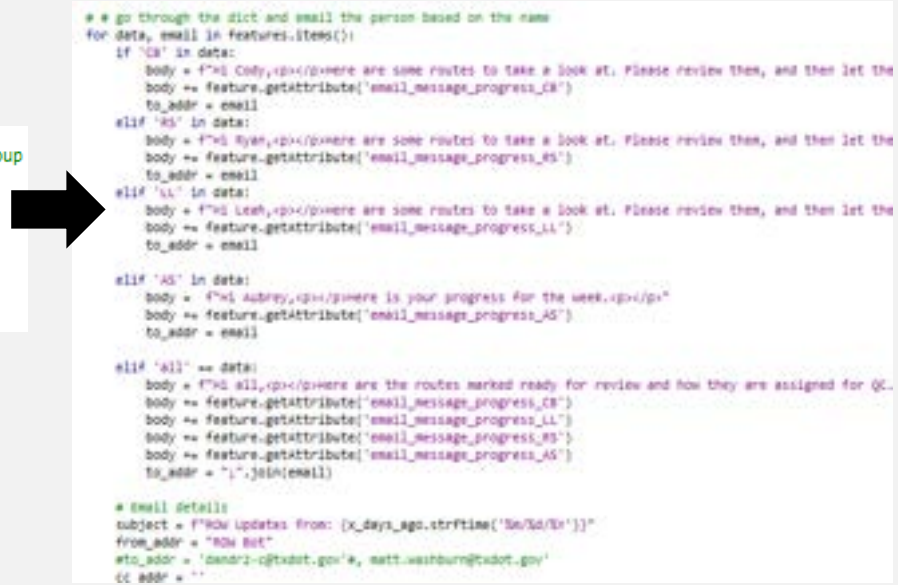
```
if i["PREV_MIL"] != '': # and (i["CURR_MIL"] != '' and i["USER"] != 'AS'):
    email_message_progress += f'<tr><td style="border: 1px solid #aaa; padding: 5px;">{i["USER"]}</td>' + \
                              f'<td style="border: 1px solid #aaa; padding: 5px;">{int(i["PREV_MIL"])}</td>' + \
                              f'<td style="border: 1px solid #aaa; padding: 5px;">{int(i["CURR_MIL"])}</td>' + \
                              f'<td style="border: 1px solid #aaa; padding: 5px;">{i["MPW"]}</td>' + \
                              f'<td style="border: 1px solid #aaa; padding: 5px;">{i["READY_FOR_REVIEW"]}</td></tr>'
```

FME Flow Automations - Metrics

A Python Caller was used to construct the emails.

- email_message* was built in previous caller
- slick way to use an object

```
# had to have a way to send emails to only the required reviewer, as well as a group
features = {'email_message_qc_CB' : 'wburt-c@txdot.gov',
            'email_message_qc_RS' : 'rschel-c@txdot.gov',
            'email_message_qc_LL': 'llobsi-c@txdot.gov',
            'email_message_qc_AS': 'aubrey.shuttles@txdot.gov',
            'all' : ['matt.washburn@txdot.gov', 'dandr2-c@txdot.gov']
            }
```



```
# # go through the dict and email the person based on the name
for data, email in features.items():
    if 'CB' in data:
        body = f"<i Cody,<p></p>Here are some routes to take a look at. Please review them, and then let the
        body += feature.getAttribute('email_message_progress_CB')
        to_addr = email
    elif 'RS' in data:
        body = f"<i Ryan,<p></p>Here are some routes to take a look at. Please review them, and then let the
        body += feature.getAttribute('email_message_progress_RS')
        to_addr = email
    elif 'LL' in data:
        body = f"<i Leah,<p></p>Here are some routes to take a look at. Please review them, and then let the
        body += feature.getAttribute('email_message_progress_LL')
        to_addr = email

    elif 'AS' in data:
        body =  f"<i Aubrey,<p></p>Here is your progress for the week.<p></p>"
        body += feature.getAttribute('email_message_progress_AS')
        to_addr = email

    elif 'all' == data:
        body = f"<i All,<p></p>Here are the routes marked ready for review and how they are assigned for QC.
        body += feature.getAttribute('email_message_progress_CB')
        body += feature.getAttribute('email_message_progress_LL')
        body += feature.getAttribute('email_message_progress_RS')
        body += feature.getAttribute('email_message_progress_AS')
        to_addr = ";".join(email)

    # Email details
    subject = f"ROW Updates from: {x_days_ago.strftime('%m/%d/%Y')}"
    from_addr = "ROW Bot"
    #to_addr = 'dandr2-c@txdot.gov'#, matt.washburn@txdot.gov'
    cc_addr = ''
```

