Maintaining 9-1-1 Data Integrity Using FME

March 09, 2022
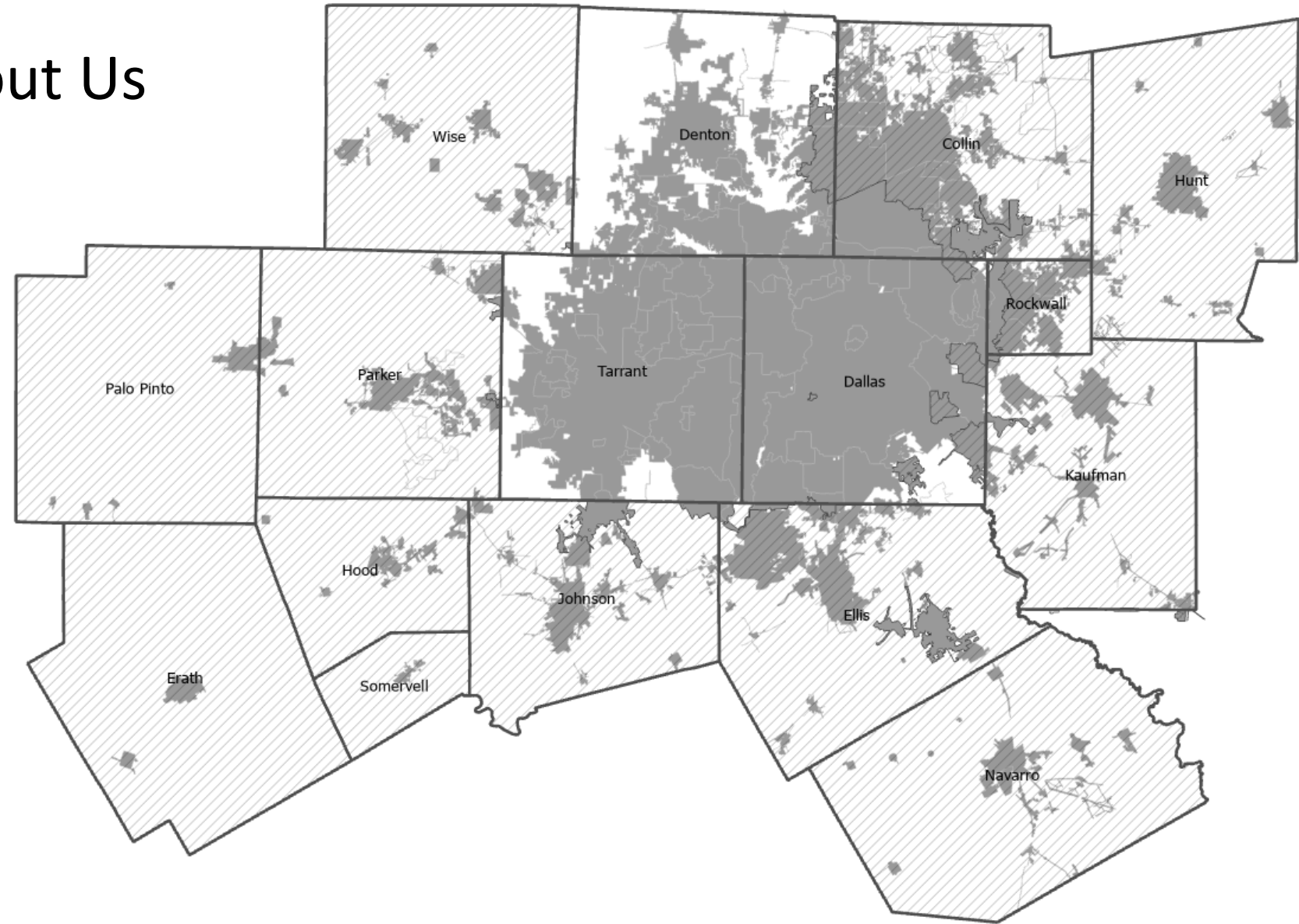
Texas GIS Forum
2022

Agenda:

- Introduction
- What are FME/ETL
- Why an ETL
- ETL @ NCT9-1-1
  - Field, Value, and Default Mapping
  - Conditional mapping
- Conditional Mapping
- Beyond an ETL
- Resources

About Us

- 13 counties in the DFW metroplex. Excluding Dallas*, Tarrant, and Denton
- 40 **E**mergency **C**ommunication **C**enters (ECC)
- ~2 million citizens served

