

Streamlining Stormwater GIS Data Collection

June 2010

Author

Andrew Rudin, GIS Programmer Analyst, Watershed Protection Department, City of Austin

Abstract

The City of Austin (COA) is conducting a comprehensive drainage infrastructure GIS data collection project. Maintaining efficiency and quality control throughout the 5 year project has required the implementation of several special applications. ArcSDE provides a central location for data editing operations. Versioning allows multiple editors access to the data. Using custom editing tools, many default attributes are managed by the system, allowing editors to focus on the most important data. ESRI Job Tracking (JTX) manages the progress of each document through the research, entry, and QC phases. JTX is an essential tool for progress reports and other metrics. ESRI Data ReViewer provides quality assurance; regularly checking for topological anomalies and attribute errors. Data ReViewer also allows staff to flag errors or incomplete data in consultant deliverables and track corrections. Using these components together has eliminated several tedious repetitive tasks for staff, freeing time for better research and data.

Introduction

The purpose of this paper is to describe the tools, methods, and guidelines used by the City of Austin's Watershed Protection Department (WPD) during the creation of the Drainage Infrastructure GIS (DIG) for use by field crews and engineers for system maintenance, as well as by environmental staff for water quality analysis and response to environmental spills.

Project Background

The need for a stormwater GIS was first realized during the late 1980's. Attempts were made to create a GIS by digitizing pipes and inlets from an old series of paper stormsewer maps, known as Engineering Section Maps (ESM), which were maintained from roughly 1920 to 1970. Pipes and other features were digitized using ArcInfo Workstation but the pipes were not snapped and common attributes such as dimension and material were almost completely absent. The data was also not maintained after its creation. Nor was any other attempt made to find construction plans for pipes built from 1970 onward.

In the years after this, engineers and field crews that needed locations of stormdrains would use the GIS or ESM's as a first reference, then search the City's development databases for siteplan or subdivision plans for other resources. The process resulted in very low assurance of accuracy, and field crews always found conditions different in the ground compared to the research. Based on these issues, it was decided the City should invest in a new GIS system, this time involving collection of stormdrain features using GPS. In 2002 a pilot project was initialized. A GIS schema was developed by City engineers and a consultant was hired to collect features within the Waller Creek Watershed.

In 2006, a second pilot of conducted in the City's Central Business District(CBD), again by a consultant, this time involving a combination of GPS collection of surface visible features, and researching old City maps and plans from submitted site plans and subdivisions to attain the underground pipe network. Upon the completion of this project it was decided to continue on and create a comprehensive GIS for the entire City. During this pilot, our contractor found research of the City's 35,000 construction plans from their remote location difficult. There was no GIS to document the geographic locations of these plans, and there was no way to determine if a plan even contained information about the drainage system until it was visually inspected. As a result, it was decided the City could more cheaply and efficiently manage the research and digitizing of stormsewer features from construction plans using internal temporary GIS staff. This allowed us to better utilize resources at the City to filter and disseminate information from our development database. This also allowed our consultant to focus on field collection.

Team

Victoria Li: Department Director - Project Sponsor

Philip Campman: IT Director - Project Manager and Lead for Field Data Collection

Andrew Rudin: Programmer Analyst Senior - Internal Project GIS Team Leader

Daniel Davis - GIS Analyst - Lead for Quality Assurance and Data Standards

Yinghong Li: Programmer Analyst Senior - Develops custom GIS data entry tools

Temporary Staff:

7 GIS Technicians: Review plans and digitize information into GIS system

2 GIS Analysts: Conduct quality checking(QC) and quality assurance(QA) on work of GIS Technicians

Data Structure

The Department operates an Enterprise 9.3.1 SDE in an Oracle 10g database . It is housed on an IBM AIX server. There is also a separate SDE for testing purposes. Staff use Direct Connect when accessing data from SDE. Direct Connect provides faster retrieval of data and reduced demand on the server by using the client PC to manage GIS data requests to the Oracle database. The City's Corporate GIS Division also maintains a Citywide SDE for publishing data. When the project reaches milestones, WPD will publish the DIG data from our maintenance SDE to the publishing SDE for use throughout the City.

Data entry staff use ArcGIS Desktop 9.3.1 SP1. The DIG is edited in an ArcSDE versioned environment. A control version, owned by the lead GIS Analyst, exists as a child of the Default version. Editors create versions off the control version to make edits. ESRI's JTX application manages the creation and deletion of these versions, preventing user error. There is a separate control version for editing of data from our consultants conducting GPS collection.

DIG GIS Schema

The DIG GIS database schema was created by Philip Campman based on input from field crews, engineers, and environmental staff. Elements from the City's Water Utility's GIS, who had also recently created a GIS for their utility network, were also used. Philip designed the schema in Microsoft Visio using ESRI Computer Aided Software Engineering (CASE) tools. It uses abstract feature classes to store fields that are common across several feature classes. This makes changes to a field (ex. changing a text field's length from 5 characters to 10) much easier across the whole system.

DIG feature classes for the internal project are housed in a feature dataset and utilize a geometric network with complex edges. Originally the system was designed using simple edges in the network, but when work on our Computer Managed Maintenance System(CMMS) started, we decided to reduce the number of lines in our network where possible. As of May 2010, the culvert subtype on our drainage pipe feature class was allowed to be a complex edge. This way a culvert will appear as one feature to field crews, even if many stormdrain pipes flow into it.

Each feature class contains several auditing fields used to track who is editing features and when. All these fields are automatically populated by a custom ArcMap extension:

- DRAINAGE_ID - Unique ID for each feature across the entire drainage network. This field is populated by an Oracle sequence for new features.
- CREATED_BY - Stores login of user that first created the feature
- CREATED_DATE - Stores date the feature was created
- MODIFIED_BY - Stores login of user to last modify the feature
- MODIFIED_DATE - Stores date the feature was last modified.
- CHECKED_BY - Stores QA staff that verified attributes were correct.
- JTX_JOBID - Stores the JOB_ID from JTX for the job on which this feature was last edited.