FME Flow Automations – Logs Logs Logs

Show the edits that were made in the last 24 hours.


```
SELECT
    [OBJECTID]
   ,[RTE_NM]
   ,[HWY_CNTY]
   ,CONVERT(date, last_edited_date AT TIME ZONE 'UTC' AT TIME ZONE 'Central Standard Time') AS EDIT_DATE
   ,CONVERT(time, last_edited_date AT TIME ZONE 'UTC' AT TIME ZONE 'Central Standard Time') AS EDIT_TIME
   ,[ASSIGNED_TO]
   ,[TXDOT_DIST_NBR]
   ,[CNTRLN_MSL]
   ,[Shape]
   ,[STATUS]
   ,[GlobalID]
   ,[last_edited_user] as LAST_EDITED_USER
   ,[RTE_TYPE]
   ,[REVISOR]
   ,last_edited_date AT TIME ZONE 'UTC' AT TIME ZONE 'Central Standard Time' AS EDIT_DATETIME_CT
   ,last_edited_date AS EDIT_DATETIME_UTC
FROM
    [dbo].[TPP].[ROW_EDITS]
WHERE
    last_edited_date > DATEADD(HOUR, -24, SYSUTCDATETIME())
    AND last_edited_user <> 'dbo$SD_TPP'
ORDER BY
    EDIT_DATE DESC, EDIT_TIME DESC;
```

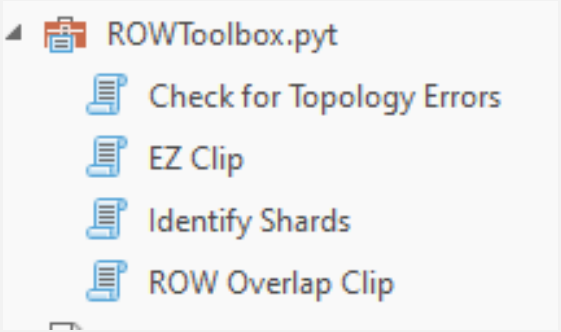
OBJECTID *	EDIT_OID	RTE_NM	HWY_CNTY	LAST_EDITED_USER	EDIT_DATE	EDIT_TIME
75284	6484	US0190	US0190-Polk	WBURT_C	10/14/2025	17:14:35
75285	6316	US0082	US0082-Bowie	RSCHEL_C	10/14/2025	16:58:44
75286	6316	US0082	US0082-Bowie	RSCHEL_C	10/14/2025	16:35:16
75287	6316	US0082	US0082-Bowie	RSCHEL_C	10/14/2025	16:28:39
75288	6316	US0082	US0082-Bowie	RSCHEL_C	10/14/2025	16:09:28
75289	6316	US0082	US0082-Bowie	RSCHEL_C	10/14/2025	15:40:40
75290	6316	US0082	US0082-Bowie	RSCHEL_C	10/14/2025	15:18:05
75291	6316	US0082	US0082-Bowie	RSCHEL_C	10/14/2025	15:07:54
75292	6316	US0082	US0082-Bowie	RSCHEL_C	10/14/2025	14:57:02
75293	6615	US0380	US0380-Young	ASHUTTLE	10/14/2025	14:48:47

Python

ROW Toolbox

As the project progressed, certain tools to use in ArcGIS Pro were requested. We ended up with four custom script tools.

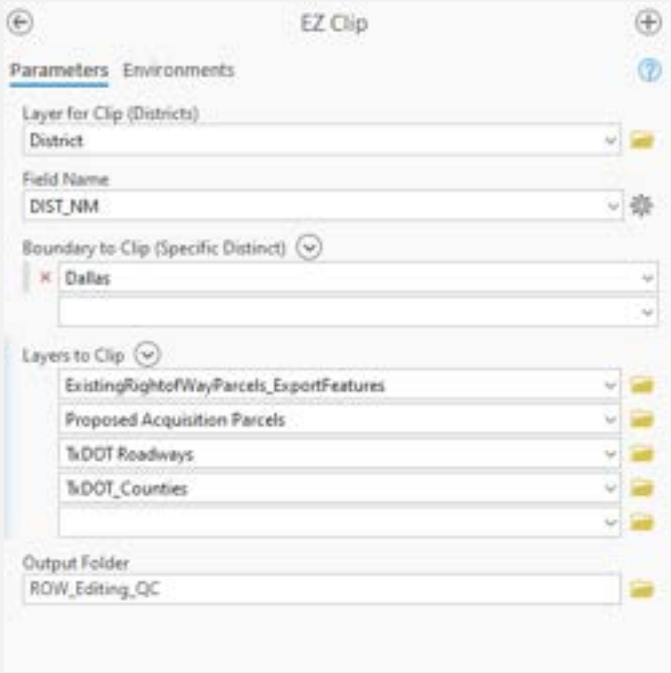
- Save time on clipping datasets for a new district
- Handle intersections and hierarchy assignment
- Save time on identifying topology errors