*The cities of Sachse, Sunnyvale, Balch Springs, Seagoville, Combine, Wilmer, and Cockrell Hill are part of NCT9-1-1's service area as well
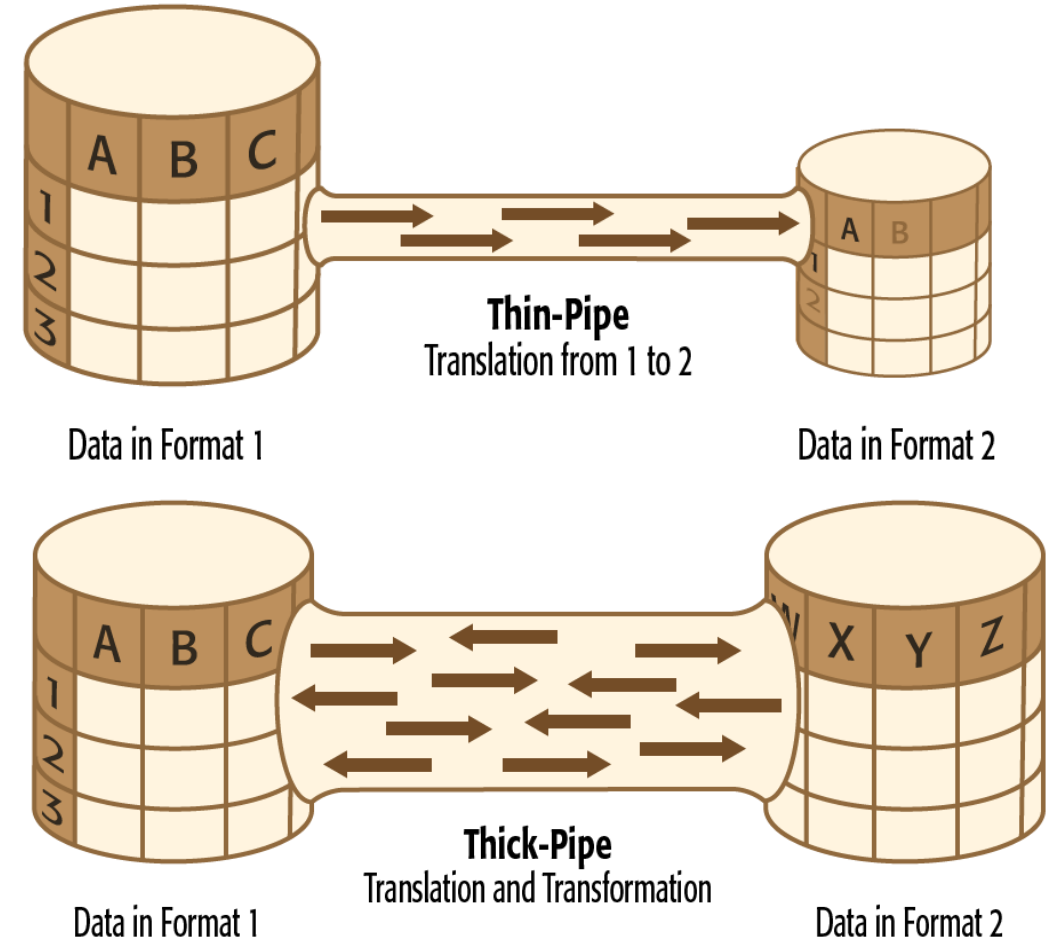
- Dozens of entities using database replication and 5 cities* submitting data via an ETL
- Biweekly updates to public safety systems
  - ECRF & LVF
  - 9-1-1 Call taking map
  - Locators
  - Supplemental apps
  - Download site
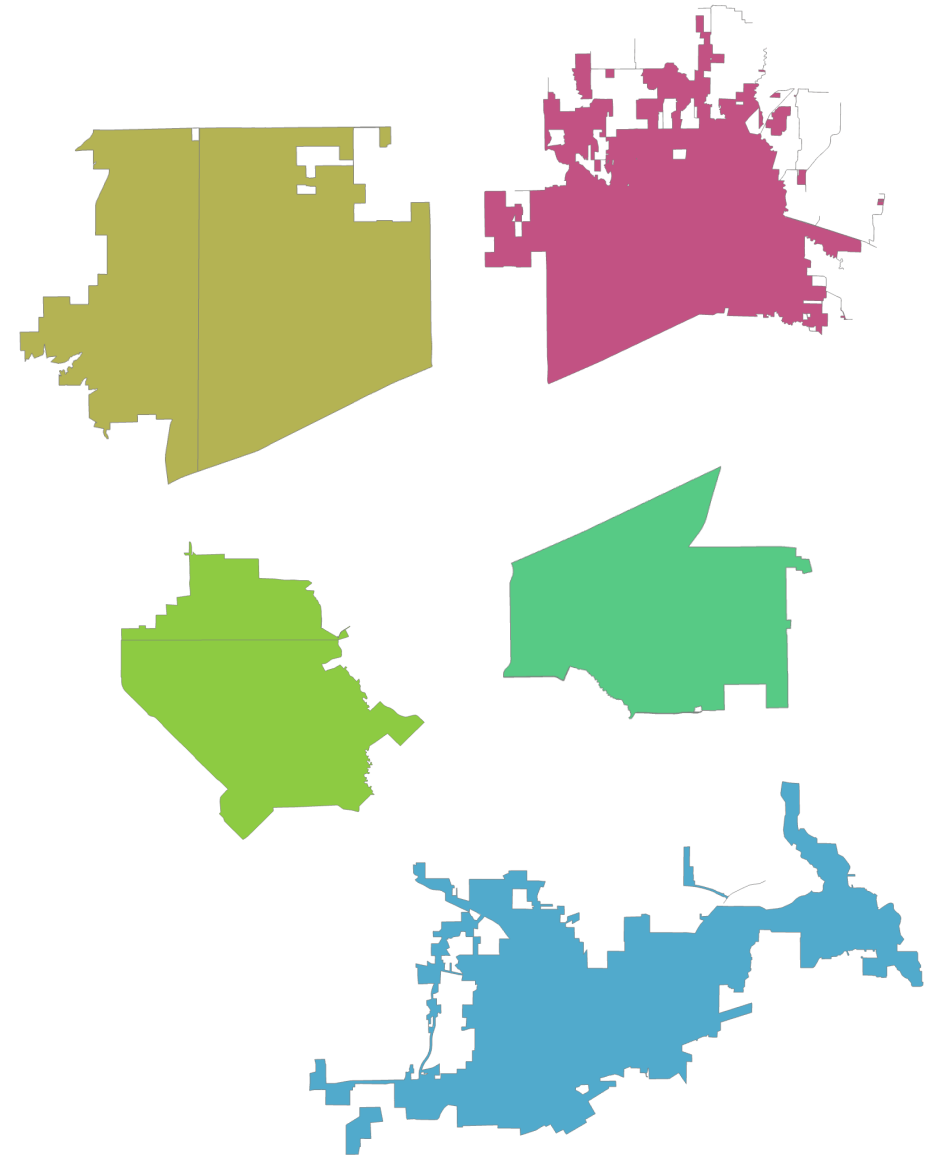
*The cities are Frisco, McKinney, Allen, Sachse, and Weatherford

What is FME

- **F**eature **M**anipulation **E**ngine. Enables data transformations via its rich data model which covers all possible geometry and attribute types.
- FME was the first tool designed to be a spatial ETL application. Today, FME's ETL capabilities cover many different kinds of data, both spatial and non-spatial.

Thin-Pipe
Translation from 1 to 2

Data in Format 1   Data in Format 2

Thick-Pipe
Translation and Transformation

Data in Format 1   Data in Format 2

Info here obtained from https://docs.safe.com/

What is ETL

Extract, **T**ransform, **L**oad. Process that extracts data from one data source/format, transforms it, and loads in a destination database and format.

