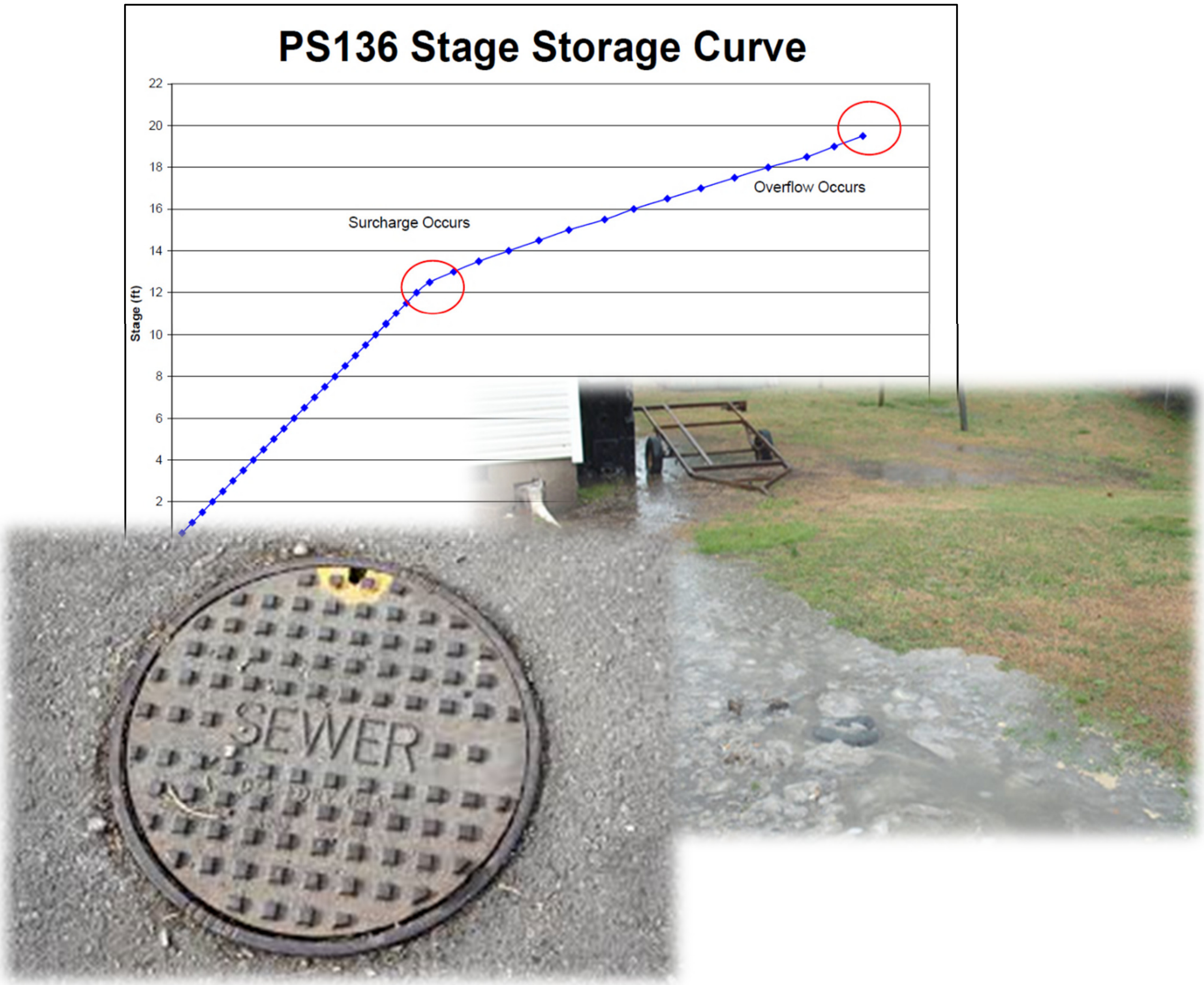


Determining Sanitary Sewer Gravity System Storage: A Standardized Approach

Shonia Holloway, GISP

Pennsylvania State University

December 2015

CONTENTS

1 INTRODUCTION

Local, state, federal governments, and private entities that own and operate sanitary sewer collection and conveyance systems need to know the capacity of their systems in order to and prevent releases of wastewater into the environment. A primary concern of system operators is the capacity of the system or the total amount the system can successfully convey from the collection point to a treatment facility. If the capacity of the system is inadequate and wastewater is released into the environment, called a Sanitary Sewer Overflow (SSO, see Figure 1), then owners and operators are in violation of the Clear Water Act and can be held liable for such releases through fines and/or disciplinary action.

Figure 1. Example of Sanitary Sewer Overflow (SSO).

This concern requires a significant amount of focus and effort to be placed on calculating flow and developing hydrologic/hydraulic models of the systems. The process for determining the capacity, or amount of sewage flow a system can accommodate without SSOs, is called a *Capacity Analysis* and is typically conducted when new systems are designed and existing systems are being rehabilitated. As illustrated in Figure 2, the main components of a capacity analysis are:

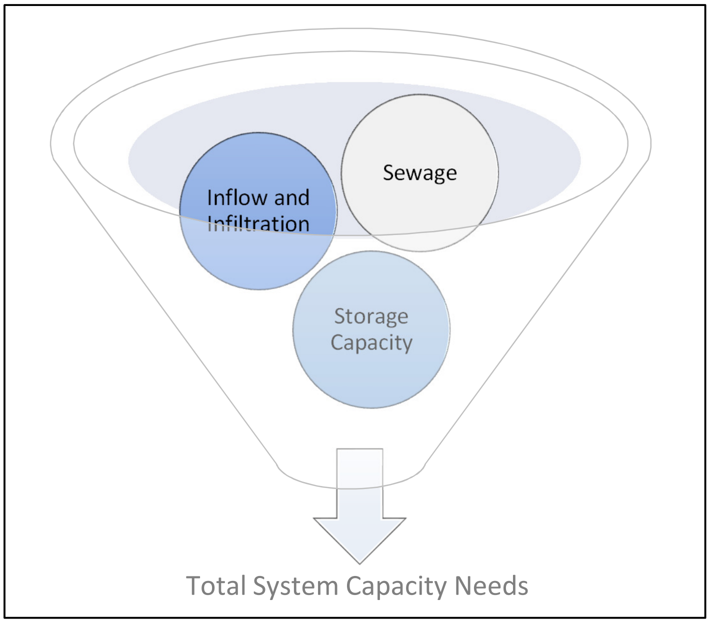


Figure 2. Main components of a Capacity Analysis.

- Sewage – this is the wastewater flow that is expected to be in the system from sewer system customers, like residential homes and/or businesses
- Inflow and Infiltration (I&I) – this is water (rainwater and/or groundwater) that unintentionally gets in sanitary sewer systems through defects in the system – like cracks in pipes or uncapped cleanouts. Excessive I&I can reduce the capacity of the system and be a significant contributor to SSOs.
- Storage Capacity – this is the storage capacity that is provided in the system through various means including storage tanks, wet wells, pipes, and manholes.

In order to accurately monitor peak sewer inflow (the highest rate of flow during a significant wet weather event), utilities should be accounting for the storage in the gravity systems. Otherwise, the apparent capacity of the system is smaller, which would indicate a need to reduce I&I, and rehabilitate pump stations (i.e. larger pumps and wet wells) in order to reclaim capacity. These projects would consume resources that might be better applied elsewhere in the system.

This project demonstrates the development of a standardized toolset that incorporates gravity storage volumes into capacity analyses, allowing utilities to accurately account for flow into the pump station, a key parameter to understanding the amount of I&I in their system taking up capacity for sewage. The tool includes a data model that provides a standardized data template, as well as a toolset that provides the ability to load data, run the analysis, and export results in the form of tabular and/or mapping products. The exports will allow entities to view, analyze, and manage result files using off-the-shelf software like Microsoft Excel and ESRI ArcGIS Desktop. Preliminary development has illustrated that accounting for gravity storage volumes in a system has proven to be critical in evaluating the capacity of sanitary sewer systems during wet weather events, and that in order to make accounting for the gravity storage volumes practical, an automated tool is required.

1.1 Background

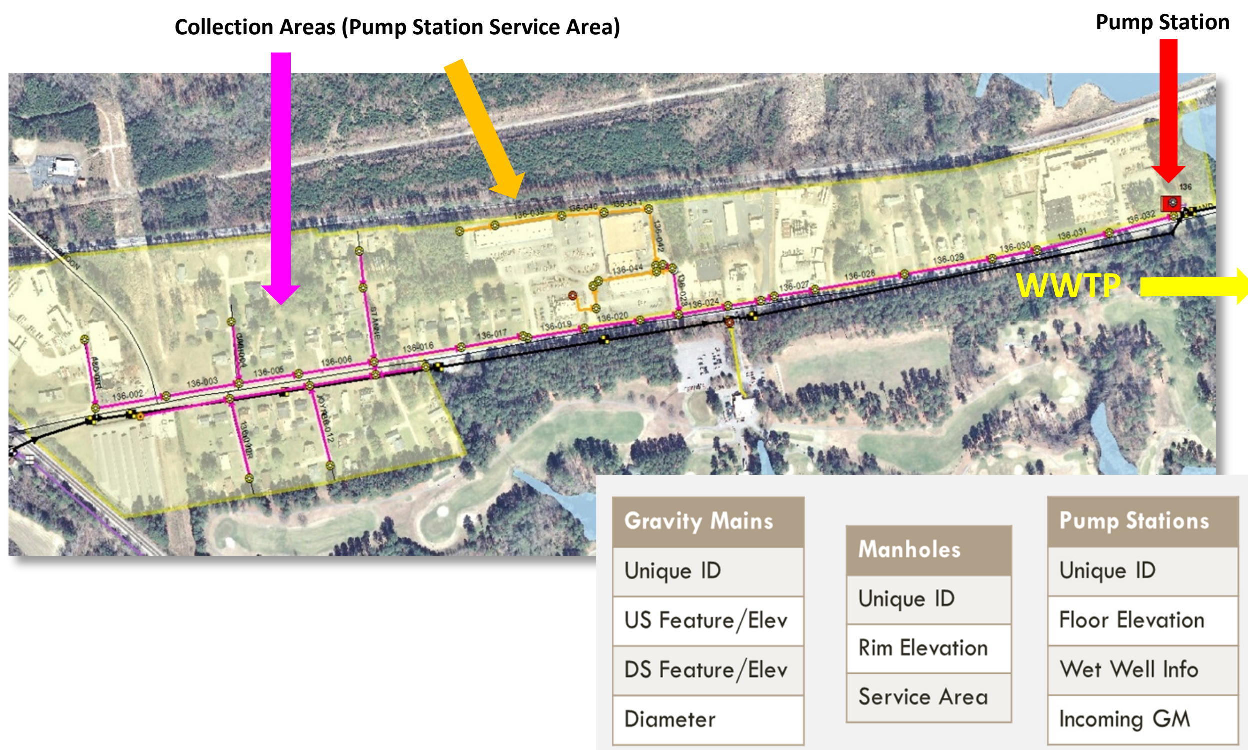


Figure 3. Example of a wastewater gravity collection system.

Sanitary Sewer gravity systems are typically underground piping systems that convey sewage (wastewater) from residential, commercial and industrial buildings. As illustrated in Figure 3 above, the sewage is collected from buildings through a lateral, that empties into a system of gravity pipes (purple and orange lines) which convey the sewage by gravity to a pump station (red rectangle). Pump stations are built in various locations throughout the entire system to accommodate for flow by gravity. The gravity pipes and manholes that convey the flow to a particular pump station, as a group, are typically referred to as the *Pump Station Service Area* or *Collection Area*. Once the sewage reaches the pump station, it empties into a large container, called a wet well. The sewage levels in the wet well can fluctuate based on the amount of flow coming into the station. Pumps in the station are used to force the sewage into a system of pressurized pipes and eventually convey it to a wastewater treatment plant (WWTP). Once at the WWTP, the sewage is treated and ultimately discharged to a water body.