There are also location fields that hold attributes for the most commonly requested 'select by location' searches by users:

- WATERSHED - Name of watershed the feature is in.
- PLACE_ID - ID of closest address to feature
- MAPSCO_PAGE - Name of the Citywide Map Grid the feature is in.

There are also fields that editors will populate to document the accuracy of the data we are entering. It is important to City staff to know whether the feature they are looking at was digitized from a tattered 1920's map of East Austin, plans from a street reconstruction project from 2008, or a GPS field visit.

- LOCATION_SOURCE - Documents the type of plan the geometry for the feature came from
- DIMENSIONS_SOURCE - Documents the type of source for the dimensions such as width, height, material, shape, etc.
- ELEVATION_SOURCE - Documents the source for elevation values like inverts or slopes. For linear features, there is an Up and Down Elevation Source field.

Each feature class also has a corresponding standalone table, also referred to as a REFDOC table. These tables are designed to hold reference information for each feature, such as the construction plan or map they were digitized from, links to field photos, or any other document which has information about the feature. Since a feature could have more than one related document, the tables are linked to the feature classes using an ArcGIS one to many composite relationship class.

Types of Source Documents

Based on research and interviews with City staff closely involved with the development process and our Public Works document archive specialists, the following key document types were singled out as the best resources for finding old plans for drainage construction.

Engineering Section Maps - Although details about the origins and maintenance of these maps is no longer exists. They were initially developed by City staff back around the 1920's. They took existing City plat maps and would draw stormdrain pipes, culverts, inlets and manholes using red ink. Only basic information about dimension was annotated. These maps were maintained until about 1970. At that time developers began submitting construction plans for subdivisions and site plans (construction on an individual parcel) to the City. Since the location and details about pipes and other features were documented on these submitted features, they decided the ESM's were no longer necessary and they stopped maintaining them.

Stormsewer Section Maps - A series of plan and profile maps which are meant to be a supplement to the ESM's. They provide slope and invert information for portions of the City's stormdrain network. Maintenance of these maps appears to have been discontinued at the same time as the ESM's.

Old City Drainage, Bridge, and Roadway As-builts - Since the 1800's, City staff have made several maps and construction plans documenting the location of existing or newly constructed stormdrains, ditches, or other drainage features. Scanned TIF images of these plans are all stored in the City's development review database, known as AMANDA.

Subdivision Plans - Usually construction of a new neighborhood after 1960 to present, these plans show existing and new drainage features. There are all stored in the AMANDA database.

Site Plans - Construction plans for work on an individual lot or parcel, these plans usually document privately maintained drainage on the property, but can also show improvements to City pipes due to construction.

After 40 years, the City of Austin has seen a lot of development and now our database of plans contains over 35,000 sets of plans, many of which overlap each other geographically.

Shared MXD

DIG staff use a master MXD file as their template for editing. It contains all the DIG features and reference layers symbolized and labeled for fast drawing and display of the most important information. Editors can then tweak the MXD to their personal preference after they copy it from a shared location. Since our Oracle licensing requires that every user of our SDE login to the database using their own credentials, a VBA script is run against our template MXD after changes to remove any logins and passwords from the layers. This will force ArcMap to request SDE credentials from users when the MXD opens.

Although JTX has an option to launch MXD's from the job's workflow and repoint the layers' version, we stopped using it at JTX 9.0 when ESRI removed the option to launch the MXD asynchronously (allowing interaction with JTX application while ArcMap is open) and save any modifications to it. Now that our workflow has changed to more shorter jobs, we are revisiting the option to use JTX to launch MXD's for editors, but they will lose the option to save any modifications they make to a job's MXD. The benefits of this option are now beginning to outweigh trusting users to manually switch versions between jobs.

Custom Tools

Our programmer has written custom ArcGIS Desktop tools for editing the DIG features. The tools are written in C#.NET. They are derived from VB6 tools Austin Water Utility (AWU) had written during their GIS implementation several years ago. After obtaining AWU's code, several modifications, enhancements, and brand new tools were created.

Automatic Population Tools - Automatically populate auditing attributes on features, and populate location fields when features are moved.

- **Sticky Attributes Tool** – Allows each user to store custom default values for fields. When new features are created, the custom values are automatically populated into fields.

Feature Editing Tools

- **Merge Lines** - Allows user to merge to line features together. The tool also merges all the REFDOC table records for each feature and intelligently handles copying of up and downstream inverts.
- **Split Lines with new junction**. Splits one linear feature into two. The tool allows any of the point features involved in the network to be placed as a junction where the split occurs. The tool also copies any existing reflow's from the old line to the two new ones.
- **Copy features between feature classes** - Allows user to move a feature from one feature class to another. Copies REFDOC's and alerts users of any schema issues when copying.
- **Elevation Editing Tools**
 - **Update Elevation and Invert Fields Tool** - Calculates intermediate points' elevation and update the corresponding line's invert fields.
 - **Update Invert Fields Tool** - Update line feature's invert fields by from and to points' elevation values.
 - **Calculate Length Tool** – Allows user to select several connected line features, then enter a total length given on a plan. The tool will then proportion the given length across all the segments.

Reference Document Editing Tools

- **Create new REFDOC for many features** - Allows user to select many features on many feature classes. Then the editor can use this tool to enter details about the reference document once and batch create a REFDOC record on all the selected features.
- **Copy REFDOC tools** - Allows user to copy existing REFDOC records from one feature to many others.