EXTRACT   TRANSFORM   LOAD

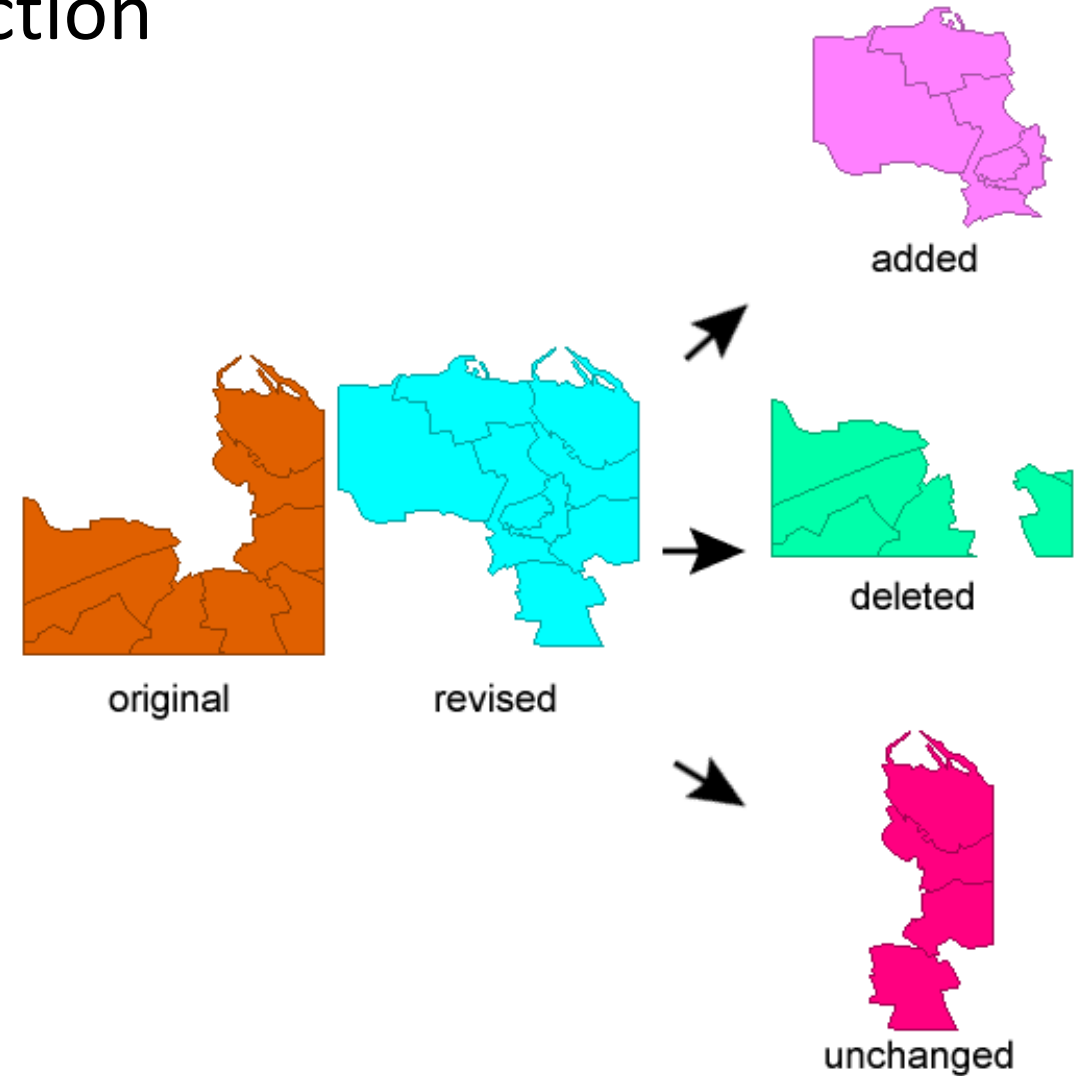Graphics from https://www.talend.com/resources/what-is-etl/
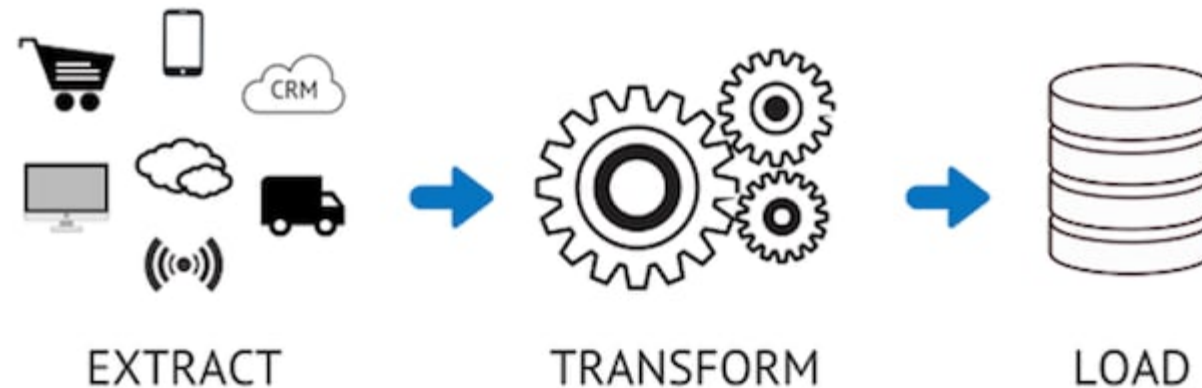
Why an ETL

- An ETL does not require a unified schema
- QAQC can be largely automated via on-the-fly fixes and reporting
- Automation potential from beginning to end
- Way to bypass budgetary/licensing restrictions on the client side
  - FME can read & transform virtually all spatial data types
- Change Detection

ETL PROCESS

SOURCE SYSTEM — EXTRACT, TRANSFORMS, LOAD — DATA WAREHOUSE

Change Detection

- Change detection eliminates the need for wipe & replaces
  - Simple to use & reliable
- Example: New data (revised) is compared to existing data in destination databases (original)
  - If present in revised but missing in original, *add* to original
  - If it no longer exists in revised, but is present in original, *delete* from original
  - If it exists in both, but is slightly different in each, delete original version and add revised version ("edits" or *delete* + *add*)
  - If identical, leave them unchanged

original   revised   added

deleted

unchanged

Image from http://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_Transformers/Transformers/changedetector.htm

Field, Value Mapping

Value Mapping

- Field mapping allows for *Field A* = *Field B* translations
- Value mapping allows for *Value 1 of Field A* = *Value 3 of Field B* translations
- Default mapping allows for *Field B* = *Uniform Value*

Field & Default Mapping

GC_Exception	GC_Exception
Source	ALLEN ETL *(Default Mapping)*
FromAddr_L	LeftFromAddress
ToAddr_L	LeftToAddress
FromAddr_R	RightFromAddress
ToAddr_R	RightToAddress

Attribute Selection

Source Attribute:	COA_Classification
Destination Attribute:	NCT_Type
Default Value:	Problem in NCT_Type transformer

Value Map

Mapping Direction:	Forward (Source To Destination)

Source Value	Destination Value
CONVENIENCE STORE	Shop
GARAGE_PUBLIC PARKING	Garage
2_3_4PLEX	Duplex
AMUSEMENT_COMM_IN	Other
ANTIQUE_SHOP	Shop
APT	Apartment
AUTO_PAINT_BODY_SHP	Shop
AUTO_PARTS_SALES	Shop
AUTO_REPAIR_MAJOR... (MultiLine)	Shop
AUTO_REPAIR_MAJOR... (MultiLine)	Shop
AUTO_REPAIR_MINOR	Shop