ROW Toolbox – EZ Clip (Setup Environment)

An early request in the project was a way to automate the clipping of the various layers (roads, counties, etc.)

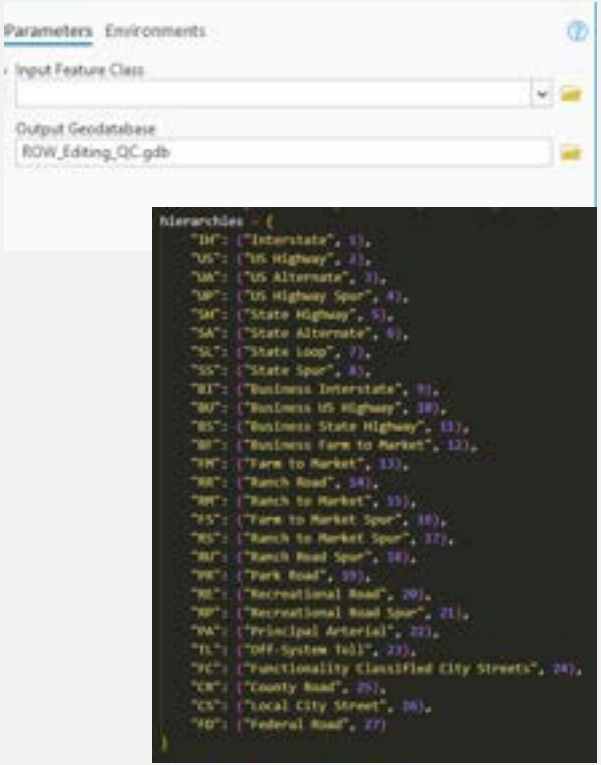
- Input is the TxDOT District layer
- Attributes of Field Name generate automatically
- Add vector / raster to be clipped



ROW Toolbox – ROW Overlap Clip

Later in the project, it became apparent that intersections took a while to cleanup.

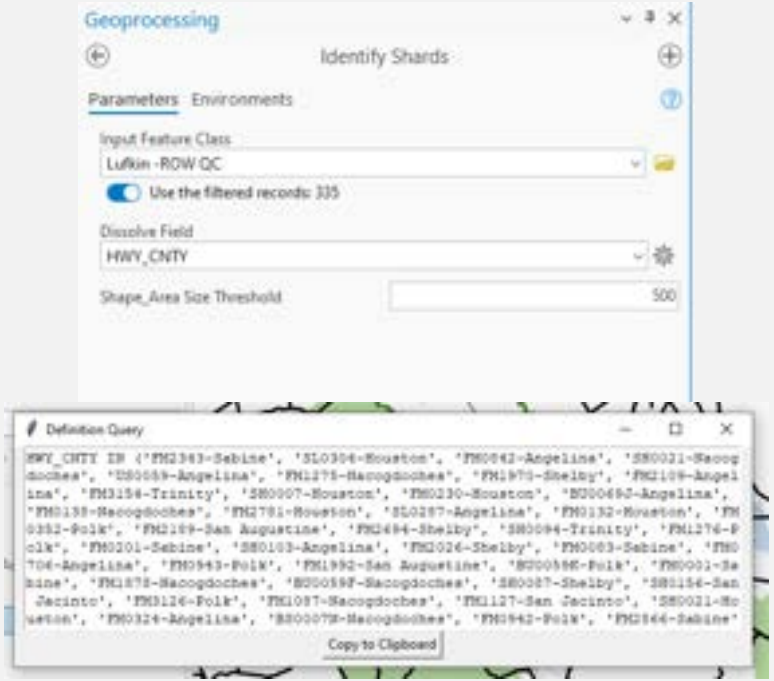
- The tool takes the hierarchies of routes and ensures the highest ranked clips the lower ranks
- Developed by our summer intern, Easton Moore!



ROW Toolbox – Identify Shards

An early tool used in the process to identify shards (slithers). The tool was quick to write and helped with QC.