Entering ESM's and SSM's

When the internal project began, the first goal was to enter the ESM's into GIS. They are considered a complete and pre-digested rendering of the stormsewer network as it existed around

1970. This seemed like a good base layer to get into the system to create a city-wide up to date GIS. The maps were entered in groups of ten. Each editor was assigned a group of ten to enter in one version. That work was then QC'd by a more experienced GIS Analyst, then posted to the control version. After all the maps were entered, GIS technicians were assigned sections of town to peer review. This provided a chance to double check work and to fix any features that were entered using an older data entry standard.

It was at this time that our data entry standards document was developed. This document is used as our master resource for attribute and connectivity rules. It also contains rules about data entry procedures for when data gaps or anomalies exist in plans. It is a living document that is updated whenever new types of information are discovered on plans, whenever a new project begins, or whenever the DIG schema changes.

After the ~900 ESM's were digitized, work began on entering the accompanying elevation information, the SSM's. This was a very challenging portion of the project. We went into it not realizing the complexities and errors of the information we were entering. It involved training GIS technicians in slope and invert calculations. Also, many of these profile maps were from 1930-1950. They did not always include all the information, requiring GIS technicians to do math calculations, and also often contained obvious mathematical errors, such as the slope marked being radically different than if it is calculated using the marked up and downstream inverts and length. A lot of time was spent consulting with engineers about these issues and how they would enter numbers for their models. After a month long pilot, we decided to scrap the data entered and spend more time creating new tools to assist with slope/elevation calculations and better standards. Although the data contained some errors, it is still considered very valuable by our engineers. After some down time we decided to enter the information as is from the SSM's. Since we store the source of the elevation data on each feature, it was decided staff will notice these elevations are from SSM's and thus are prone to more error than modern plans.

Researching Construction Plans

Once ESM's were completed, we began assessing the scope of other plans we would have to enter to completely build out our system. We knew there were site plans and subdivisions, but we didn't know how many or if there were any other document types in the City's possession. After conferring with staff from the Public Works Department plus the Planning and Development Review Department, some initial plan types were identified. All of these plans existed in the City's AMANDA Database. Even old documents going back to the 1800's had been scanned into the system.

We were able to attain complete lists of plans from AMANDA for each of these plans, which included about 35,000 records. GIS temporary employees then began researching these plans looking for types that could be eliminated from our data entry project since they would not contain drainage. They were able to filter some site plans and subdivisions because they were preliminary legal descriptions, or old versions of plans that had a more current revision, or never got a permit which means they were never built. Through this research we were able to eliminate 10,000 plans. This left 25,000 plans for our GIS technicians to visually review.

These lists of plans were converted into standalone tables in SDE. GIS technicians would then get assigned a subset of records in the table, pull up the plan's scanned images and review each page, and identify if there was drainage. If there was they would mark a field on the table in SDE. They also grabbed the date of the plan, and digitized a polygon on a related polygon feature class for the job's location within the City. Fields were added to the table for technicians to document the plan's date, a boolean field to document if the plan shows drainage features, a boolean field to document if the plan is an as-built.

To speed up the process, staff was instructed to move on to the next the plan the moment they noticed drainage on a plan. This prevented wasting time reviewing all 100 pages of a plan if drainage appears on the 5th page.

We decided to review all plans before beginning to digitize stormdrain features from them. The first reason was so we could draw polygons for their locations within the City. By knowing their locations we could prevent versioning conflicts since we could assign editors non-overlapping work areas. The second reason was we wanted to enter plans in chronological order beginning with the oldest plan. A major concern for our engineers is the age of the infrastructure. By entering the plans in order, we can document pipes with the date of the plan they were digitized from. If other plans from a later date show the road was expanded and the drainage network has changed, we can abandon the old features, and enter new ones with the new construction date.

Field Crew GPS Collection

To ensure accurate and up to date locations for existing drainage infrastructure a field inventory using GPS handheld units was also seen as a necessary requirement for a complete GIS. A consultant, Parsons, was hired by the City to conduct the field collection. So far they have completed a first pass of the City, collecting over 65,000 surface features. All the completed field data is stored in an SDE on the consultant's side. They check that data out to a file geodatabase. Next the data is checked out for each GPS unit and the data is copied onto Trimble GPS units running ArcPad 8. Each GPS unit is assigned to a two person crew. One person will take measurements of the inlet or other feature using tape measure, ruler, or level. A second person will enter the information into the Trimble unit as they are called off and also take a photo of the feature. The camera is a Bluetooth enabled Ricoh 500SE. When the crew member takes the photo, a thumbnail of the photo sent from the camera to the GPS unit and the file path is entered as a value in the attribute table for the feature. Custom forms were written by Parsons for ArcPad to expedite the data entry process. For example, if a curb inlet does not have a manhole, the other fields describing the manhole dimensions are automatically populated and grayed out.

Once entry for the day is complete, crews return to the office and the data from each unit is backed up and then checked back into the main file geodatabase. Photos are downloaded from the digital cameras. The next day every Trimble unit is updated with data from the main file geodatabase, ensuring no crew recollects features from another crew. QC and QA are conducted in the file geodatabase using various VB and VBA scripts. These scrips also move the photolink from the feature class attribute table into the REFDOC table as a new record. When ready to be delivered to the City, new features are sent via a file geodatabase to the City along with the field photos. The City then runs custom scripts on the new features to populate them with unique drainage ID's from the City's Oracle sequence. The link for the photo in the REFDOC table is updated with the new path to the photo on the City's shared drives. Next, the City will load the new features and REFDOC records into SDE. A two-way disconnected replica is used to send the field points and their new drainage ID's back to Parsons. Parsons will now have features with accurate Drainage ID's. When Parsons makes updates to features they have already collected and submitted to the City, they will make updates in the field or office, and then make the updates to their SDE and send back via the replica.