Conditional Mapping

- "Conditional mapping" refers to any relational "translation" from source to destination features that requires more than a simple 1 = a, Apt = Apartment, etc.
- Similar idea to if/else statements
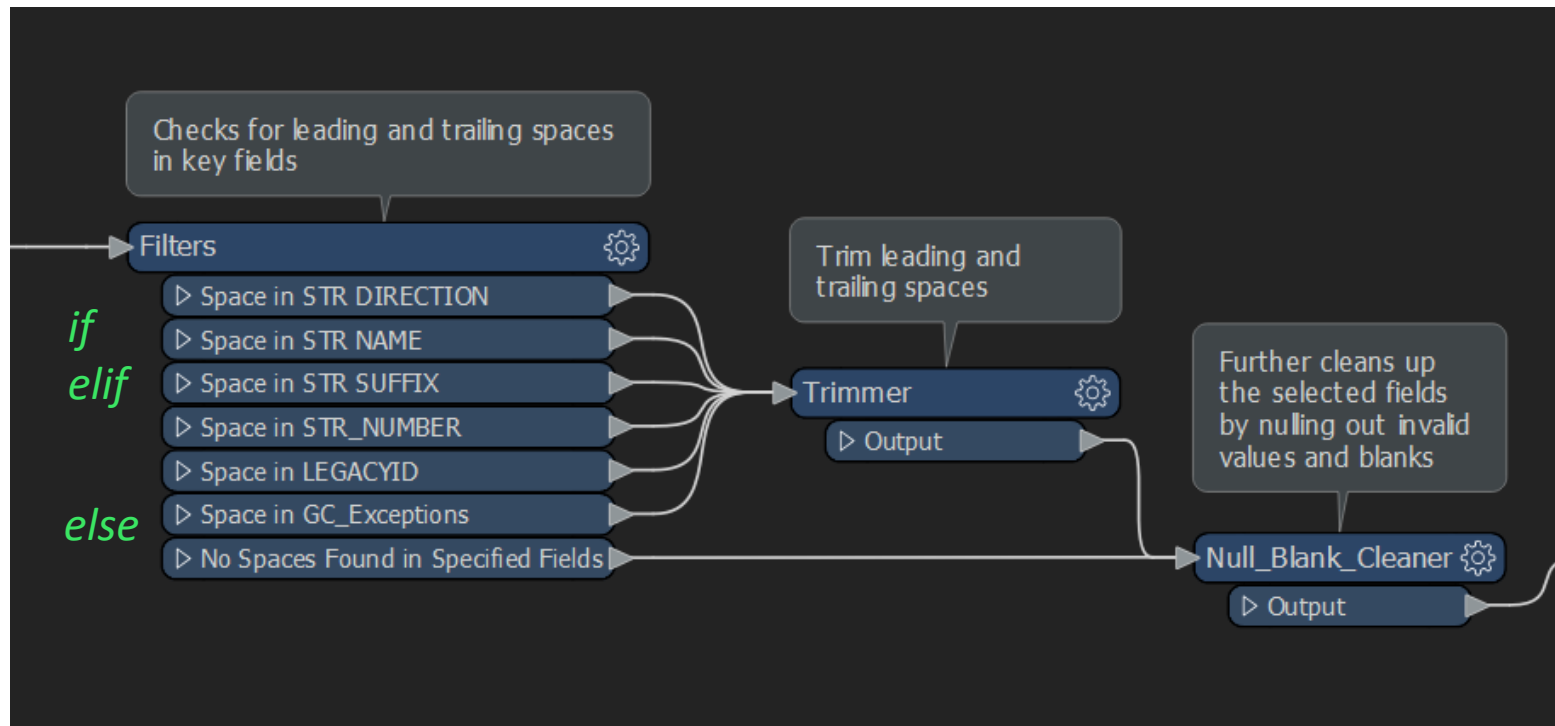
Conditional Mapping – Examples

Goal: Create and populate the field "Low Range" from the values of "LeftFromAddress" and "RightFromAddress." This field stores the lowest of all "from" ranges.

Calculate *LowRange*:

    If *LeftFromAddress* = 0, \<Null>

        Then, *Low Range* = *RightFromAddress*

    Elif *RightFromAddress* = 0, \<Null>

        Then, *LowRange* = *LeftFromAddress*

    Else, take the minimum of *LeftFromAddress* and *RightFromAddress*

	Test Condition	Attribute Value
If	LeftFromAddress ATTRIBUTE_VALUE_NULL OR LeftFromAddress ATTRIBUTE_IS_EMPTY OR LeftFromAddress ATTRIBUTE_IS_MISSING OR @Value(LeftFromAddress) = 0	⬅ RightFromAddress
Else If	RightFromAddress ATTRIBUTE_VALUE_NULL OR RightFromAddress ATTRIBUTE_IS_EMPTY OR RightFromAddress ATTRIBUTE_IS_MISSING OR @Value(RightFromAddress) = 0	⬅ LeftFromAddress
Else If		
Else	\<All Other Conditions>	@Evaluate(@min(@Value(LeftFromAddress),@Value(RightFromAddress)))

Conditional Mapping - Examples

Goal: Create and calculate the field "Parity_L" from the values of "sFromAddr_L" and "sToAddr_L." This field indicates whether the left ranges for a given road are **E**ven, **O**dd, **Z**ero, or **B**oth*.

Calculate *Parity_L*:

    If *sFromAddr_L* ENDS_WITH with 0,2,4,6,8

        AND *sFromAddr_L* ≠ 0**

        AND *sToAddr_L* ENDS_WITH with 0,2,4,6,8

            AND *sToAddr_L* ≠ 0**

      Then *Parity_L* = **E**ven

    Elif *sFromAddr_L* ENDS_WITH with 1,3,5,7,9

     sToAddr_L ENDS_WITH with 1,3,5,7,9

      Then *Parity_L* = **O**dd

    Elif sFromAddr_L = 0

      AND sToAddr_L = 0

      Then Parity_L = **Z**ero

    Else Parity = **B**oth

*These special scenarios are dealt with at a later step

** This accounts for numbers that end in 0 but do not equal 0. E.g.: 10, 20, etc.

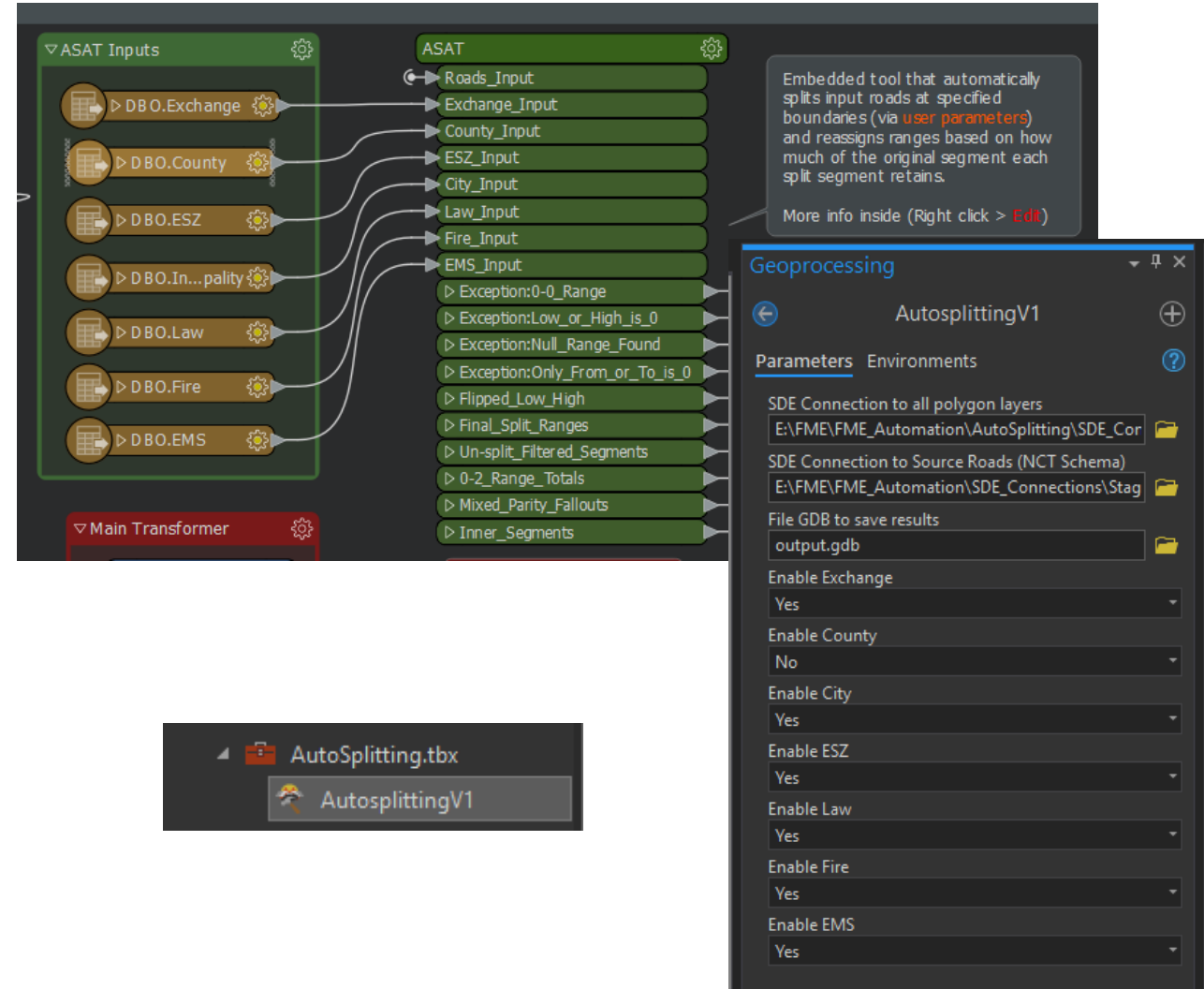| Test Condition	Attribute Value
If @Value(sFromAddr_L) ENDS_WITH 0	
@Value(sFromAddr_L) ENDS_WITH 2	
@Value(sFromAddr_L) ENDS_WITH 4	
@Value(sFromAddr_L) ENDS_WITH 6	
@Value(sFromAddr_L) ENDS_WITH 8	
@Value(sFromAddr_L) NOT_= 0	⬚ E
@Value(sToAddr_L) ENDS_WITH 0	
@Value(sToAddr_L) ENDS_WITH 2	
<4 more test(s)>...	
Composite Test: (1 OR 2 OR 3 OR 4 OR 5 AND 6) AND (7 OR 8 OR 9 OR 10 O...	
Else If @Value(sFromAddr_L) ENDS_WITH 1	
@Value(sFromAddr_L) ENDS_WITH 3	
@Value(sFromAddr_L) ENDS_WITH 5	
@Value(sFromAddr_L) ENDS_WITH 7	
@Value(sFromAddr_L) ENDS_WITH 9	⬚ O
@Value(sToAddr_L) ENDS_WITH 1	
@Value(sToAddr_L) ENDS_WITH 3	
@Value(sToAddr_L) ENDS_WITH 5	
<2 more test(s)>...	
Composite Test: (1 OR 2 OR 3 OR 4 OR 5) AND (6 OR 7 OR 8 OR 9 OR 10)	
Else If @Value(sFromAddr_L) = 0 AND @Value(sToAddr_L) = 0	⬚ Z
Else If	
Else <All Other Conditions>	⬚ B

Conditional Mapping – Examples

Goal: Split incoming road segments at specified boundaries (city limits, county boundary, etc.), assign the corresponding left & right values (County_L, County_R, etc.), and rearrange the road ranges based on how much of the original road each segment retained.
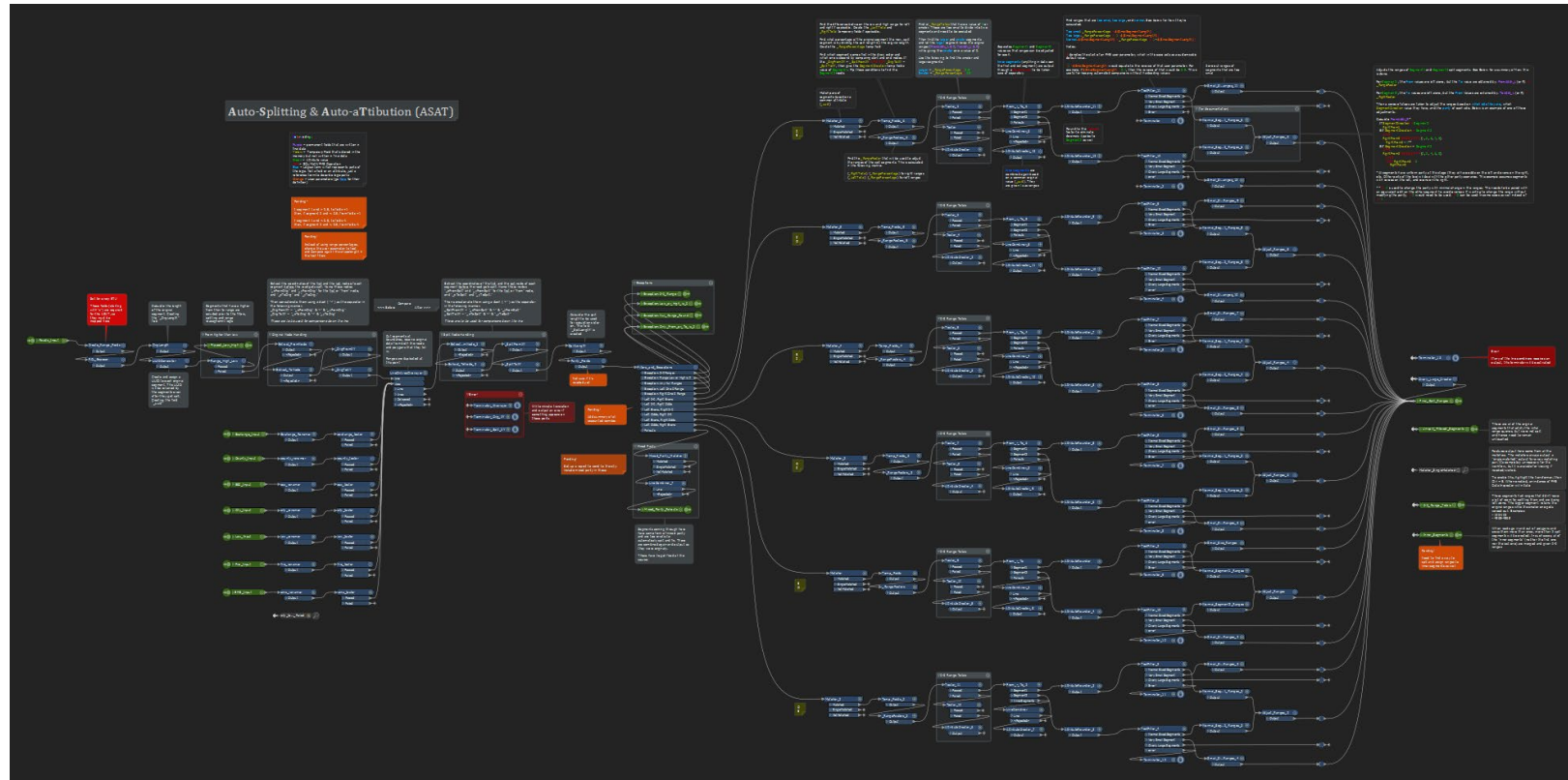
Beyond an ETL