One of the most important factors for large, public sewer systems, is the storage of asset information in a geographic information system (GIS). Organizations will typically store not only the location of the assets in the GIS but also various attributes about those assets that are used for various operations, maintenance, and management activities. As shown in Figure 3, the various attributes reqired for capacity analysis, regarding the pump stations, manholes, and gravity mains, are typically available in the organization's GIS.

1.1.1 Gravity System Storage

As discussed above, once the sewage reaches the pump station, it empties into the wet well and depending on the amount of flow, the sewage levels in the wet well can fluctuate. As illustrated in Figure 4, if the pumps in the pump station can not keep up with the amount of flow into the wet well, eventually the sewage in the wet well can reach a level (elevation) where it will begin to backup into the gravity system upstream of the pump station. This sewage is now stored in the gravity system and eventually, if the levels rise high enough, will reach a point in the system where the sewage is released as a SSO.

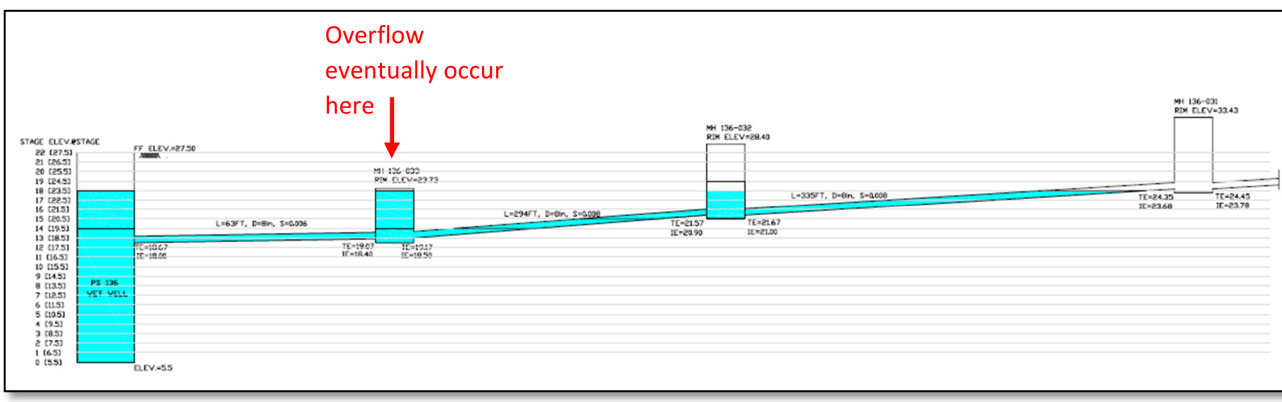


Figure 4. Gravity System Storage – illustrating wastewater being stored in the pump station wet well, gravity mains, and manholes. Blue indicates the available storage in the gravity system, at which point when the lowest elevation is reached, an SSO will occur.

This volume of sewage in the system is what is known as *Gravity System or Stage Storage Volume*. Typically the volume is illustrated as a graph, as shown in Figure 5. The X-axis is the Stage Elevation, or the increment of rise of the sewage in the wet well, in this case expressed as feet. The Y-axis is the Total Volume in the system for the wet well, gravity pipes, manholes, and laterals, in this case expressed as cubic feet.

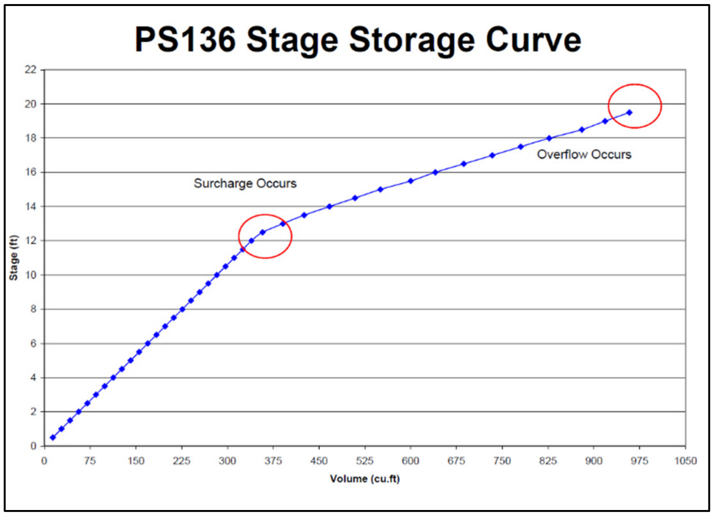


Figure 5. Example Stage Storage Curve.

The initial deflection shown on the curve is typically the elevation where the surcharge (or backup into gravity system) begins until it reaches the elevation where a release or SSO occurs, which is typically the last point in the curve.

1.1.2 Original Stage Storage Tool

In 2007, the City of Suffolk and 12 additional Hampton Roads localities and the Hampton Roads Sanitation District entered into a Consent Order with the Virginia Department of Environmental Quality to ensure the reduction and/or elimination of SSOs from their collection systems. The Consent Order contained various activities and milestones aimed at collection system operators analyzing and determining an accurate picture of the condition and capacity of their systems in order to eventually develop a strategy for managing wet weather impacts to the sanitary sewer system (Regional Wet Weather Management Plan). As part of these efforts, the City of Suffolk recognized that it was necessary to account for the storage in the gravity system to accurately monitor peak sewer flow in the gravity system and a custom tool was developed to calculate the City of Suffolk's gravity storage volume (also called *Stage Storage*). A significant amount of research, development, and testing was required for the initial custom tool developed for the City. The preliminary development of the database and toolset has resulted in the following findings:

- Accounting for gravity storage volumes in a system has proven to be critical in evaluating the capacity of sanitary sewer systems during wet weather events.
- To make accounting for the gravity storage volumes practical, an automated tool is required (spreadsheet/hand calculations are infeasible for large, relatively flat, systems).
- Collection and conveyance system configurations are not uniform and vary in many factors (i.e. design, condition, and quality of construction), therefore a thorough understanding of those variations is necessary to develop a standardized toolset that addresses the potential system configurations appropriately.
- In order to obtain the widest user base, a ubiquitous platform (or technology) should be selected to develop and deliver the database and toolset (i.e. python).

The tool developed in 2007 is developed in Microsoft Access 2003 (Figure 6) and provides the following functionality:

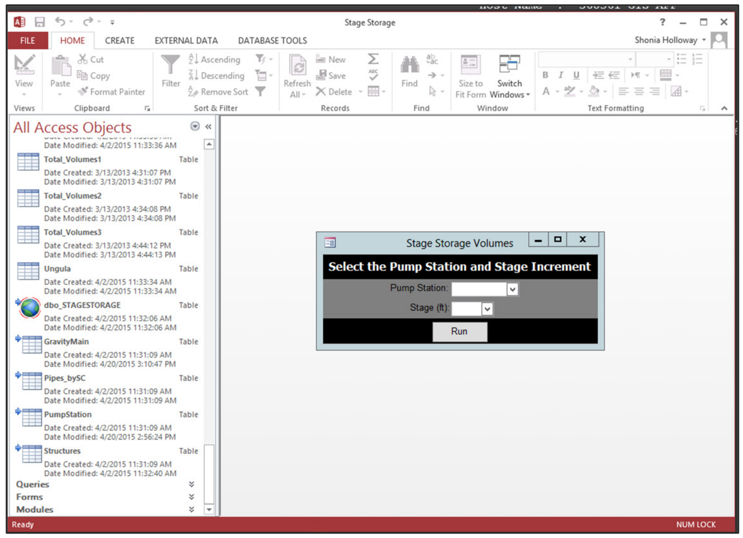


Figure 6. Original Stage Storage Tool, City of Suffolk, Virginia.

- Linkage to sanitary sewer asset data in a separate ESRI personal geodatabase for pump stations, manholes, and gravity mains. The sewer asset data is exported directly from the City's sanitary sewer dataset in their GIS. The type of information required by the current tool includes, but is not limited to:
    - Pump Station identifiers, wet well dimensions, finished floor elevations, incoming gravity pipe invert elevations
    - Manhole identifiers, rim elevations
    - Gravity pipe identifiers, invert elevations, upstream and downstream manholes, and diameters
- Logic written in Visual Basic for Applications (VBA) and through the use of queries to run the stage storage calculations and provide an output volume table.
- Very simplistic user interface to select the desired pump station and stage elevation increments (rise of water in wet well increments, i.e. 0.1 feet).

No current integration with GIS exists other than the linkage to the tables in the personal geodatabase.

2 Purpose and Objectives

The purpose of this project is to develop a standardized toolset, including a database template and stage storage calculation interface that sewer system operators can easily and inexpensively incorporate into their flow monitoring system and capacity assessment process.

Based on the findings from the initial development supported by the City, it was determined that a more robust, standardized toolset that integrates easily with off-the-shelf GIS products would best serve the data import, curve generation, and curve/data quality control procedures that are required during curve generation.

The objectives of the toolset are to provide:

- Importing of a variety of data schemas into a standard geodatabase template for consistent usage in the toolset, specifically the ESRI Locality Government Information Model (LGIM) for wastewater.
- Integration of the data review and quality control processes with a spatial visualization component (through ESRI ArcMap interface) that would provide the user with the ability to visualize which assets in the system would experience storage and/or surcharge.
- Use with a single toolset that supports the import, calculation, visualization, and quality control procedures required to generate accurate stage storage volumes.

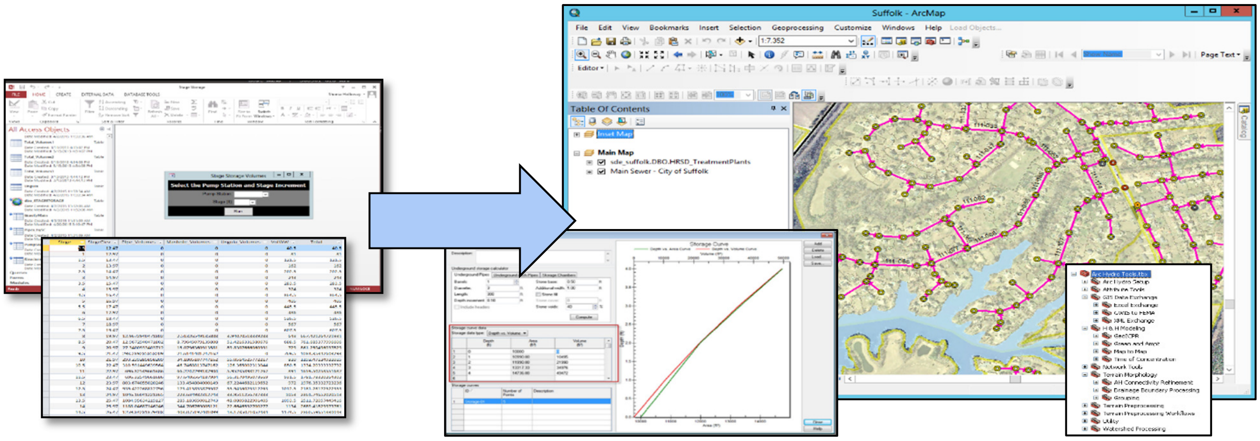


Figure 7. Illustration of expected outcome of new toolset.