The field data uses separate feature classes than the historic plan research project to store data. The City feature classes were used as a template, then fields were added or removed as needed to accommodate GPS work. Some examples are fields to hold PDOP values or for holding temporary field measurements until scripts back at the office can be run to summarize them.

Features collected via field inspections are treated as a completely separate product from features collected from historic construction plans. This allows crews to work efficiently, collecting new features and not spending time searching for City-digitized features and then comparing attributes. Collecting the data independently from two different sources also provides a way to compare each source and ensure everything was collected. Eventually, both datasets will be merged together, combining the accuracy and up to date information from GPS for surface features with underground features and historical information from construction plans. The City DIG data entry staff use existing field points for reference. If the correlation is obvious City staff will snap features from plans to features collected using GPS. This provides more accurate pipe

alignments and reduces the amount of work required when the field and historical data are merged.

Entering Data from Construction Plans

It took nearly 9 months for temporary staff to review the 25,000 plans. When the project was complete, 15,000 were identified containing drainage infrastructure information. 7,000 of those were site plans, which we decided to leave to enter last. The time required to enter all these plans in the project deadline was daunting, and site plans will most likely only show privately owned drainage. So we wanted to focus on the City maintained pipes first.

The big challenge for construction plan entry was job management. When we viewed the polygon layer for construction plan locations there was severe overlap, there was no way to place every plan into a region. Some plans would always be on a region boundary. Instead of forcing plans into regions, digitizing them, and then edgematching, plans are now being entered one by one. Our programmer wrote a custom tool for JTX. It allows the user to provide an input polygon. The tool then searches for any plans that are in that polygon which do not have a JTX job. Then it sorts them chronologically. It takes the oldest plan and then searches for other plans that overlap it. If there is a plan with an older date that overlaps it, the plan is skipped, and the tool goes on to the next plan in the search polygon. If the plan does not have any overlapping plans with older dates, the tool creates a JTX job and associates the JTX job with the construction plan by placing the JTX JOB_ID in the construction plan table.

Using JTX

The DIG project has used the ESRI Production Line Toolset (PLTS) Job Tracking Extension (JTX) product since the beginning of the project. JTX is an application that provides workflow management in the GIS environment. Each Job was a workflow. That workflow is comprised of steps. Steps can be 'Edit Data', 'QC Data', 'Post Edits', 'Fill out paperwork', etc. Each of these steps can be assigned to an individual user or a group of users.

For the DIG project, each construction plan gets its own job. The DIG staff use JTX to track where jobs are in the workflow and assists them with finding jobs that are ready for review after entry using queues. They also use it to populate details about each construction plan as they review them. JTX also helps with project management by showing how many plans have been entered, how long they took, how many plans are currently being entered, and how many are left to do.

JTX also provides version management. When a job is created, an editor will grab the job from a queue and assign it to themselves. Then they run a custom step that creates an SDE version. Based on preconfigured rules for these kind of jobs, JTX knows which control version to create the new version from, which SDE user to use to create the version, and then gives the version a name including the job's job ID, a unique identifier within JTX for each job.

Using Python Scripts for QC

Many of our 24 feature classes have 20+ attributes each that editors must populate. This can be an overwhelming task prone to a lot of user error. To check versions for missing values before sending them to peer review editors run a custom python script, exposed through an ArcToolbox tool, that will execute a series of SQL statements against features the editor has selected. These queries will look for blatant errors, such as a required field being null. They will also do more intelligent checks such as comparing pipe up and down inverts, pipe length, and slope to look for mathematical discrepancies, or look for features that have elevation values but the source of the elevation data was not populated. Once the script is complete a report of errors along with the feature's ID is returned the editor. Peer review and QC staff will also run these checks to ensure the original editor entered all information completely.

The SQL queries are held in a standalone table in SDE and are easily editable by QA staff. To make the script capable of handling the many versions an editor could have in one day, the

python script uses layers as input, so when editors launch the ArcToolbox tool for the script, they can drag their layers in and the script will check the data in their version.

```

Completed
Close the dialog when completed successfully.

Running check YEAR_BUILT and YEAR_ABANDONED are null, check YEAR_SOURCE or update year_built/abandoned
***** Writing errors to Report. *****
Logging HEADLINE_ID 141131
Running check YEAR_BUILT is not null, check YEAR_SOURCE or remove YEAR_BUILT.
No errors found
Running check YEAR_BUILT is outside of the range
No errors found
Running check YEAR_SOURCE cannot be null
No errors found
Running check An elevation is outside of range
No errors found
Running check An elevation source is null or ERM, check elev_source
***** Writing errors to Report. *****
Logging HEADLINE_ID 141131
Running check An invert field has a value, check corresponding ELEV_SOURCE or update invert.
No errors found
Running check Calculate length
No errors found
Running check Calculate slope
No errors found
Running check Calculate the null invert.
No errors found
Running check DATE_BUILT is outside of range
No errors found
Running check DRAIN_SOURCE cannot be null
No errors found

##### Check Complete #####
Found 14 errors in 3 checked features
##### Summary #####
HEADLINE_ID    LAYER NAME    DESCRIPTION SOURCE OBJECTID    SEVERITY    BATCH
141148    Header    CHECKED BY cannot be null    8104    I    M
141148    Header    Elevation not null, check ELEV_SOURCE or update elevations    8104    I    M
141149    Header    YEAR_SOURCE cannot be null    8104    I    M
141150    Curb Inlet    CHECKED BY cannot be null    84401    I    M
141150    Curb Inlet    Elevation Fields are null, check DRAIN_SOURCE or update drainations    84401    I    Y
141150    Curb Inlet    Elevation not null, check ELEV_SOURCE or update elevations    84401    I    M
141150    Curb Inlet    ROTATION cannot be null or 140    84401    I    M
141150    Curb Inlet    YEAR_BUILT and YEAR_ABANDONED are null, check YEAR_SOURCE or update year_built/abandoned    84401    I    M
141150    Curb Inlet    MODIFIED DATE cannot be null    84401    I    M
141151    Swinlege Pipe    CHECKED BY cannot be null    27121    I    M
141151    Swinlege Pipe    MODIFIED DATE cannot be null    27121    I    M
141151    Swinlege Pipe    WIDTH and HEIGHT cannot be null, check dimensions    27121    I    M
141151    Swinlege Pipe    YEAR_BUILT and YEAR_ABANDONED are null, check YEAR_SOURCE or update year_built/abandoned    27121    I    M
141151    Swinlege Pipe    An elevation source is null or ERM, check elev_source    27121    I    M

Completed script QCDataCheckTools...
Executed (QCDataCheckTools) successfully.
End Time: Tue Jun 01 11:01:17 2009 (Elapsed Time: 45.00 seconds)
  