- Custom tools can be created within an ETL or as a standalone tool. These tools can help with:
  - QAQC
  - Data quality
  - Reporting
  - General automation
- Can integrate existing scripts & automation
- FME server allows for further automation, scheduling, etc.
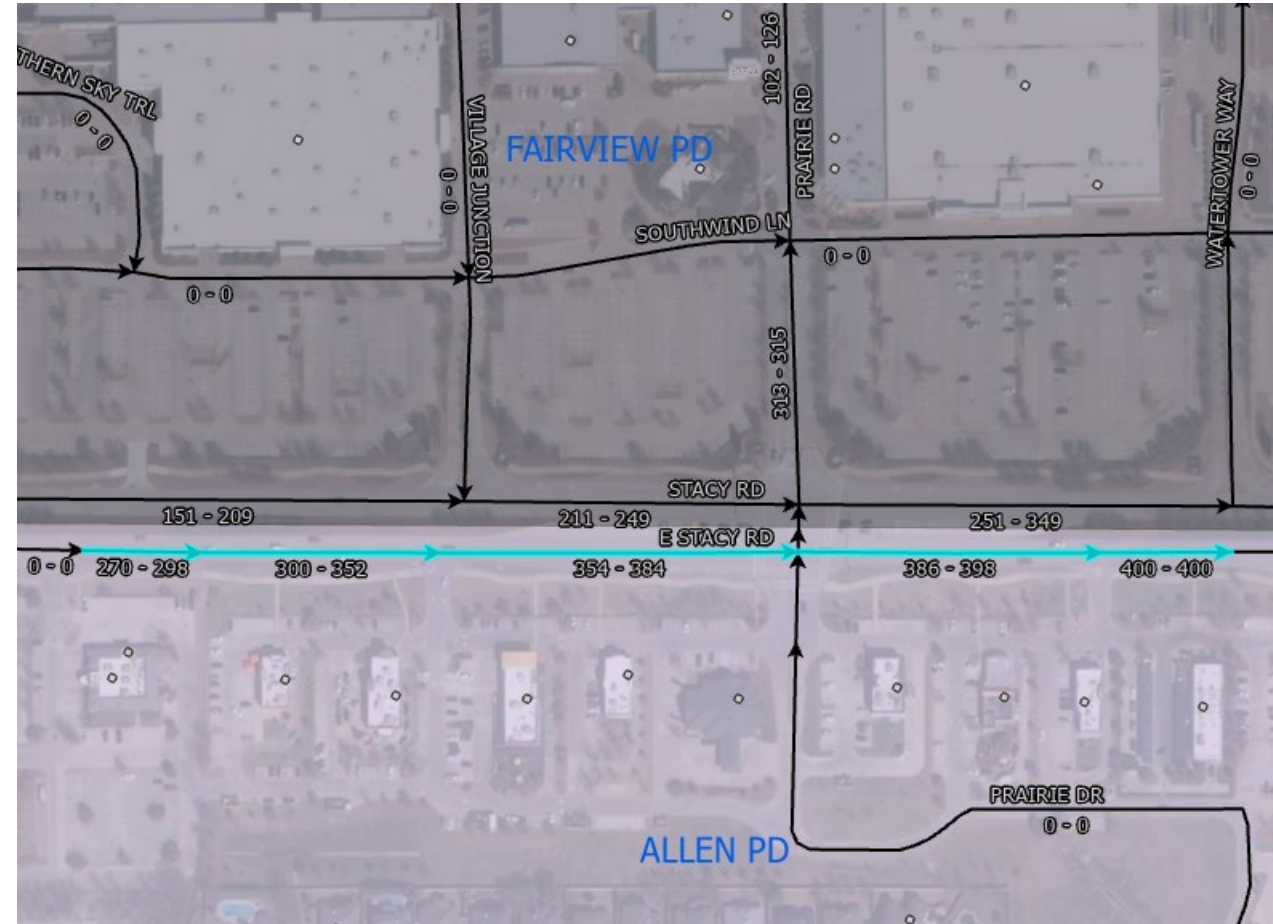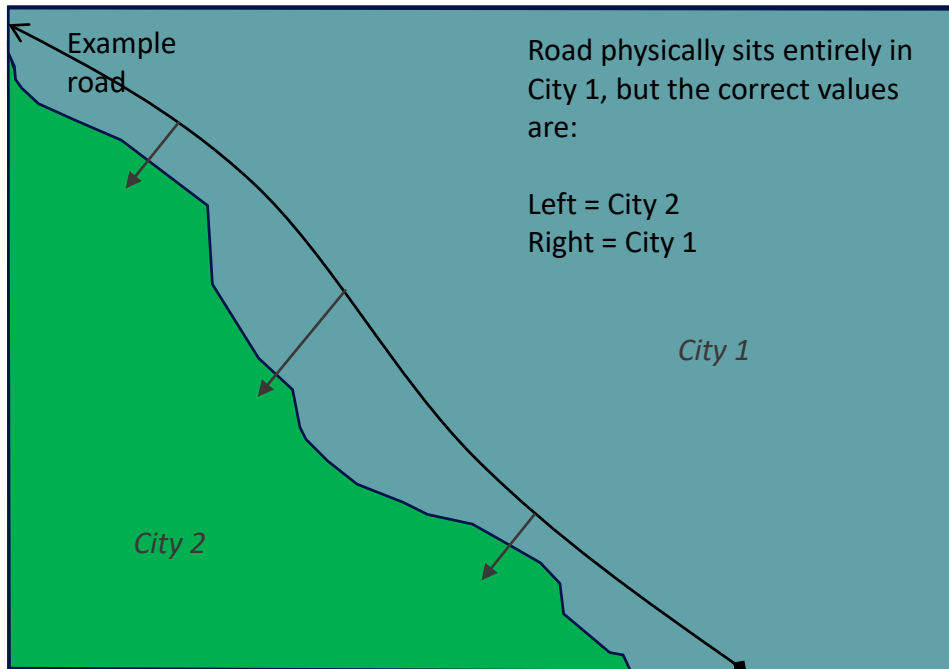
Beyond an ETL - Examples