As illustrated in Figure 7, the expected outcome was a standardized, packaged tool that can be used on an ESRI ArcGIS 10.X Desktop and Microsoft Windows platform. This toolset will not alter any of the engineering calculations or logic used to generate the stage storage volume curves.

2.1 Literature Review

Once the concept for the new toolset was determined, the first step was to ensure that a similar tool did not exist in the market place, specifically for sanitary sewer. A literature review was conducted researching and analyzing the following components:

- Stage Storage - What is it and why is it important? To answer this question, a review of papers, articles, journals, user manuals, websites, and other resources was conducted that focused on the engineering concepts behind gravity storage and why it is important in regard to wastewater utilities.
- Available on the Market - How is stage-storage currently handled by applications already on the market? To answer this question, a review of current applications and tools on the market that support the generation of gravity storage volumes was conducted. As necessary, documentation of the various software tools was reviewed as well as discussions with software vendors and end-users of the products.
- Building Standardized Tools - What are standardized tools and how are they built? To answer this question, a review of papers, articles, journals, best practices, websites, and other resources in the information technology industry was conducted. This review was done in order to determine the best approach for creating a tool that can accommodate users from multiple organizations.
- ESRI Data Model - What are the ESRI Data Models and their associated tools? To answer this question, a review of how the ESRI data models support tools that are out on the marketplace was conducted. In order to make this tool ‘portable’, it is beneficial to adopt a data model template that is already popular and familiar to utilities in the marketplace. An understanding of how other tools use the data models was also conducted by researching various offerings by ESRI and their business partners.

Based on the findings from the review, it has been determined that developing a standardized tool that can be used by various entities to calculate gravity storage volumes in their wastewater systems would be beneficial and fill a gap that currently exists for this type of tool in the marketplace. The following conclusions provide the basis for validation that further research, planning, and execution of the development of the tool is warranted and could provide a direct benefit to wastewater utilities capacity assessment activities:

- The calculation and logic used to develop stage storage volumes in the City of Suffolk case study are valid and accurate.
- A stand-alone standardized tool to calculate stage storage volumes that integrates a wastewater utilities GIS data does not currently exist.
- The methods used to develop a software product are well documented and can provide a basis for the approach that will be used to develop the tool.
- The ESRI data models have been well adopted and are well understood in the marketplace and would be a viable option for the back-end data storage mechanism for the tool.

3 Design and Development

After it was determined the tool would be a viable option for calculating sanitary sewer gravity storage, the next phase of the project was to develop and implement a design and development approach. As shown in Figure 8, the typical iterative software development lifecycle was employed, where design and development was accompanied by iterative review by outside entities. An iterative approach has the following benefits:

- Allows for a more active role in guiding the project and aids in the entire project team understanding the ultimate design and solution.
- Builds a relationship of trust between all parties by fostering their continued understanding of the risks and decisions and progress being made on the project.
- Allows for time to react from lessons learned and integrate them into the project.
- Revisits the progress and refreshes understanding on the ability to deliver on a consistent basis.
- Allows the end-user to fully understand the project as whole.
- Aids in managing end-user expectations and understanding of project requirements.

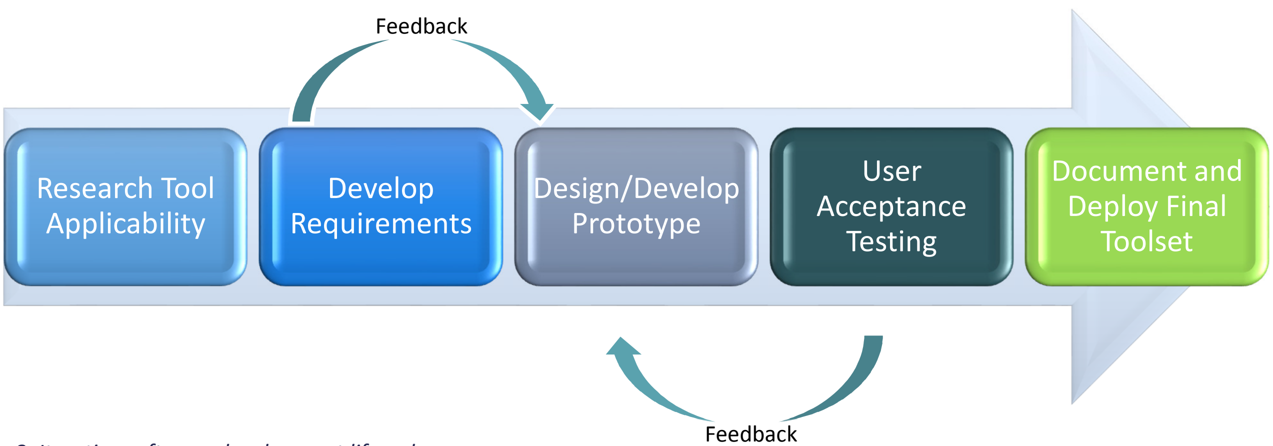


Figure 8. Iterative software development lifecycle.

3.1 Develop Requirements

Once it was determined the development of such a tool would be applicable and viable, then the specific requirements for the toolset functionality had to be developed. At a minimum, the toolset needed to accomplish the tasks that were already available in the current tool. In addition, a more robust, standardized toolset is desired that integrates easily with off-the-shelf ArcGIS Desktop products to best serve the data import, curve generation, and curve/data quality control procedures that are required during curve generation. Such a toolset would provide the:

- Import of a variety of data schemas into a standard geodatabase template for consistent usage in the toolset.
- Integration of the data review and quality control processes with a spatial visualization component (through ESRI ArcMap interface) that would provide the user with the ability to visualize which assets in the system would experience storage and/or surcharge.
- User with a single toolset that supports the import, calculation, visualization, and quality control procedures required to generate accurate stage storage volumes.

In order to accomplish these goals, the following functionality requirements were included in the Stage Storage Toolset:

1. Use of the most recent ESRI Local Government Information Model for sanitary sewer as the standardized geodatabase template.
2. Tool interface, available through ArcGIS Desktop, including:
    a. Tool for user to import their sanitary sewer GIS data into the standardized geodatabase template and save import configurations for later use.
    b. Tool for user to run quality control (QC) checks on data after import.
    c. Tool for user to run stage storage volumes, view results in tabular and graphic format, and export results into Microsoft Excel.
3. Map interface, available through ArcGIS Desktop, including:
    a. Visual display and linkage to imported sanitary sewer information in standardized geodatabase template in ArcMap.
    b. Visual display and linkage to stage storage results in ArcMap.
4. Development of help documentation for end users.

3.2 Design and Develop Prototype

A prototype tool was designed and developed that could be reviewed and tested by end-users. This process entailed three main components – the selection of the technology that would be used to build the toolset, design of the toolset, and development of a prototype.

3.2.1 Technology Selection

One of the most important steps in the design process was determining which technology would be used to build the toolset. In the GIS industry, specifically for building tools that can integrate with ESRI ArcGIS Desktop, many options are available. Therefore, a list of criteria had to be developed in order to determine the main objective of the tool and narrow down the technology options so a selection could be made. Based on the goals and objectives of the tool and the intended end-user audience, the following selection criteria for the technology platform was developed:

1. Had to be a platform that was accessible by most of the organizations in the Hampton Roads, Virginia region, which would be the main focus area for the tool distribution,
2. Had to be readily available as part of an out of the box (OTB) solution and would not require any additional components that would cause users to incur additional costs, and,
3. Had to be easily updated/customizable by normal GIS users, without the need for specialized training in advanced coding technologies, like C#, .NET, etc.

Several options for extending the ESRI ArcGIS Desktop software packages were reviewed against these selection criteria, including a custom add-in, python add-in, python toolbox, ArcGIS Pro, and ArcGIS toolbox using python scripts. Based on the comparison of each technology against the selection criteria and a concurrent review of the strengths and limitations of each, it was determined that the toolset would be best suited to building within an ArcGIS toolbox using python scripts.

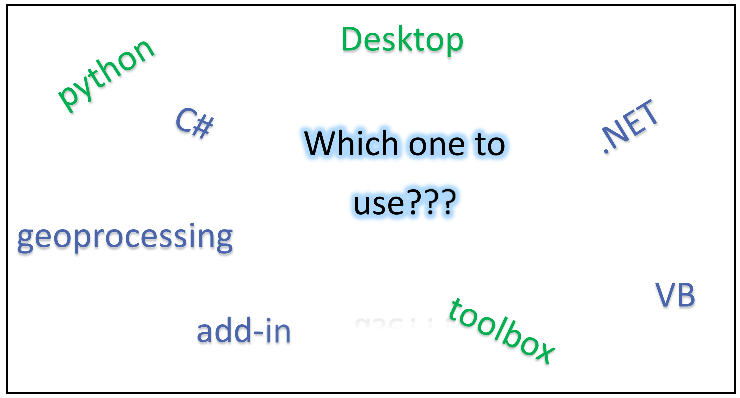


Figure 9. Technology Selection.

3.2.2 Design Process

Once the final technology was selected, then the toolset had to be designed to meet the functional requirements as outlined in Section 3.1. In order to accomplish this task, two main phases of design were conducted: 1) determination of the standardized database model that would be used; and 2) review of the available python parameters to determine if they would meet the requirement objectives.

3.2.2.1 Determination of Standardized Data Model

Since the current (previously developed) tool only ran on the City of Suffolk sanitary sewer dataset schema and one of the functional requirements was to develop a new tool that would run on a standardized data platform, several options were reviewed:

1. Continue to build the tool on the City of Suffolk data model and have other organizations import their data to adhere to the City's schema. This option would not break any of the current tools functionality or logic.
2. Develop a complete custom data model that was specific to the stage storage tool and calculations and would not contain any ancillary attribute information outside of what was needed for the tool. This option would break the current tools functionality and logic.
3. Utilize the ESRI LGIM for wastewater and have other organizations import their data to adhere to the City's schema. This option would not break any of the current tools functionality or logic.

Because of the ubiquitous nature of the data model, the familiarity of the local organizations with the data model, and the ability to expand storage of asset information beyond what was needed for the stage storage tool, option 3, a dataset called SewerStormwater within the ESRI LGIM, was selected to be used as the standard data model for the toolset.

After the selection, it was determined that it would be beneficial to review a broader sample of the types of data that would be loaded into the standardized data model. Therefore, three local organizations (the City of Virginia Beach, City of Portsmouth, and City of Suffolk, VA) all provided their sanitary sewer datasets from their GIS operations for review and analysis. A data crosswalk, as shown in Figure 10, was developed for each organization to determine:

- The target feature classes in the LGIM SewerStormwater dataset would be utilized in the tool
- The target fields in LGIM SewerStormwater dataset would be utilized in the tool
- The incoming feature class from the organization that would load into the target feature class
- The incoming fields and field types that would be used (input features) to populate the standardized LGIM data model (target features) and any issues or comments with each field

Feature Class	Target Field Description	Target Field Name	Target Field Type	Incoming Feature Class	Incoming Field Name	Incoming Field Type	Comments
ssNetworkStructure	Asset ID	FacilityID	Text, 20	PumpStation	FACILITYID	Text, 30	
	Floor Elevation	FLOORELEV	Double, 0, 0		FLOORELEV	Double, 0, 0	
	Wet Well Shape	WETWELLSHAPE	Text, 1		WWSHAPE	Text, 3	
	Wet Well	WETWELLDIA	Double, 0, 0		WWLENGTH	Double, 0, 0	
	Wet Well Length	WETWELLLEN	Double, 0, 0		WWBREADTH	Double, 0, 0	
	Wet Well Depth	WETWELLDEPTH	Double, 0, 0		WWDEPTH	Double, 0, 0	
	Incoming Gravity Main ID	WETWELLINCOMINGPIPE	Text, 50		Need to add in pre-processing	Text, 50	Can add in because have '*-000' in the DS_STRUC of GravityMain for ones flowing
	Wet Well Invert Elevation	WETWELLINVELEV	Double, 0, 0		WW_INV_ELEV	Double, 0, 0	
ssManhole	Asset ID	FacilityID	Text, 20	Structures	FACILITYID	Text, 30	Have to filter for STRUC-TYPE = "M" or "C"
	Rim Elevation	RIMELEV	Double, 0, 0		RIMELEV	Double, 0, 0	
	Pump Station Service	LOCDESC	Text, 200		SERVICEAREA	Text, 5	
	In Service	ACTIVEFLAG	Boolean		Need to add in pre-processing	Boolean	Have to add boolean field, calc by POSTING = 4 for Active
	Asset ID	FACILITYID	Text, 20		FACILITYID	Text, 30	
	Diameter	DIAMETER	Double, 0, 0		DIAMETER	Double, 0, 0	
	Upstream Manhole ID	FROMMH	Text, 11		US_STRUC	Text, 30	Changed LGIM to 255

Figure 10. Example of data cross-walk.