```

Example error report from Python attribute check

Using Data Reviewer

The newest tool used being used on the DIG project is the PLTS Extension Data Reviewer. This extension is a more advanced version of the Python script currently being run. Data Reviewer is a tool for GIS data quality assurance. It allows editors to run a variety of attribute and spatial checks on data and identify problems that need correction. Some simple examples are checking if all values in a field are in the domain assigned to that field, or checking if duplicate features exist on top of each other. Another benefit of Data Reviewer is the errors returned are stored in a geodatabase table. They can then easily be sorted by error, and revisited, and marked as corrected using ArcMap tools.

The original goal was for editors to run a series of Data Reviewer checks on each of their versions before sending it off for QC. This proved troublesome with previous versions of Data Reviewer, since it was incompatible with our SDE environment of many editors connecting with different logins, and the frequent switching of versions.

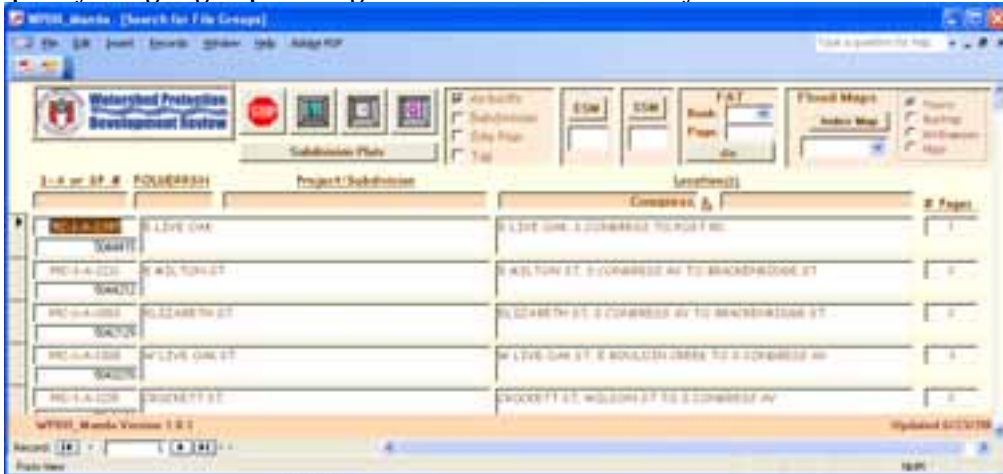
Until these problems are resolved, Data Reviewer checks will be used for weekly QA of the SDE Default version of our data. At the time this paper is being written, Data Reviewer checks are being written to run against the Default version to look for the following issues:

- Invalid geometries
- Multipart Lines
- Lines with true curves
- Lines that loop on themselves

- Lines with slivers or cutbacks
- Extremely long or short lines
- Features drawn outside the project area
- Attribute domain validation
- Subtype validation
- Point feature at same location as another point feature
- Duplicate vertices within same feature
- In-service pipes that cross other in-service pipes without a junction feature
- Orphan junctions
- Duplicate Unique ID's across entire network
- Every feature has at least one related record in its REFDOC table.

Quick Viewing of Plans

Visually inspecting hundreds of pages in a modern subdivision plan on screen can be very time consuming. To speed up the process, WPD obtained and modified a Microsoft Access Database from the City's Public Works Department, now known as WPDR_MANDA. That database acts as a front end for searching the construction plans in the Oracle Development Review Database, AMANDA. Once the desired plan is found, the user can copy all the plan's TIF images to their C drive for quick loading. Then they can use a 3rd party image viewer to scroll through the pages quickly using Page Up and Page Down buttons on their keyboard.



Example query of construction plans on Congress Ave

Conclusion

All of the methods and tools described above used together have alleviated many manual tasks. Editors can focus more on disseminating information from plans, and less on repetitive tasks like populating the date an edit occurred. Using one common MXD across all editors helps the team edit and comprehend the GIS in a similar way. JTX allows thousands of plans to be individually tracked through the editing process and sorted in the proper order for entry. Python scripts and PLTS Data Reviewer ensure data integrity. SDE provides one common location for publishing data and edits, allowing multiple users to edit the data simultaneously. One problem with using so many tools is they must all be kept up to date. If one changes, it can affect the others. Documenting geodatabase designs and tracking changes to the schema as they happen provide the best information ensuring other applications will not break. Having a testing environment that is as close to the implementation environment is also crucial with so many customized and interdependent components.

Acknowledgements

I would like to thank the following people for their contributions to this project.