- "Neighborhoods" tool
  - Tool that downloads latest subdivision layer from relevant appraisal districts, cleans up geometries & labels, and loads finalized product in destination databases
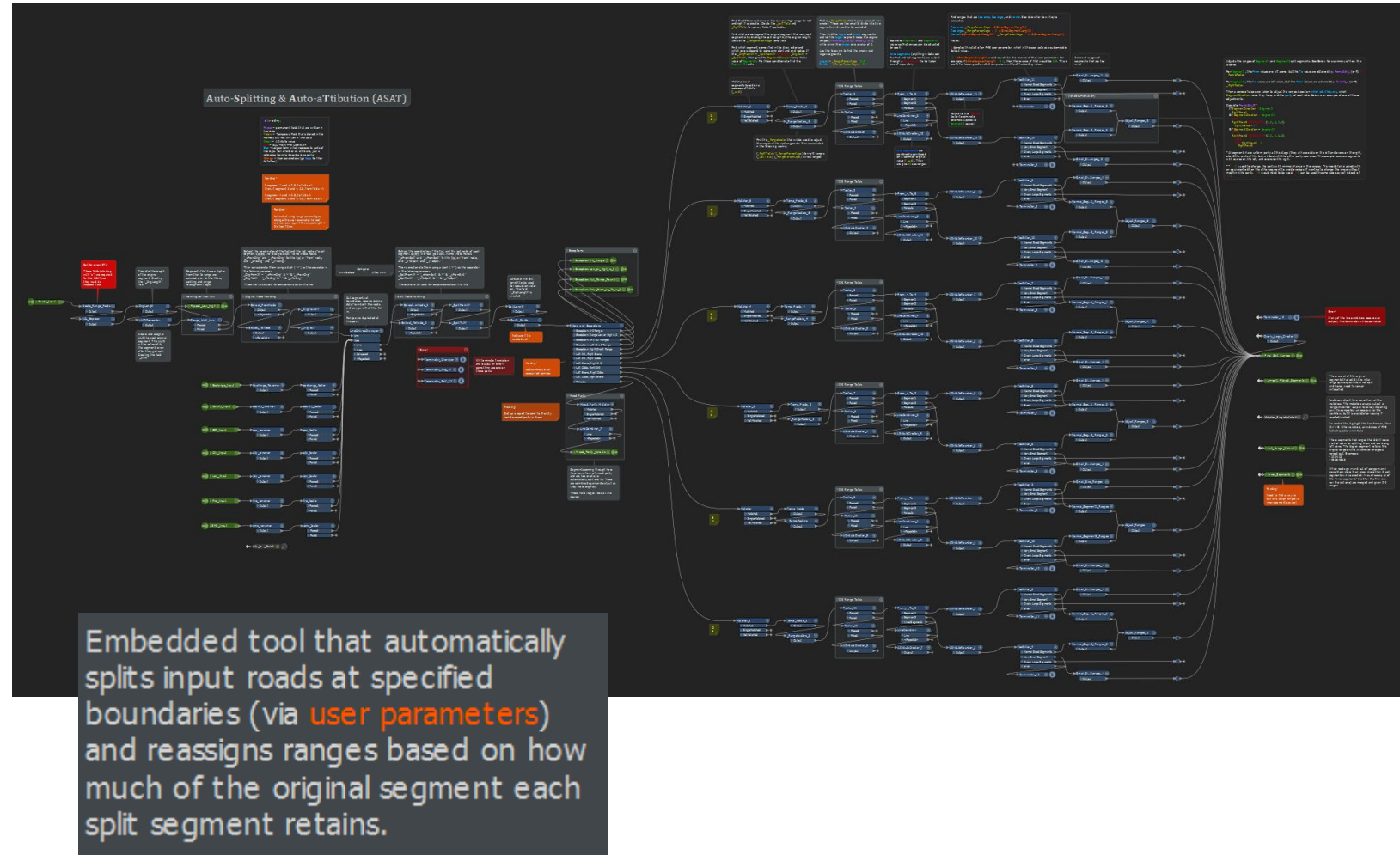

Before

After
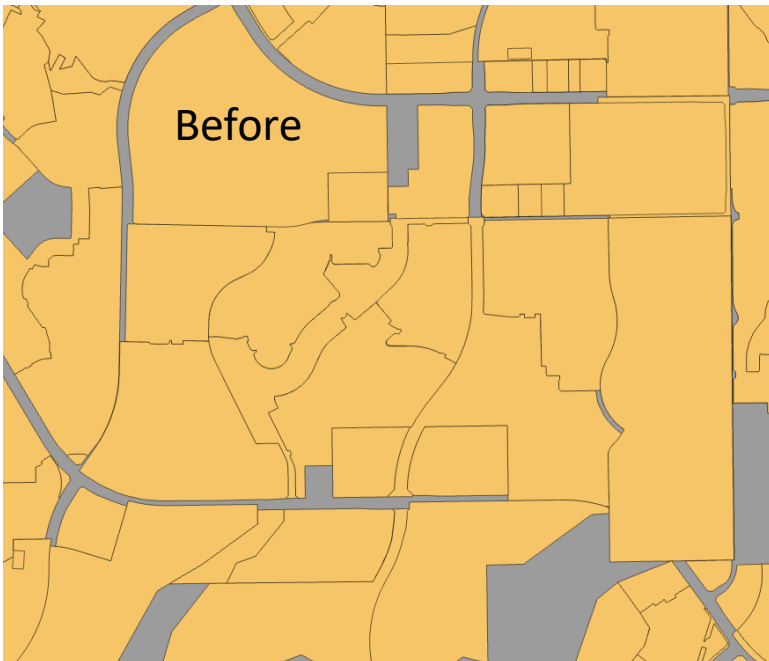
Beyond an ETL - Examples

- Road Centerline "Bufferer"
  - Tool that grabs attributes from nearby polygons and not where the road sits in

Example road

Road physically sits entirely in City 1, but the correct values are:

Left = City 2
Right = City 1

City 1

City 2

Beyond an ETL - Examples

- **A**uto-**S**plitting & **A**uto-A**t**tribution (ASAT):
  Tool that splits incoming road segments at specified boundaries (city limits, county boundary, etc.), assigns the corresponding left & right values (County_L, County_R, etc.), and rearranges the road ranges based on how much of the original road each segment retained.