Once the cross-walks were completed for all three organizations, it was further validated that the ESRI LGIM SewerStormwater dataset would prove a viable option for serving as the standard stage storage data model.

3.2.2.2 *Review of Available Python Scripting Parameters*

The second component to the design process was to ensure that the technology selected (python scripts) would provide what was needed to complete all of the functional requirements. To make this determination, a functionality matrix was developed (Appendix A) of all the major functional components and each major component was researched and a preliminary prototype was developed. The main intention of each of these prototypes was to demonstrate that the functionality required would be achievable using python scripting from within an ArcGIS toolbox. For example, the prototype for the 'Generate Results' functionality is shown in Figure 11. This prototype was built to ensure that python parameters and scripting could be used to achieve the generation of a graph and output table based on calculations provided from the stage storage volume tool. This ensured that the desired outcome of this particular tool was achievable. A prototype similar to the one shown was developed for each component of the toolset – from importing of the data, to QC, to results generation.

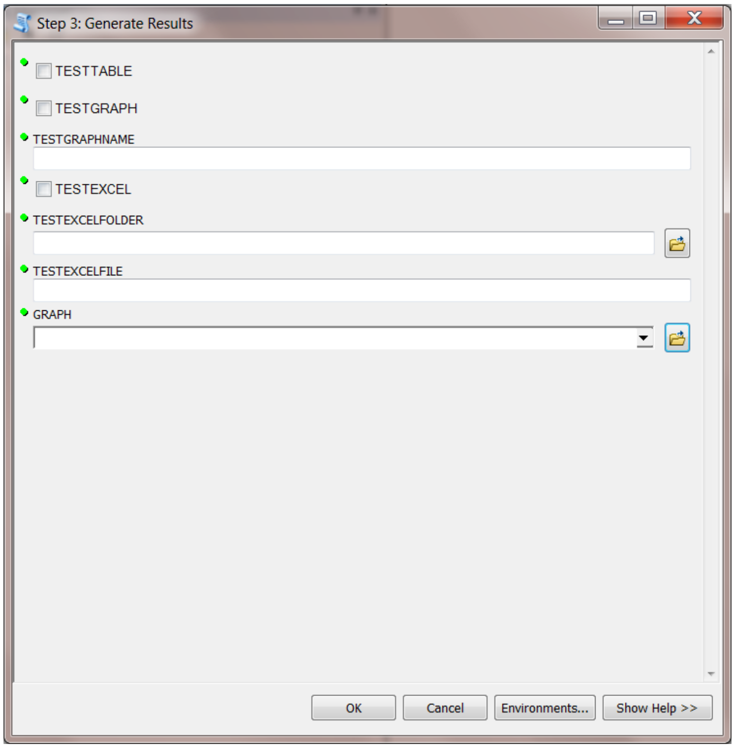


Figure 11. Example of toolset prototype to confirm python scripting would accomplish functional requirements.

3.2.3 Development of Prototype

Aside from ensuring the selected technology would meet the needs of all the functional requirements, the development of initial tool prototypes lead to a better understanding of:

- Tool workflow – how the user should interact with each step of the tool to generate the final results. Final prototype of tool workflow shown in Figure 12.
- Tool layout – how the toolset should be designed within the ArcGIS toolbox as well as within the ArcGIS ArcMap interface. Final prototype of tool layout shown in Figure 13.

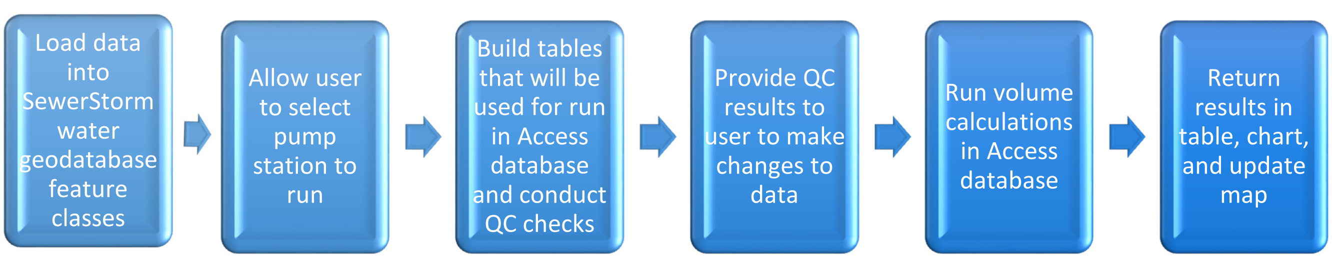


Figure 12. Prototype workflow from user perspective.

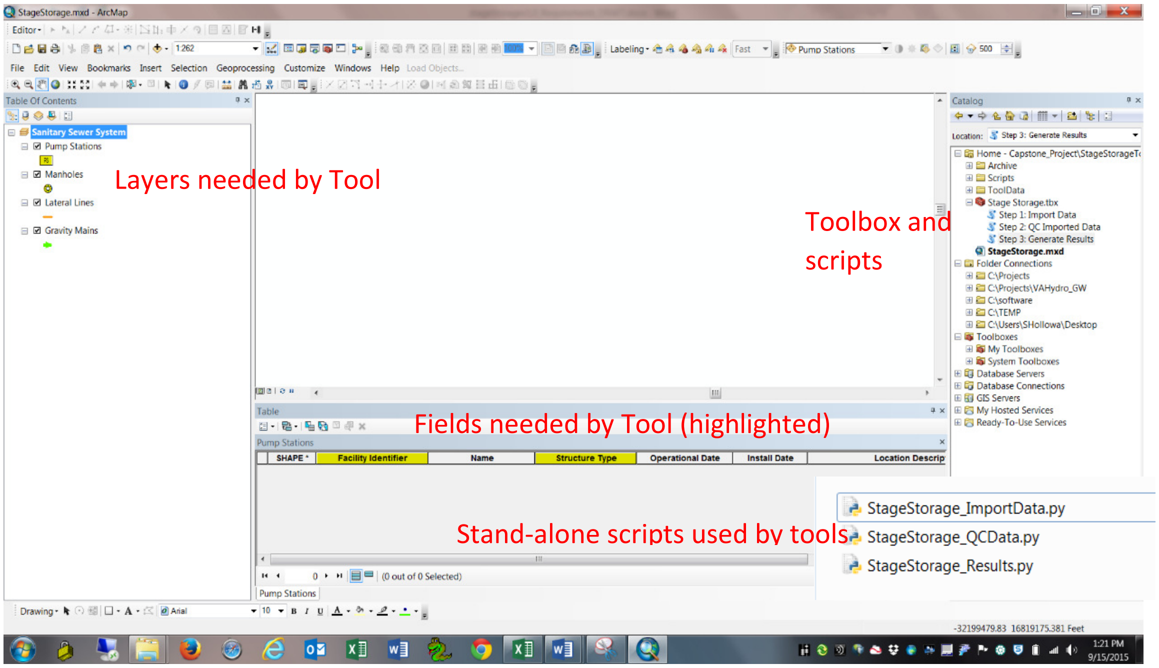


Figure 13. Prototype map document, toolbox, and scripts.

After the initial prototype was developed, full development started on each component of the toolset to accomplish the entire user workflow. Development involved updating the ArcMap document interface (table of contents, symbology, layer names, field properties, etc...), updating and enhancing the 'Stage Storage' toolbox with python scripts, setting of the associated script parameters, and development and testing of the stand-alone scripts that support each tool. After several development iterations for each tool, a final toolset (Figure 14) was complete, with details for each tool outlined in Table 1.

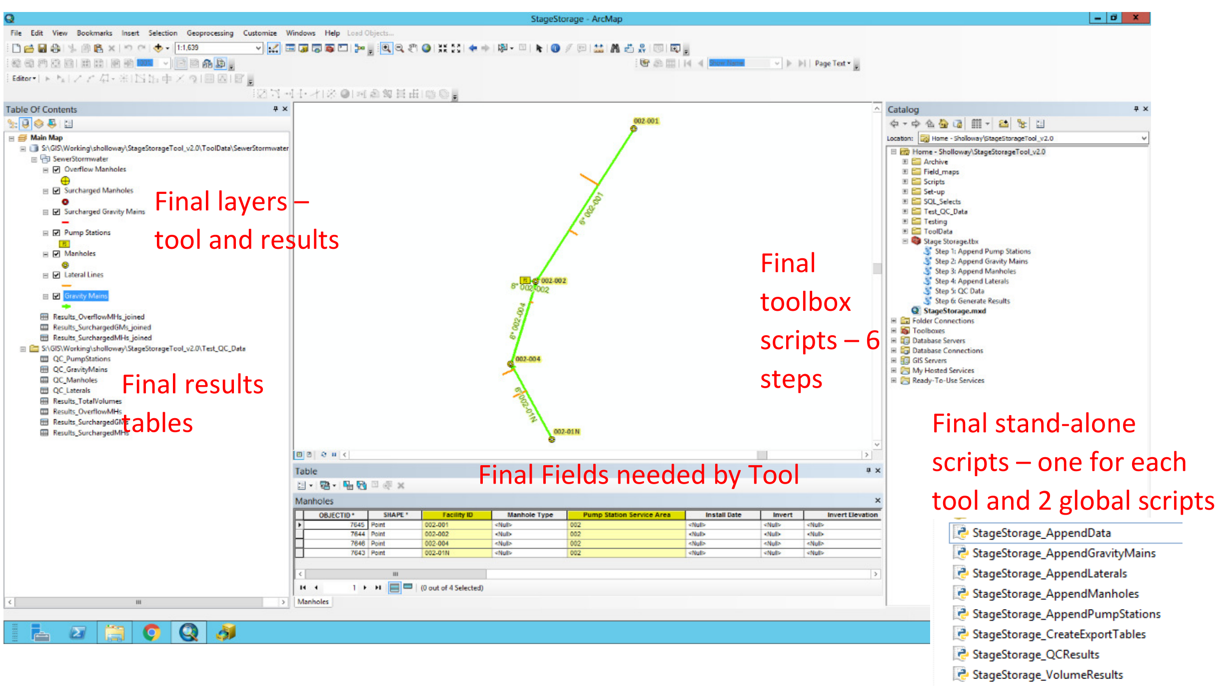


Figure 14. Final map document, toolbox, and scripts.