Philip Campman - City of Austin
Mike Ihnat - City of Austin
Victoria Li - City of Austin
Ann McCracken - City of Austin
Yinghong Li - City of Austin
Daniel Davis - City of Austin
David Rodriguez - City of Austin
Jason Baker - City of Austin
Sarah Burzynski - City of Austin
Henna Chou - City of Austin
Roberto Davila - City of Austin

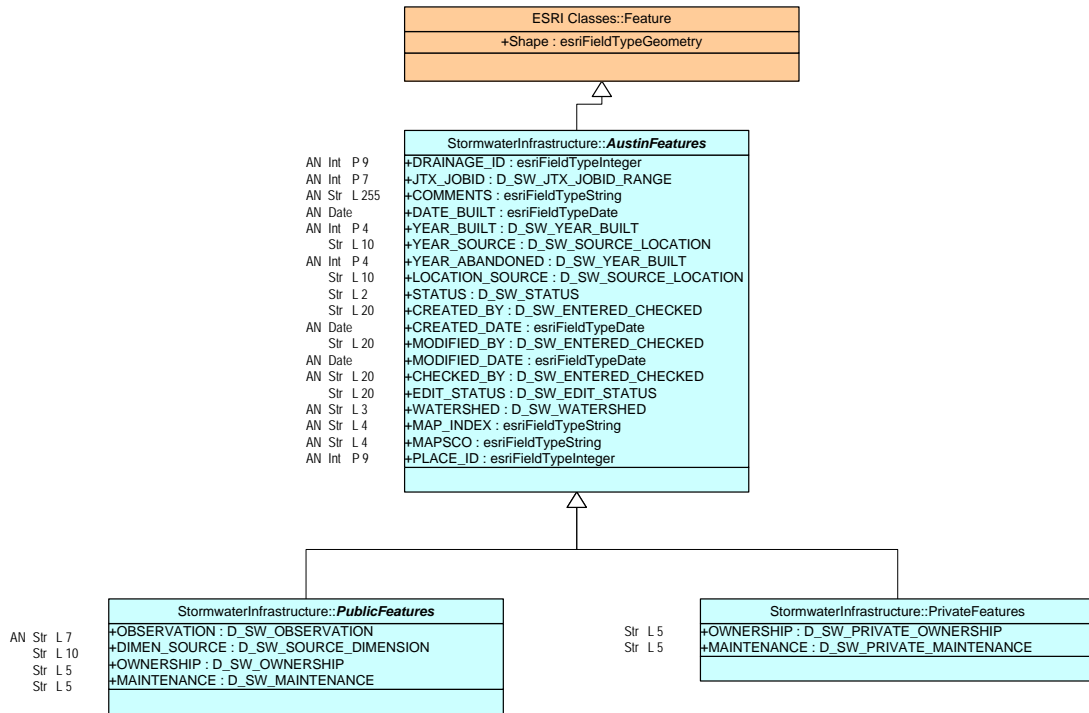
Justin Holder - City of Austin
Shannon Johnson - City of Austin
Lincoln Kauffman - City of Austin
Dave Lewis - City of Austin
Jennifer Loeffler - City of Austin
Rachel Moellmer - City of Austin
Chad Phelan - City of Austin
Logan Pugh - City of Austin
Nik Pullias - City of Austin
Garner Peterson - Parsons
Andrew Hands - Parsons

Author Information

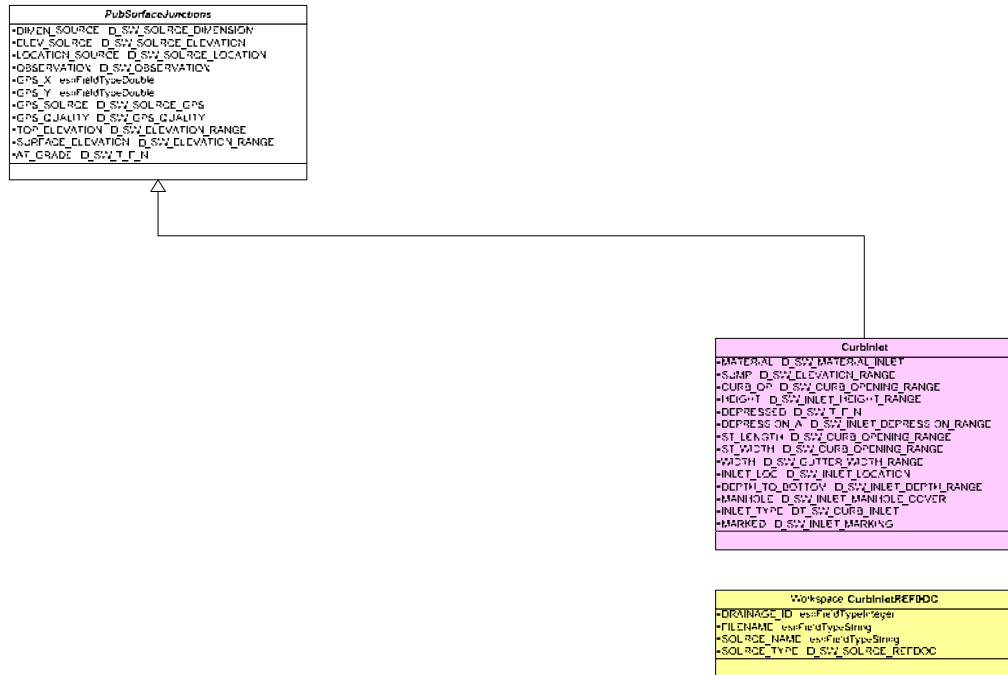
Andrew Rudin, Programmer Analyst Senior
City of Austin, Watershed Protection Department
505 Barton Springs Rd., 11th Floor
Austin, TX 78704
Ph: (512) 974-6319
Andrew.Rudin@ci.austin.tx.us

Appendix A – Geodatabase Design

Example of abstract feature classes



Example of abstract feature class attached to the actual curb inlet feature class. Also shows relation of the feature class to the REFDOC Table.