Embedded tool that automatically splits input roads at specified boundaries (via user parameters) and reassigns ranges based on how much of the original segment each split segment retains.

Beyond an ETL - Examples

- Other custom FME tools used at NCT9-1-1:
  - Standalone change detection tool
  - Highway shield extractor
  - Raster formatter
    - Reprojects & converts from one format to another (depending on what the user chooses)
  - Road full name concatenator
  - Standalone geometry cleaner
  - Standalone NGUID concatenator
  - Label cleaner
    - Removing "phase 1," "tract 2," etc. from subdivision names

- Document your tools
- Give unique names to transformers
- Avoid hardcoded values
  - Use User Parameters as much as possible
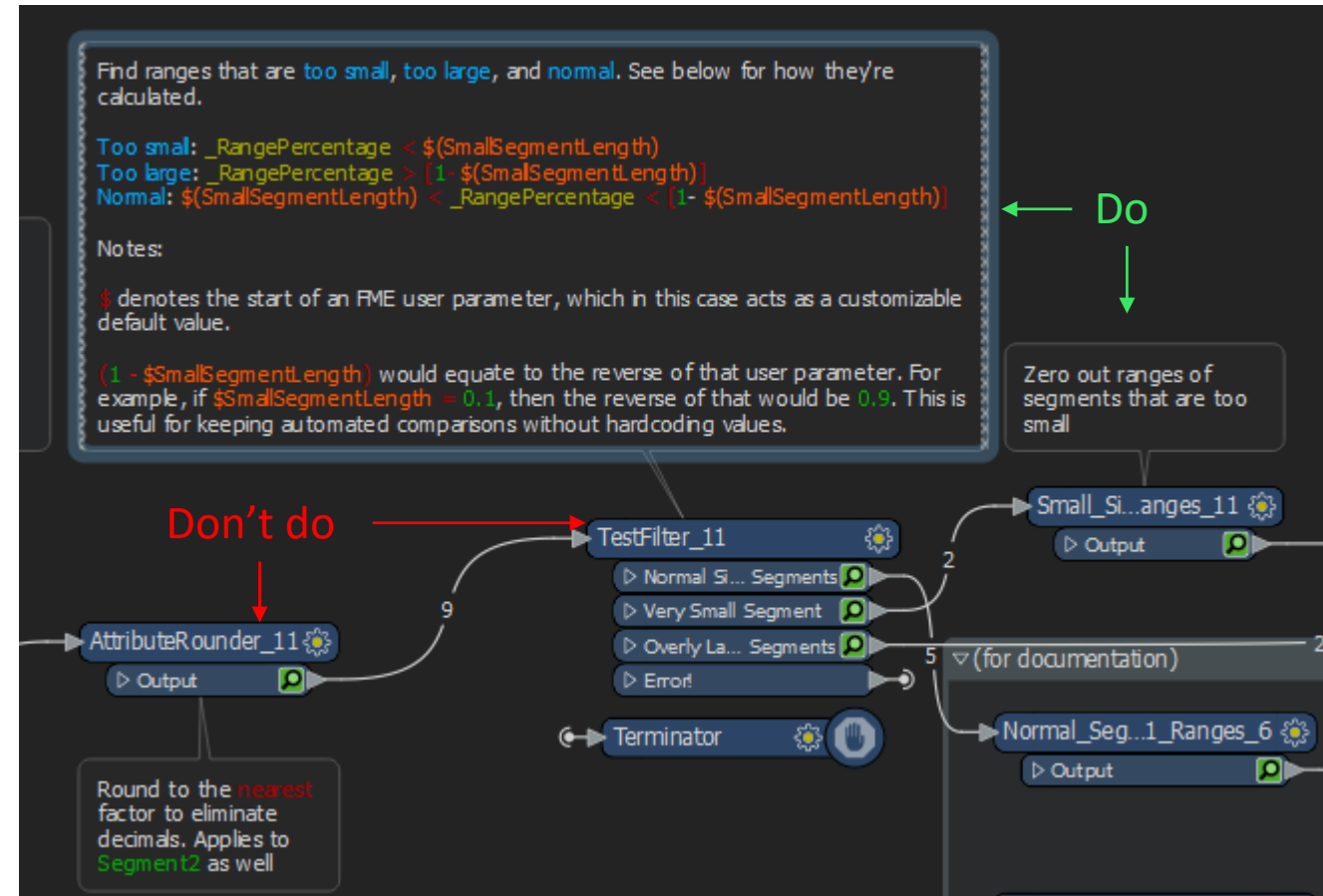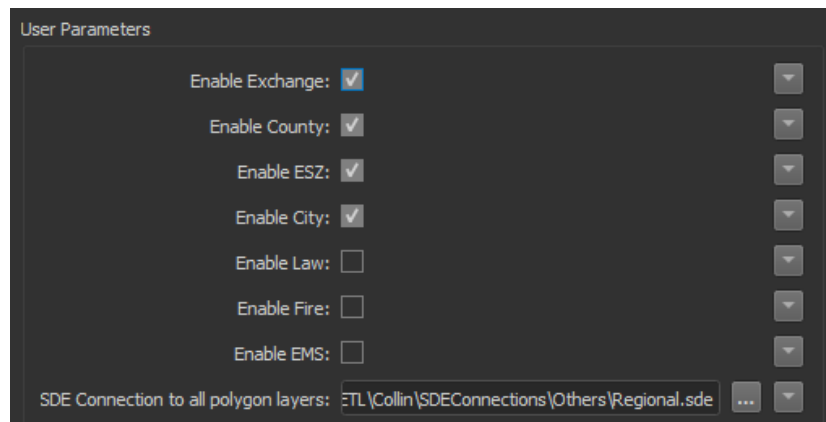
Resources

- Safe.com
  - Tutorials & Training
  - Community "Hub"
  - Official documentation
  - Links to FME partners
  - FME Hub
- Webinars & Conferences
  - Look for FME/ETL contents!

Questions or Comments?

Contact Info

Bruno Blanco

9-1-1 GIS Specialist

BBlanco@nct911.org or
911GISTeam@nct911.org

817-980-2586