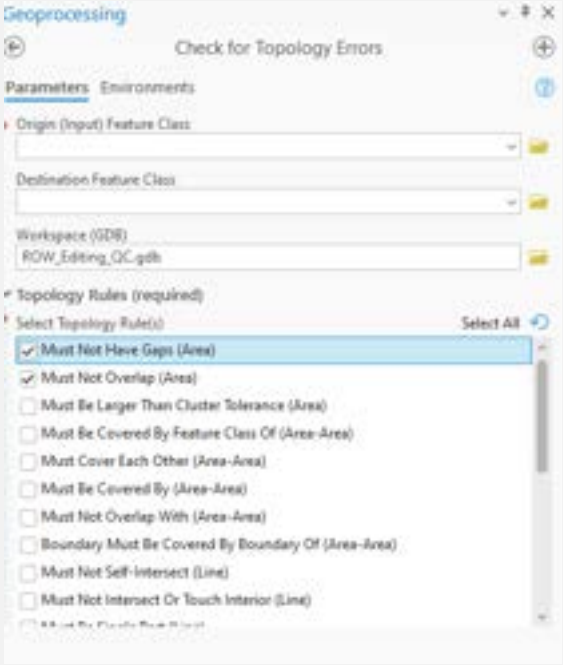
- Did a good job at finding topology issues, but was not perfect
- Provided a feature class with identified records and a def query to copy and run on the existing feature class
- Was replaced by the “Check for Topology Errors” tool



ROW Toolbox – Check for Topology Errors

Took existing topology workflow (adding feature class, creating the rules, validating, exporting) and condensed it into four inputs.

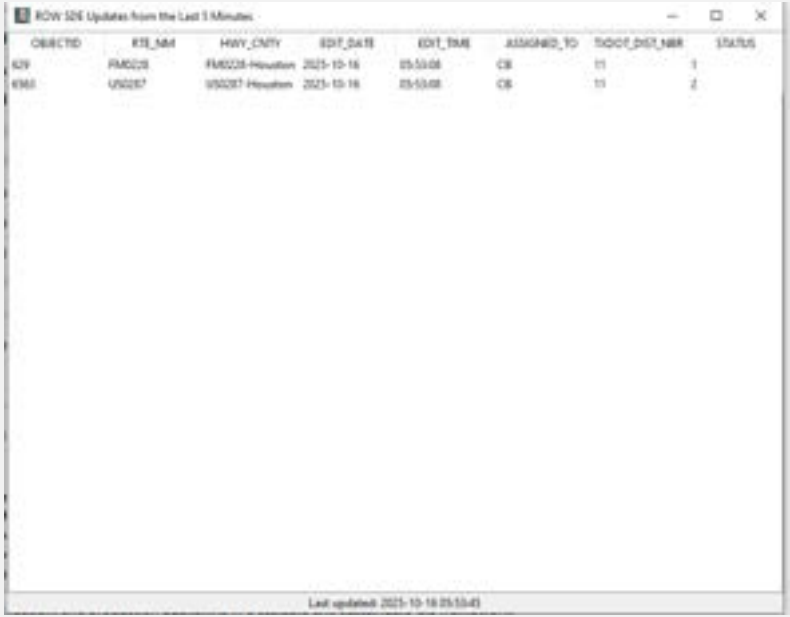
- Output is topology errors
- Saved time for the technicians



Realtime Edit Log Viewer

A compiled Python .exe that queries the database for edits every five minutes.

- Gave the technicians the peace of mind that edits were being saved
- Leveraged PyInstaller to compile, this allowed the app to be ran outside of Pro



Finishing Up / Lessons Learned

TxDOT ROW – hosted feature service

A feature service hosted on ArcGIS Online and shared with the organization is the final product for this effort.

- Built the service in the summer of 2025 and began adding districts as they are completed
- Along with the final product of this effort, the service includes TxDOT's existing and proposed ROW layers
- An FME workbench is currently under development to align geometry from future ROW acquisitions with the mapping grade product created in this effort

Lessons Learned

This effort spanned over many years, so we had some lessons learned

- Keep the crowdsourcing folks excited to participate
- Don’t be afraid to start with what’s familiar and then switch to what’s optimal
- Feature services published with custom datastores is better than working directly out of SDE
- Ask the folks doing the editing what would make their life easier, for real, bother them “what can I build you?”

A wonderful team effort.

Thank you to everyone who participated!

Questions?

Dave Andresen, GIS Developer

TxDOT, Transportation Planning and Programming Division
dandr2-c@txdot.gov