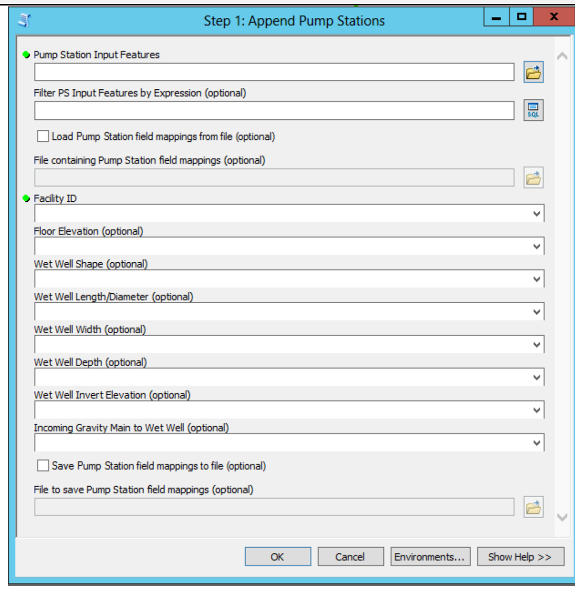
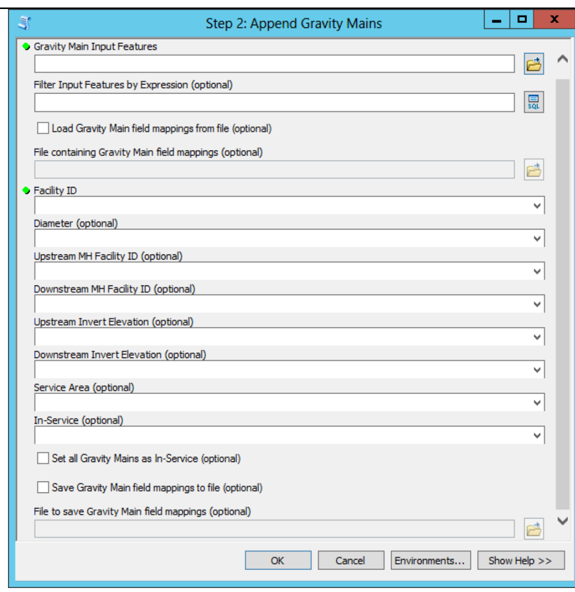
Table 1. Description of each tool within final prototype toolset.		
Tool Name and Description	Screenshot of Python Script Dialog	Name of Standalone Python Script(s) that supports tool and modules used
Append Pump Stations – appends selected user pump stations to the ssNetworkStructures feature class in the LGIM		StageStorage_AppendPumpStations and StageStorage_AppendData – uses os and arcpy modules
Append Gravity Mains – appends selected user gravity mains to the ssGravityMain feature class in the LGIM		StageStorage_AppendGravityMains and StageStorage_AppendData - uses os and arcpy modules

Table 1. Description of each tool within final prototype toolset.

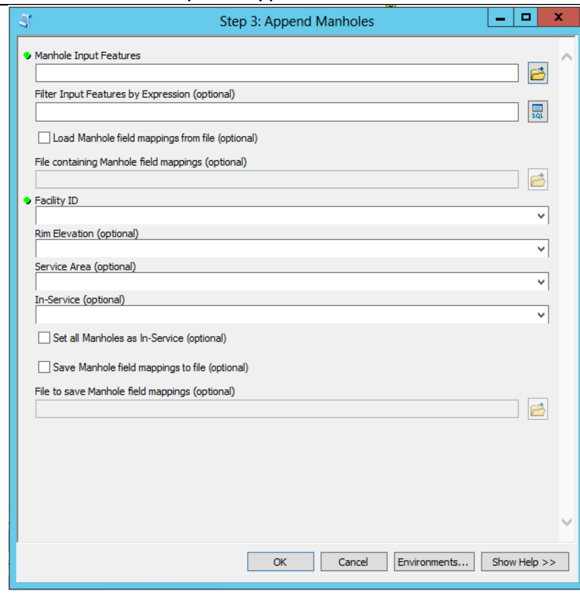
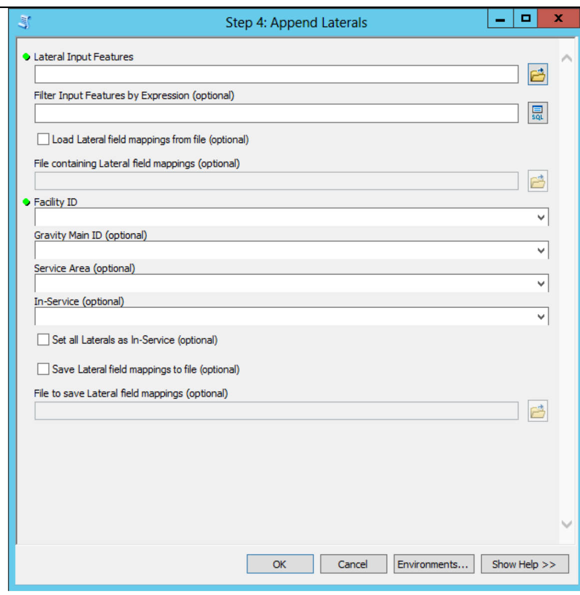
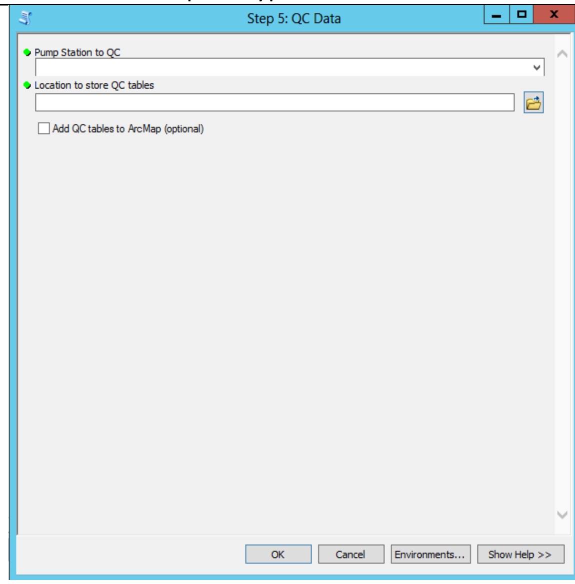
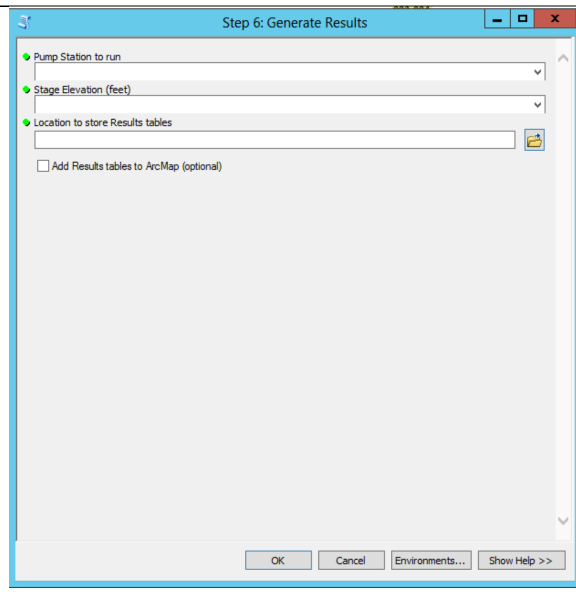
Append Manholes – appends selected user manholes to the ssManhole feature class in the LGIM		StageStorage_AppendManholes and StageStorage_AppendData - uses os and arcpy modules
Append Laterals – appends selected user laterals to the ssLateralLine feature class in the LGIM		StageStorage_AppendLaterals and StageStorage_AppendData - uses os and arcpy modules

Table 1. Description of each tool within final prototype toolset.		
QC data – allows user to select PS to QC, runs QCs, returns results in ArcMap document or in saved table location, as desired		StageStorage_QCResults and StageStorage_CreateExportTables – uses arcpy, adodbapi, and win32api modules
Generate Results - allows user to select PS to QC, runs volume calculations, returns results in ArcMap document or in saved table location, and updates results layers on map		StageStorage_VolumeResults and StageStorage_CreateExportTables - uses arcpy, adodbapi, and win32api modules

3.3 User Acceptance Testing

A major phase of the design and development lifecycle is to ensure appropriate and comprehensive testing of the toolset. Testing is iteratively conducted by the developer during the prototyping and development phases, and issues are resolved during those testing processes, but additional testing from end users is also required, called User Acceptance Testing (UAT). UAT is where the product is tested in a scenario that mimics how it will be used once deployed ("real world usage") and provides the following benefits:

- Ensures the tool functions as expected and as outlined in the requirements,
- Ensures procedures for using the tool and any associated help documentation actually illustrate how the tools should be used and confirm they are valid and accurate. By exposing end-users to this

information early in a development lifecycle, they become familiar with the tools early in the process, which can achieve reduction in training later in the final deployment stages, and,

- Finds issues and errors early in the development process will reduce the time required to resolve those issues if waiting until design completion to conduct testing.

For the stage storage tools, the following UAT procedures were implemented:

1. The toolset was deployed on a server environment where end-users would typically run the tool for the City of Suffolk project.
2. Two types of users were selected to test the tool:
    a. A GIS Analyst that is intimately familiar with the usage of tools from within the ArcMap environment and can provide feedback on those specific components of the design.
    b. A wastewater Engineer that is intimately familiar with the design and expected outcomes of the stage storage volume calculations that can provide feedback on those specific components of design.
3. A Tool help function was developed within the tool itself, where overarching information about each dialog was provided (Figure 15), as well as specific information about each parameter (Figure 16).
4. Tool test steps were developed (Appendix B), which provided a general overview of how to use the tool but did not provide a detailed 'step by step' narration so the user would have to rely on the information provided in the built-in tool help, therefore being able to identify any deficiencies or inaccuracies.
5. A test results tracking spreadsheet was also provided to the user to document any issues or provide feedback with each step of the test procedures (Figure 17). This document will be used by the developer to address any issues and then provide results back to the user for final verification testing of issue resolution.

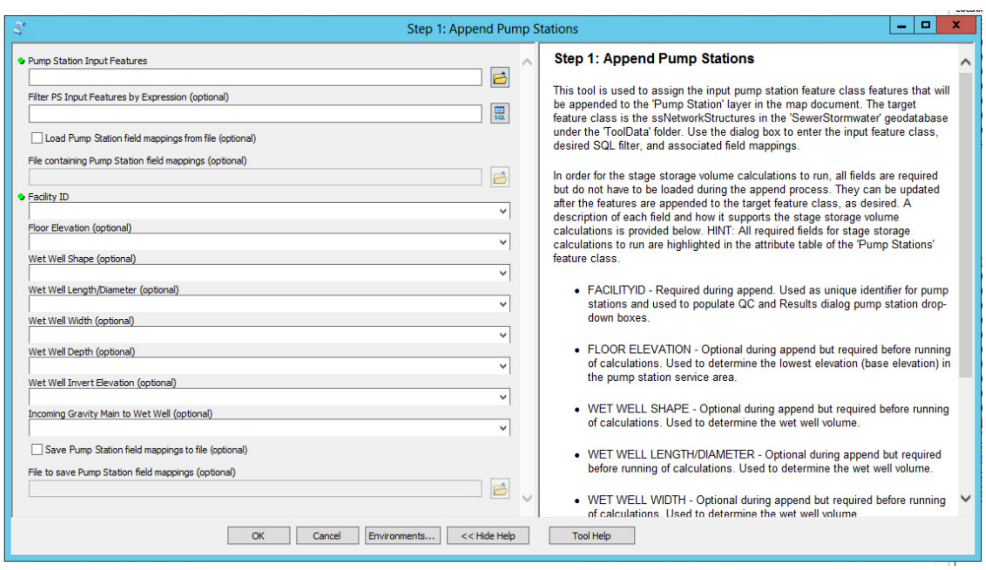


Figure 15. Tool help within tool documenting overall usage of each tool, Pump Station append tool example.