Appendix B – Data Dictionary

CurbInlet		Curb Inlet - An inflow point for surface storm water to be diverted into underground drainage pipes. Curb inlets are typically composed of an opening in the curb at street level which funnels water into a temporary storage chamber called a catch basin. The catch basin is drained by a drainage pipe.						
Field Name	Domain Name	Description	Allow Nulls?	Group Requesting Data	Notes	ESM	SSM	Construction Plans
OBJECTID	-	ArcSDE Internal Identifier.	NO	All		A	A	A
ANCLLARYROLE	-	Field used by geometric to determine source or sink	YES	All		A	A	A
ENABLED	EnabledDomain	Field used by geometric network to allow flow through the feature	YES	All		A	A	A
DRAINAGE_ID	-	Feature dependent unique ID.	YES	All		A	A	A
JTX_JOBID	D_SW_JTX_JOBID_RANGE	Primary key of job tracking record in JTX	YES	Data Management	JTX	A	A	A
ROTATION	D_SW_ROTATION_ANGLE_RANGE	Symbol rotation angle for cartographic representation	YES	All		M	M	M
COMMENTS	-	Field data collection comments.	YES	All		M-S (NULL)	M-S (NULL)	M-S (NULL)
DATE_BUILT	-	Date of construction completion (mm/dd/yyyy).	YES	Field Operations	AMANDA	+S (NULL)	+S (date)	M-S (NULL)
YEAR_BUILT	D_SW_YEAR_BUILT	Estimated or actual year of construction	YES	All		-S (NULL)	+S (year)	M-S (NULL)
YEAR_SOURCE	D_SW_SOURCE_LOCATION	Description of the source of the year built.	NO	All		+S (2-ESM)	+S (2-SSM)	-S (NULL)
YEAR_ABANDONED	D_SW_YEAR_BUILT	Estimated or actual year of abandonment	YES	All		-S (NULL)	+S (year)	M-S (NULL)
STATUS	D_SW_STATUS	Whether facility is in service, abandoned, etc.	NO	All		+S (In Service)	+S (In Service)	+S (In Service)
CREATED_BY	D_SW_ENTERED_CHECKED	Name of person to create feature	NO	All		A	A	A
CREATED_DATE	-	Date feature was created	YES	All		A	A	A
MODIFIED_BY	D_SW_ENTERED_CHECKED	Name of person to last update feature	NO	All		A	A	A
MODIFIED_DATE	-	Last date of data update	YES	All		A	A	A
CHECKED_BY	D_SW_ENTERED_CHECKED	Name of person to check last update	YES	All		A [Only in QC Step]	A [Only in QC Step]	A [Only in QC Step]
EDIT_STATUS	D_SW_EDIT_STATUS	Code describing current GIS data entry status	NO	Data Management		+S (Entered)	+S (Entered)	+S (Entered)
WATERSHED	D_SW_WATERSHED	Watershed name by 3-letter abbreviation.	YES	All	AMANDA (as full name)	A	A	A
MAP_INDEX	-	Austin grid map index.	YES	All		A	A	A
MAPSCO	-	MAPSCO grid page number and letter.	YES	All		A	A	A
PLACE_ID	-	Street centerline segment most closely associated with feature.	YES	All		A	A	A
OWNERSHIP	D_SW_OWNERSHIP	Inlet ownership.	NO	Field Operations	AMANDA	+S (City)	+S (City)	+S (City)
MAINTENANCE	D_SW_MAINTENANCE	Structure maintenance	NO	Field Operations	AMANDA	+S (City)	+S (City)	+S (City of Austin)
DIMEN_SOURCE	D_SW_SOURCE_DIMENSION	Description of source of dimension data.	NO	All		+S (2-ESM)	+S (2-SSM)	-S (NULL)
ELEV_SOURCE	D_SW_SOURCE_ELEVATION	Description of source of elevation data.	NO	All		+S (2-ESM)	+S (2-SSM)	-S (NULL)
LOCATION_SOURCE	D_SW_SOURCE_LOCATION	Description of source of location data.	NO	All		+S (2-ESM)	+S (2-SSM)	-S (NULL)
OBSERVATION	D_SW_OBSERVATION	Description of viewable condition.	YES	Watershed Engineering, Field Operations		+S (No Field Assessment)	+S (No Field Assessment)	+S (No Field Assessment)
GPS_X	-	GPS derived coordinate (original location to remain unchanged for historical purposes) (ft)	YES	All		-S (NULL)	-S (NULL)	M-S (NULL)
GPS_Y	-	GPS derived coordinate (original location to remain unchanged for historical purposes) (ft)	YES	All		-S (NULL)	-S (NULL)	M-S (NULL)
GPS_SOURCE	D_SW_SOURCE_GPS	Source of the GPS data	NO			+S (No GPS)	+S (No GPS)	+S (No GPS)
GPS_QUALITY	D_SW_GPS_QUALITY	Quality of the GPS data (sub-M or sub-CM)	YES			+S (No GPS)	+S (No GPS)	+S (No GPS)
TOP_ELEVATION	D_SW_ELEVATION_RANGE	The ground elevation at the inlet (ft). Top of curb.	YES	Watershed Engineering		+S (NULL)	+S (NULL)	M-S (NULL)
SURFACE_ELEVATION	D_SW_ELEVATION_RANGE	Rim elevation of inlet (ft). Top of inlet box.	YES	Watershed Engineering		-S (NULL)	-S (NULL)	M-S (NULL)
AT_GRADE	D_SW_T_F_N	Is structure at grade (True/False)	NO	Watershed Engineering		+S (No Data)	+S (No Data)	+S (No Data)
MATERIAL	D_SW_MATERIAL_INLET	Material of the structure.	NO	Watershed Engineering, Field Operations		+S (No Data)	+S (No Data)	+S (No Data)
SUMP	D_SW_ELEVATION_RANGE	Invert elevation of the Sump (ft).	YES	Watershed Engineering		-S (NULL)	-S (NULL)	M-S (NULL)
CURB_OP	D_SW_CURB_OPENING_RANGE	The inlet opening length (in).	YES	Watershed Engineering		-S (NULL)	-S (NULL)	M-S (NULL)
HEIGHT	D_SW_INLET_HEIGHT_RANGE	The opening height of an inlet (in).	YES	Watershed Engineering		-S (NULL)	-S (NULL)	M-S (NULL)
DEPRESSED	D_SW_T_F_N	Has depressed Gutter? (True/False)	NO	Watershed Engineering		+S (No Data)	+S (No Data)	+S (No Data)
DEPRESSION_A	D_SW_INLET_DEPRESSION_RANGE	Gutter depression (in) (the "a" value defined in Drainage Criteria Manual).	YES	Watershed Engineering		-S (NULL)	-S (NULL)	M-S (NULL)
ST_LENGTH	D_SW_CURB_OPENING_RANGE	Inside length of inlet structure (in).	YES	Watershed Engineering		+S (If ESM says 10' Inlet, enter here. Otherwise NULL)	+S (If ESM says 10' Inlet, enter here. Otherwise NULL)	M-S (NULL)
ST_WIDTH	D_SW_CURB_OPENING_RANGE	Inside width of inlet structure (in).	YES	Watershed Engineering		-S (NULL)	-S (NULL)	M-S (NULL)
WIDTH	D_SW_GUTTER_WIDTH_RANGE	Gutter width (in).	YES	Watershed Engineering		-S (NULL)	-S (NULL)	M-S (NULL)
INLET_LOC	D_SW_INLET_LOCATION	Is the inlet located in sag or on grade	NO	Watershed Engineering		+S (No Data)	+S (No Data)	+S (No Data)
DEPTH_TO_BOTTOM	D_SW_CODE_DEPTH_RANGE	Depth to bottom of inlet sump box (in) (subtracted from RIM elevation as feet to derive inlet invert SUMP elevation).	YES	Watershed Engineering		-S (NULL)	-S (NULL)	M-S (NULL)
MANHOLE	D_SW_INLET_MANHOLE_COVER	Manhole present? (Present/Not Present/Covered)	NO	Field Operations		+S (No Data)	+S (No Data)	+S (No Data)
INLET_TYPE	DT_SW_CURB_INLET	Type of inlet as defined by Drainage Criteria Manual	NO	All		+S (No Data)	+S (No Data)	+S (No Data)
MARKED	D_SW_INLET_MARKING	Condition of inlet "Drains to Creek" marking	NO	ERM		+S (NULL)	+S (NULL)	+S (No Data)
GLOBALID	-	Unique system-maintained identifier	NO	All		A	A	A
SHAPE	-	All	NO	Automatic		A	A	A
ArcSDE maintained fields								
Abstracted attributes common to more than one feature class								
Feature class specific attributes								