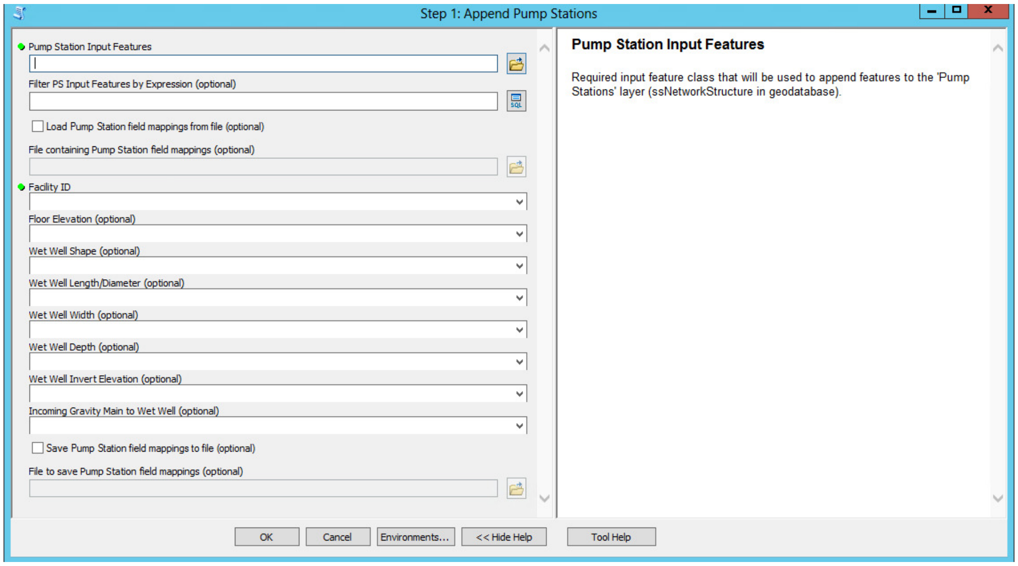


Figure 16. Tool help within tool documenting each parameter, Pump Station Input Features example.

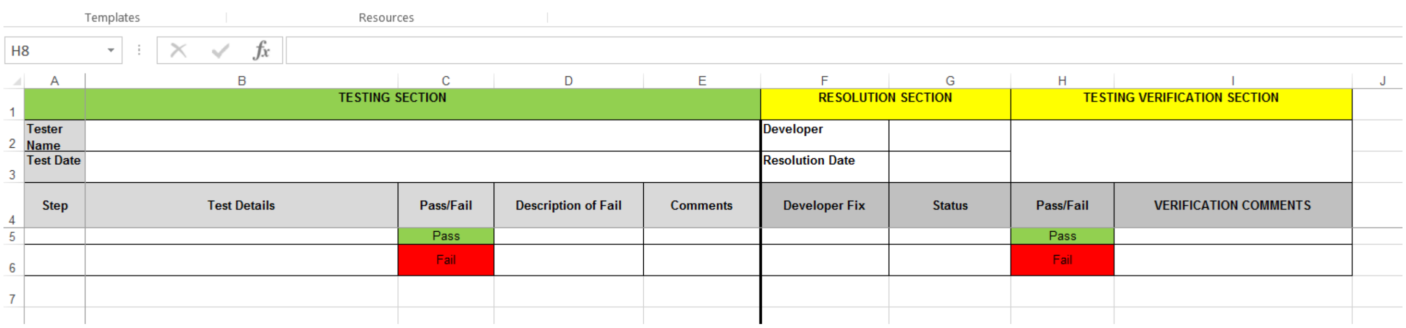
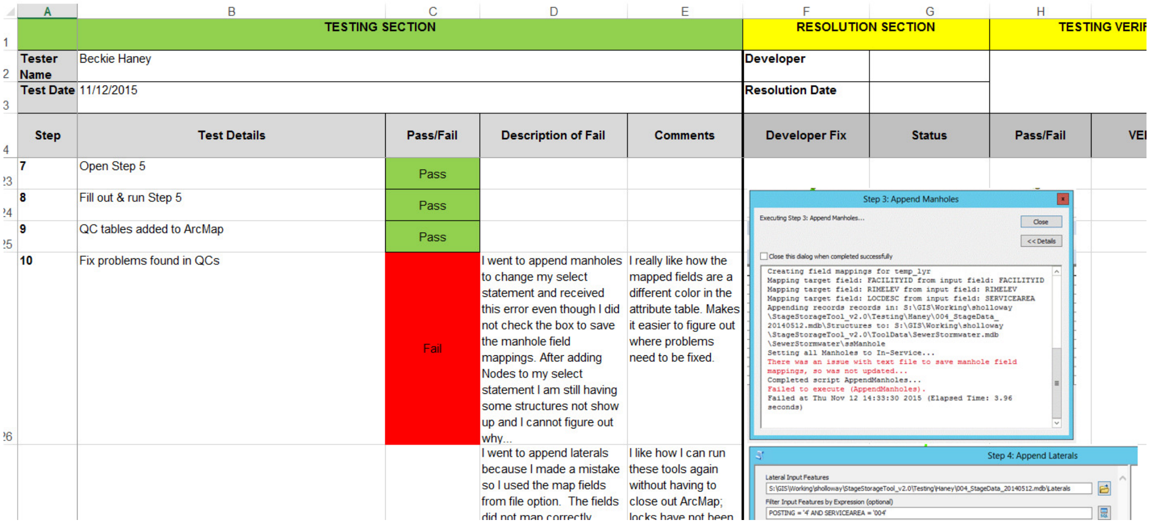


Figure 17. Test document provided to user to track issues/comments with testing.

3.3.1 Test Results

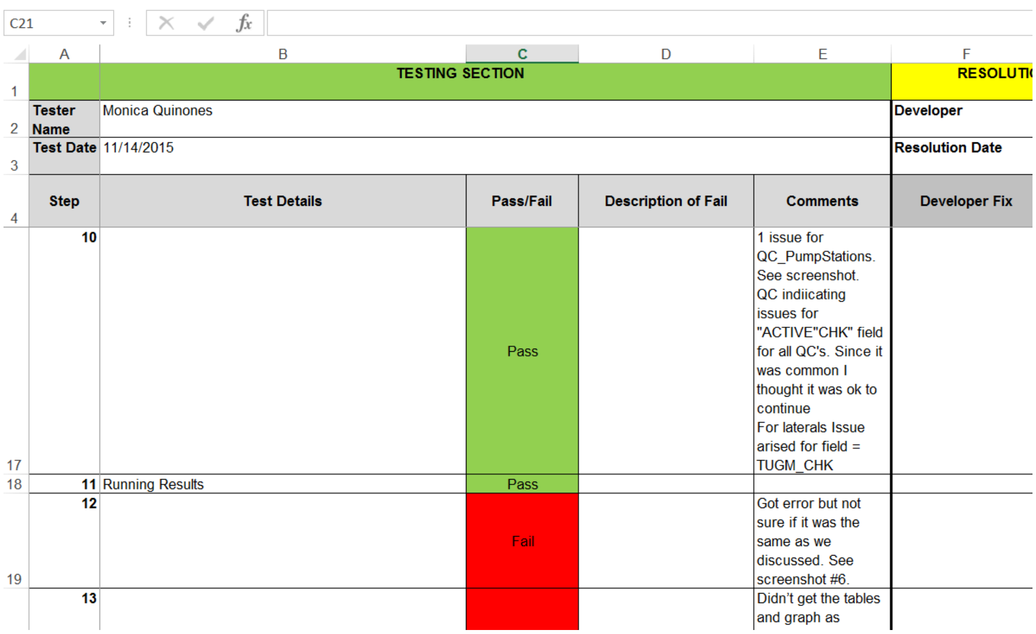
To date, the toolset has received one round of initial UAT. The testers walked through the procedures and recorded the results in the tracking spreadsheet, as shown in Figures 18 and 19. Overall, the tool worked as expected and functioned with no critical errors, but the following issues will need to be addressed:

1. Using the append tools having the fields marked as 'optional' but they are really required to run the tool is confusing to the end users, therefore most of the errors were really not having the correct data in the tool before attempting to run the 'QC' or 'Volume Results' scripts.
2. More detailed step by step procedures should be documented and provided in the help, directly in the tool to assist users in ensuring they are appending the data and interpreting the QC results accurately.
3. Users did not understand you could edit the data from within the source database once desired features and fields were appended, this needs to be better documented and examples provided.
4. A few errors messages were misleading, and did not truly cause a failure of the tools, so these need to be changed to informative messages for the user and not handled as errors.



	TESTING SECTION				RESOLUTION SECTION		TESTING VERIF
Tester Name	Beckie Haney				Developer		
Test Date	11/12/2015				Resolution Date		
Step	Test Details	Pass/Fail	Description of Fail	Comments	Developer Fix	Status	Pass/Fail
7	Open Step 5	Pass					
8	Fill out & run Step 5	Pass					
9	QC tables added to ArcMap	Pass					
10	Fix problems found in QCs	Fail	I went to append manholes to change my select statement and received this error even though I did not check the box to save the manhole field mappings. After adding Nodes to my select statement I am still having some structures not show up and I cannot figure out why...	I really like how the mapped fields are a different color in the attribute table. Makes it easier to figure out where problems need to be fixed.			
			I went to append laterals because I made a mistake so I used the map fields from file option. The fields did not map correctly	I like how I can run these tools again without having to close out ArcMap; locks have not been			

Figure 18. Test results for GIS Analyst.



	TESTING SECTION				RESOLUTI
Tester Name	Monica Quinones				Developer
Test Date	11/14/2015				Resolution Date
Step	Test Details	Pass/Fail	Description of Fail	Comments	Developer Fix
10		Pass		1 issue for QC_PumpStations. See screenshot. QC indiicating issues for "ACTIVE"CHK" field for all QC's. Since it was common I thought it was ok to continue For laterals Issue arised for field = TUGM_CHK	
11	Running Results	Pass			
12		Fail		Got error but not sure if it was the same as we discussed. See screenshot #6.	
13				Didn't get the tables and graph as	

Figure 19. Test Results for Wastewater Engineer.

4 RESULTS

The results of the design, development, and implementation of the toolset can be determined by illustrating the success of the UAT as well as the success of the technology that was selected. Additionally, the goal of the toolset is provide long-term viability and usage, so throughout the duration of the project, additional requirements were identified for future versions of the tool.

4.1.1 UAT Results

After the UAT results were received, it was determined that the major functionality of the toolset has been accomplished (as outlined in Section 3.1 and Appendix A), with some minor fixes needed on each tool. The test results are being reviewed, resolutions are being implemented, and feedback will be provided back to the users. They will use the feedback to retest the failed portions of the test results and confirm the issues have been resolved, as shown in Figure 20.

Step	Pass/Fail	Description of Fail	Comments	Developer Fix	Status	Pass/Fail	VERIFICATION COMMENTS
10	Pass		1 issue for QC_PumpStations. See screenshot. QC indiicating issues for "ACTIVE"CHK" field for all QC's. Since it was common I thought it was ok to continue For laterals Issue arised for field = TUGM_CHK	Use the 'Set all features as Active' option in the append dialogs, or you can edit the data directly and set all Active = 1 and all InActive = 0. See tool help on this field. Stage storage will only use active features.	Ready for retest		

Developer provide comments back to user of resolution and asks for retest verification. These iterations will continue until issue resolved.

Figure 20. Example of developer resolution comments back to user for additional testing.

The last phase of the design and development is to document and deploy the final toolset. Due to the nature of this project, that phase will require ultimate acceptance by the City of Suffolk, and possibly other localities that they would like to adopt and use this tool to develop stage storage volumes moving forward. At that time, a final deployment and usage of the tool will be planned and implemented with the organization.

4.1.2 Technology Selection

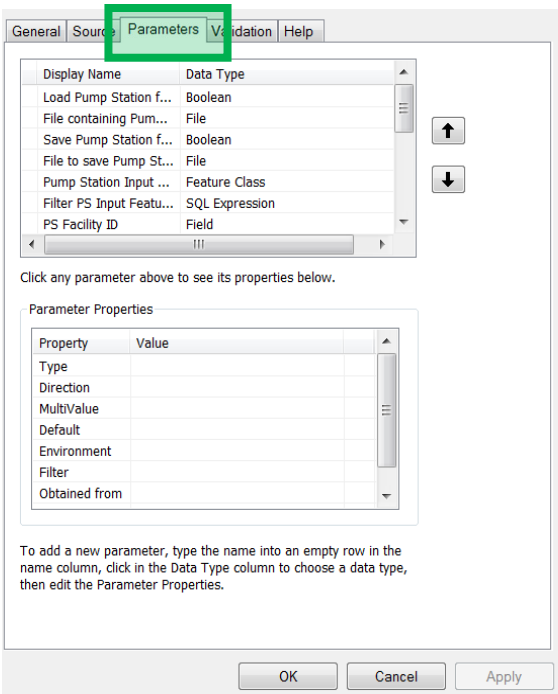
In addition to ensuring the functionality of the toolset would ultimately provide what was intended, the other goal of this project was to ensure the selected technology, ArcGIS ArcMap with a toolbox using python scripts, was a success and accomplished the following goals:

1. Had to be a platform that was accessible by most of the organizations in the Hampton Roads region, which would be the main focus for the tool distribution,
2. Had to be readily available as part of an out of the box (OTB) solution and would not require any additional components that would cause users to incur additional costs, and,
3. Had to be easily updated/customizable by normal GIS users, without the need for specialized training in advanced coding technologies, like C#, .NET, etc...

In order to accomplish these goals, some very critical components of the out of the box python scripts were leveraged:

- Python Parameters (Figure 21) – python parameters represent all of the parameters available to the user in each dialog box for input about how they want to run each step of the tool. These parameters have been exposed by ESRI for use in python scripts and are the same parameters they use in the ArcToolbox dialogs, therefore offering the same built-in functionality. The use of these parameters saves time and effort by eliminating the need to write custom code or build custom user interface dialogs.

- Tool Validator Class (Figure 22) – the tool validator class has also been exposed as a python script properties to allow the developer to write custom code for interacting with the user as they provide values in the dialog boxes to the parameters. For example, if the user selects the wrong type of file for storing the field mappings during append, this class can be used to send messages back to the user to resolve before the tool can proceed with execution.
- arcpy modules (Figure 23) – each tool used modules available, for free, for use with the out of the box arcpy scripting provided with ArcGIS Desktop, including:
    - arcpy – python site package that ships and is loaded with ArcGIS Desktop software.
    - win32api – python site package that extends functionality through python with Microsoft windows, including interaction with Microsoft Office products (free for download).
    - adodbapi – python site package that ships with Python 2.7 (10.1) and greater in ArcGIS Desktop and can be used to extend interaction with data sources, like Microsoft Access.



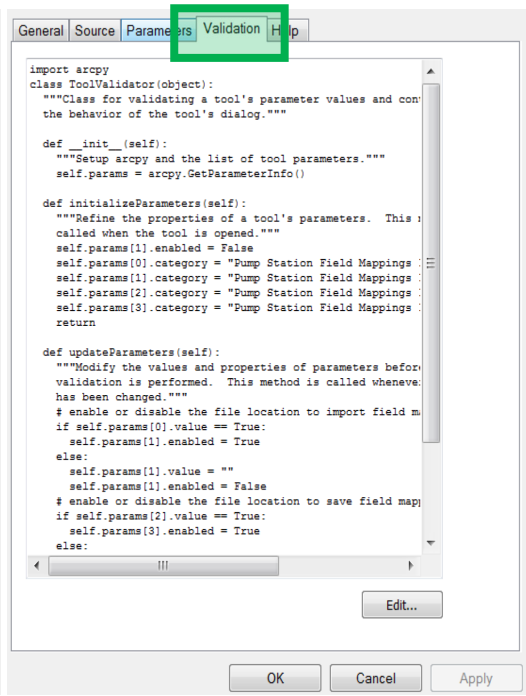


Figure 21. Parameters property of ESRI python script.

Figure 22. Tool Validator Class property of ESRI python script.

```
import arcpy, adodbapi, StageStorage_CreateExportTables
from arcpy import env
import win32api
from win32com.client import Dispatch
```

Figure 23. arcpy and arcpy modules available for free use.

4.1.3 Recommendations

At the onset of the project conception and throughout the project design and development, additional requirements were identified that could not be implemented during this project timeline but would be desired in future versions of the toolset. It is recommended the following requirements are reviewed and implemented, as necessary, in future versions of the toolset:

- Full migration of all stage storage calculations that currently reside in Microsoft Access to a python script that can referenced by the tools.
- Addition of tool to automatically re-create the LGIM geodatabase from an XML workspace document, if desired.
- Packaging of toolset in a manner that would allow for deployment on any environment, in any location, without any constraints on folder structure, etc.…

5 Conclusion

At the conclusion of this project, a toolset has been developed that meets the following functional requirements (as illustrated in Figure 24):

- Imports of a variety of data schemas into a standard geodatabase template for consistent usage in the toolset, specifically the ESRI Locality Government Information Model (LGIM) for wastewater.
- Integrates the data review and quality control processes with a spatial visualization component (through ESRI ArcMap interface) that provides the user with the ability to visualize which assets in the system would experience storage and/or surcharge.
- Provides the user with a single toolset that supports the import, calculation, visualization, and quality control procedures required to generate accurate stage storage volumes.

Continued resolution of issues found during testing, final testing, and final deployment (if adopted) will be the final stages of completion.

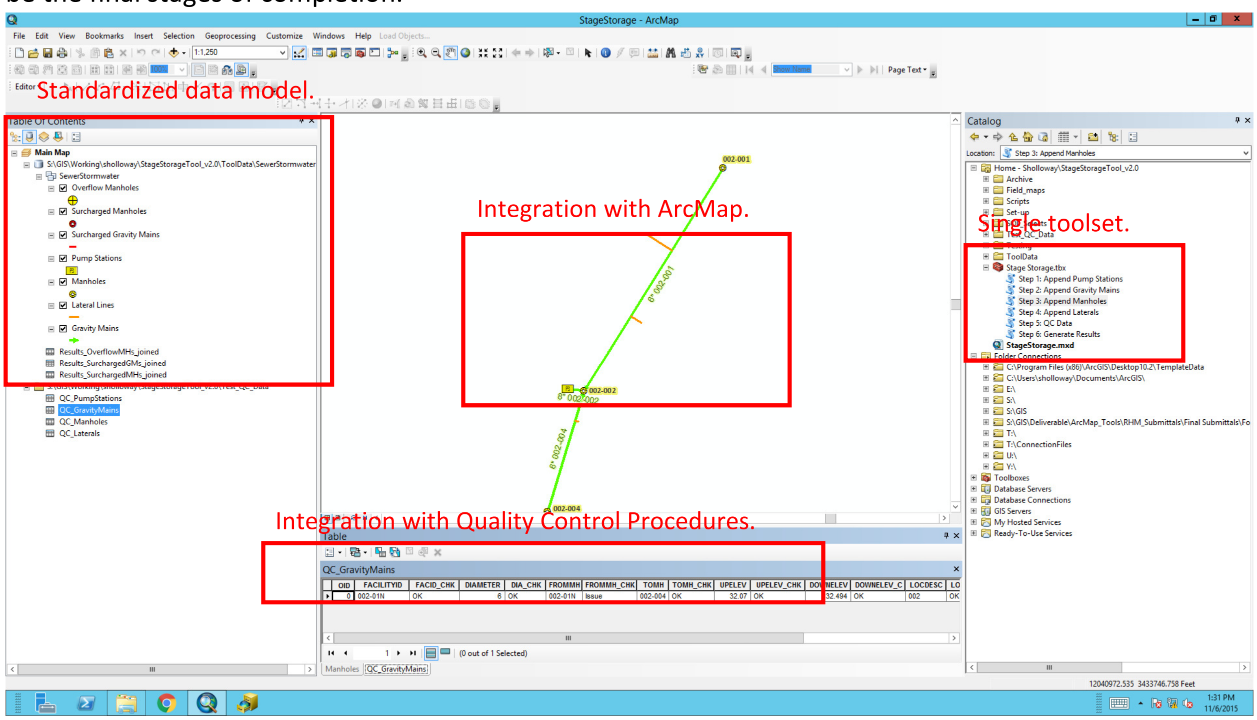


Figure 24. Final toolset.

6 REFERENCES

American Society of Civil Engineers, American Public Works Association, National Association of Clean Water Agencies, Water Environment Federation. (2010). Core Attributes of Effectively Managed Wastewater Collection Systems. Retrieved from www.wef.org/coreattributesofWWCS.

Bordo, Vince. (2015). Overview of User Acceptance Testing (UAT) for Business Analysts (BAs). Retrieved from https://www.develop.com/useracceptancetests.

Carr, R. (2001). Selection of sewer system modeling software products. Retrieved March 11, 2015, from http://www.apwa.net/Resources/Reporter/Articles/2001/2/Selection-of-sewer-system-modeling-software-products-.

CH2M HILL. (2009). *Program Report No. 13 – Flow Evaluation Report, Volume I*. Suffolk, Virginia.

Connecticut Department of Transportation. (2000). ConnDOT Drainage Manual, Stage-Storage Relationship (pp 10.7-1 – 10.7-9). Retried from http://www.ct.gov/dot/cwp/view.asp?a=3200&q=260116.

Djokic, D., Dartiguenave, C., Ye, Z. (2011). *Arc Hydro Tools Overview*. Retrieved from http://downloads.esri.com/blogs/hydro/AH2/Arc_Hydro_Tools_2_0_Overview.pdf.

ESRI. (2013). *ArcGIS Extensions*. Retrieved from http://www.esri.com/library/brochures/pdfs/arcgisextbro.pdf.

ESRI. (2014). *GIS Supports Sustainable and Effective Water Utility Practices*. Retrieved March 13, 2015, from http://www.esri.com/library/whitepapers/pdfs/gis-supports-sustainable-water.pdf.

ESRI. (2011). *GIS Data Quality Best Practices for Water, Wastewater, and Stormwater Utilities*. Retrieved March 13, 2015, from http://www.esri.com/library/whitepapers/pdfs/gis-data-quality-best-practices.pdf.

ESRI ArcGIS Resources. (1995-2013). *A quick tour of the ArcGIS 3D Analyst extension*. Retrieved March 12, 2015, from http://resources.arcgis.com/en/help/main/10.1/index.html#/A_quick_tour_of_the_ArcGIS_3D_Analyst_extension/00q800000003000000/.

ESRI ArcGIS Resources. (1995-2013). *Creating a python add-in application extension*. Retrieved March 12, 2015, from http://resources.arcgis.com/en/help/main/10.1/index.html#/application_extension/014p00000018000000/.

ESRI ArcGIS Resources. (1995-2013). *Developing with ArcGIS*. Retrieved March 12, 2015, from http://resources.arcgis.com/en/help/main/10.1/index.html#/Developing_with_ArcGIS/01w200000002000000/.