Appendix C – JTX Construction Plan Interface

Job Query Results - List View

JOB ID	Name	Description	Started Date	Assigned To
84780	DIG_Const_Plan_Entry_JOB_84780			holderj
84779	DIG_Const_Plan_Entry_JOB_84779			SW_Editors
84778	DIG_Const_Plan_Entry_JOB_84778			SW_Editors
84777	DIG_Const_Plan_Entry_JOB_84777			SW_Editors
84776	DIG_Const_Plan_Entry_JOB_84776			SW_Editors

Extended Properties

Stormwater Plan Detail

OBJECTID: 4304

FOLDERRSN: 504635

JOB_ID: 84777

*PLAN_TYPE: [Dropdown]

REFERENCEFILE: PPC-1-A-2706

ORIGINAL_REVIEW_BY: [Dropdown]

ORIGINAL_QC_BY: [Dropdown]

PLAN_DATE: 6/10/1954 [Dropdown]

PLAN_YEAR: 1954

ASBUILT: False [Dropdown]

ERROR: True [Dropdown]

DRAINAGE: True [Dropdown]

PONDS: False [Dropdown]

EROSION: False [Dropdown]

EASEMENTS: [Dropdown]

Andrew Rudn GISWPDR 8:57 AM

Appendix D – Construction Plan Data Entry Workflow from JTX

The screenshot displays the JTX - Job Tracking for ArcGIS application. On the left is a 'Jobs' tree view with categories like 'public queries', 'DIG Queries', 'Pond Queries', 'Global Queries', and 'User Queries'. The main window is split into two panes: 'Job Query Results - List View' and 'Job Workflow'.

Job Query Results - List View

JOB ID	Name	Description	Started Date	Assigned To
84780	DIG_Const_Plan_Entry_JOB_84780			holderj
84779	DIG_Const_Plan_Entry_JOB_84779			SW_Editors
84778	DIG_Const_Plan_Entry_JOB_84778			SW_Editors
84777	DIG_Const_Plan_Entry_JOB_84777			SW_Editors
84776	DIG_Const_Plan_Entry_JOB_84776			SW_Editors

Job Workflow

The workflow diagram is a vertical flowchart with several stages:

- Editor Steps:** Includes 'Create version' (yellow parallelogram), 'Edit data' (yellow rounded rectangle), and a red rectangle.
- Peer Reviewer Steps:** Includes 'Peer review data' (green rounded rectangle), a green diamond decision point, and a green oval connector.
- QC Steps:** Includes 'QC data' (blue rounded rectangle), a blue diamond decision point, and a blue oval connector.
- Version Post Steps:** Includes 'Post data' (yellow rounded rectangle), a yellow diamond decision point, and a yellow oval connector.
- Close Job:** Final steps at the bottom of the flowchart.

Arrows point from text labels to specific steps in the workflow diagram. The status bar at the bottom shows 'Andrew Rudin GISWPDR' and the time '8:59 AM'.