Grise, S. (2008). Local Government Data Model Implementation Guide. Retrieved from http://blogs.esri.com/esri/arcgis/2010/07/30/implementing-the-local-government-information-model-with-arcgis-10/.

Grise, S., Idolyantes, E., Brinton, E., Booth, B., Zeiler, M. (2001). *Water Utilities Data Model*. Retrieved Marcdh 13, 2015 from http://downloads.esri.com/support/datamodels/Water%20Utilities/ArcGIS_Water_Utilities.pdf.

IBM. (2015). *What is iterative development?* Retrieved November 12, 2015, from http://www.ibm.com/developerworks/rational/library/apr05/bittner-spence/.

International Business Machines Corporation. (2003). *Best practices for software development projects*. Retrieved March 11, 2015, from http://www.ibm.com/developerworks/websphere/library/techarticles/0306_perks/perks2.html.

Matrix Design Group, Inc. *Wastewater System Capacity Tool evaluates potential development impacts*. ESRI Proceedings. Retrieved from http://proceedings.esri.com/library/userconf/proc09/uc/papers/pap_1399.pdf.

Orchard, Hiltz, and McCliment, Inc. (2005). *Village of Dexter Sanitary Sewer Capacity Analysis*.

Regents of the University of Minnesota. (2011). Section 9: Pumping Systems. Retrieved from http://www.septic.umn.edu/prod/groups/cfans/@pub/@cfans/@ostp/documents/asset/cfans_asset_131295.pdf.

Schmidt, R. (2013). *Software Engineering: Architecture-driven Software Development*. Retried from http://irfree.com/ebooks/224753-software-engineering-architecture-driven-software-development.html.

United States Department of Transportation. (2001). *Hydraulic Engineering Circular No. 24, Highway Stormwater, Pump Station Design*. Washington, DC: Federal Highway Administration.

United States Environmental Protection Agency. (2012). *SSOAP Toolbox Enhancements and Case Study*. Cincinnati, Ohio: Office of Research and Development.

United States Environmental Protection Agency. (2007). *Computer Tools for Sanitary Sewer System Capacity Analysis and Planning*. Washington, DC: Office of Research and Development.

Unknown author. *Chapter 7 – Storage and Detention*. Retrieved from http://www.charmeck.org/stormwater/regulations/Documents/Storm%20Water%20Design%20Manual/Chapter7StorageDetention.pdf.

Appendix A – Functionality Matrix

Functionality Requirement #	Functionality Use Case Description	Additional Information	Python Parameter or Design Approach
1. Use of the most recent ESRI Local Government Information Model for sanitary sewer as the standardized geodatabase template	N/A	• Will require import of the following feature classes and fields from user's sanitary sewer GIS: ○ Pump Station: ▪ Asset ID, Text ▪ Floor Elevation, Number ▪ Wet Well Shape, Text ▪ Wet Well Width/Diameter, Number ▪ Wet Well Length, Number ▪ Wet Well Depth, Number ▪ Wet Well Invert Elevation, Number ▪ Incoming Gravity Main ID, Text ○ Manholes: ▪ Asset ID, Text ▪ Rim Elevation, Number ▪ Structure Type, Text ▪ Shape, Text ▪ Width/Diameter, Number ▪ Length, Number ○ Gravity Mains: ▪ Asset ID, Text ▪ Diameter, Number ▪ Upstream manhole ID, Text ▪ Downstream manhole ID,	• See 'DataConversion_Documentation.xlsx' for how the data will be stored using the LGIM for use with the stage storage tool. • Projection will be set for the packaged tool using the 'NAD_1983_StatePlane_Virginia_South_FIPS_4502_Feet' coordinate system. Tool can be set-up for any coordinate system. Refer to 'Tool set-up' documentation. • In order for the assets to relate back to a pump station, then will add or use the LGIM's "Location Description" field to the manholes and gravity mains that will have to include the Pump Station Service Area by Pump Station Facility ID.

Functionality Requirement #	Functionality Use Case Description	Additional Information	Python Parameter or Design Approach
		Text ▪ Upstream Invert Elevation, Number ▪ Downstream Invert Elevation, Number o Laterals: ▪ Asset ID, Text ▪ Gravity Main ID, Text • Projection information has to be stored in feet (horizontal and vertical). • Laterals is currently not used in the tool and will be incorporated as part of this project. Need to count the number of laterals per gravity main and then assume a volume for the laterals as currently done for sewer accounts by gravity main. • If the users Asset IDs do not relate to the pump stations, have to figure out a method for determining the correct assets to include in analysis. May have to use a geometric network for this to work.	
2a. Tool for user to import their sanitary sewer GIS data into the standardized geodatabase template and save import configurations for later use	User will open an interface that will allow them to locate and select the feature classes they want to import per type of feature. The interface will also allow them to map the correct fields to use during import, for example, a field called 'Diam' in the incoming	• If gravity mains do not store the upstream and downstream manhole IDs then tool needs to allow the user the choice to generate these values during the import from the connectivity. • If pump station does not store incoming gravity main ID, then tool needs to allow the user the choice to generate this value during import from the connectivity. • If laterals does not store gravity main ID, then tool needs to allow the user the choice to generate this value during import from the	• Will use "Field Mappings" python parameter for the imports • Will use a combination of "Boolean" and "File" python parameters to indicate if user wants to save the import field mapping, which can be saved as a text file. • Will use a combination of "Boolean" and "File" python parameters to indicate if user wants to import field mappings instead of building new. • Will use "Booleans" to allow the user to decide to:

Functionality Requirement #	Functionality Use Case Description	Additional Information	Python Parameter or Design Approach
	gravity mains feature class may become the final field called 'Diameter' in the template gravity mains feature class. The interface will then allow them to trigger the import and will provide feedback if an error occurs. Additionally, the interface will allow the user to save their import configuration for future use.	connectivity. • In order to save the configuration, will also have to provide a means to import the configuration file if they want to use it again later. Configuration files should be saved by feature class type (i.e. pump station, gravity main). • If the fields are not the same data type, then the tool needs to allow the user the choice to convert the field data types during import. • Allow user to assume all manhole diameters of 48", if desired. • If data already exists in the template geodatabase, allow the user to overwrite existing features. • User can import entire system or just specific service areas.	○ Determine/store upstream and downstream IDs during import ○ Determine/store incoming Gravity Main ID during import ○ Assume all manhole diameters as 48" • Will delete existing features by default • For the gravity main ID in laterals and incoming gravity ID in pump stations data generation, will have to use a spatial selection and grab the gravity main that is at the 'ToNode' of each lateral and grab the gravity main id that snapped to the pump station at the 'ToNode', respectively
2b. Tool for user to run quality control (QC) checks on data after import	Once the data is successfully imported without errors, the next step is for the user to run some quality control checks on the data to ensure it is valid and accurate for use in the stage storage volume calculations. User will open an interface that will provide them some preconfigured QC results that the user can use to address the issues either directly in the imported data or by fixing the data and	• User can edit the data directly in the template database and refresh the quality control checks. • Minimum QC checks will include: ○ Valid values in all the required fields listed in Requirement 1 ○ Valid connectivity (i.e. no missing manholes) ○ Negative slope pipes	• Will use "Table" python parameter to store location of QC tables • Run python code to access QC queries set up in Access and return those as tables ○ Ask user if wants to load results in ArcMap ○ Run network checks to list out connectivity issues

Functionality Requirement #	Functionality Use Case Description	Additional Information	Python Parameter or Design Approach
	reimporting.		

2c. Tool for user to run stage storage volumes, view results in tabular and graphic format, and export results to Microsoft Excel	Once all data is imported and QC'd, user will open an interface that allows for the generation of stage storage volumes at various stage elevation changes in the pump station wet well. Once the storage volume calculations are complete, the results will display in the interface in tabular and graphic format. Additionally, the interface will allow the user to export their results to Microsoft Excel.	• Logic from existing tool will be used and no engineering calculations will be modified.
		• QC of storage volume calculations will be achieved by comparing the volumes using the new tool to the volumes using the old tool – they should be equivalent of logic from old tool was migrated correctly.
		• Changes in stage elevations in wet well will range from 0.1 – 1.0 feet, in 0.1 foot increments.
		• Tabular format will be a table showing:
		  o Stage
		  o Stage Elevation
		  o Wet Well Volume
		  o Gravity Main Volume
		  o Ungula Volume
		  o Manhole Volume
		  o Total Volume
		• Graphic format will be a line graph showing the stage elevation (feet) against the volume of the wet well (cubic feet).
		• Export to Excel will include tabular results only.
			• Will use "Boolean" python parameter to let user decide what results to generate, including:
			  o Table of volumes
			  o Graph of stage versus volumes
			  o Export to Excel
			• Will use the "String" python parameter to store the name of the graph and the table
			• Will use "Folder" and "String" python parameters to allow user to save location and name of file for Excel to export
			• Use python to return the table and graph to the ArcMap document as results
			• Will use pyodbc or other python modules (win32.com/Dispatch) to access queries and run results in Microsoft Access stage storage database and return as table to use for table and graph creation

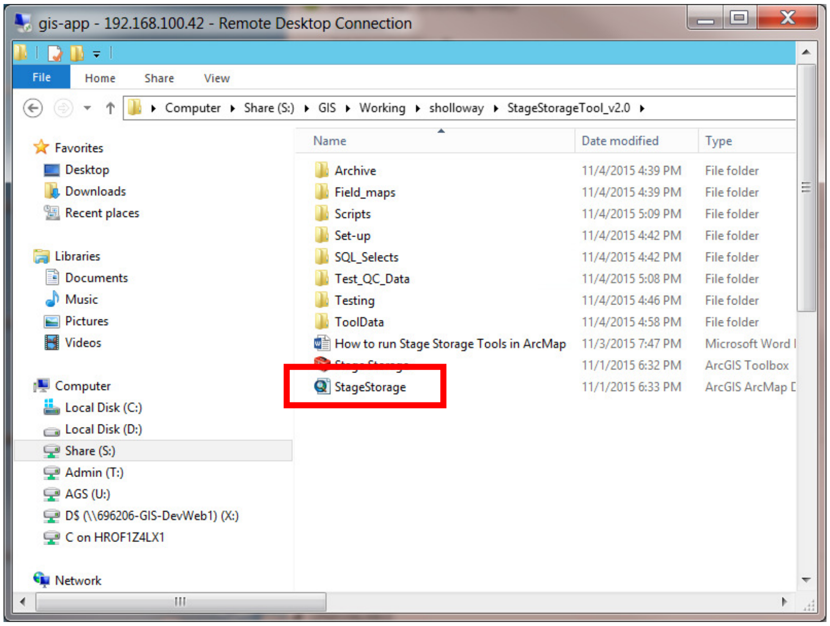
3a. Visual display and linkage to imported sanitary sewer information in standardized geodatabase template in ArcMap	User will access the imported sanitary data, geodatabase, and toolset through a standardized template in ArcMap. The map document will provide layers that are already symbolized with a standard convention.	• Tool should verify that the appropriate layers from the standard geodatabase are present and warn users if they are not before the tool can run. • Layers in geodatabase include: Pump Station, Manholes, Gravity Mains, and Laterals. • Other layers can be added or removed as desired by user.	• Use the arcpy.mapping and other arcpy modules as necessary to check the appropriate layers and database connection prior to importing any new data
3b. Visual display and linkage to stage storage	Once results are generated, the map will	• Results should include (at overflow stage): o Highlighting of pipes/manholes that	• Will create symbolized results layers ahead of time that will be refreshed using arcpy.mapping and

Functionality Requirement #	Functionality Use Case Description	Additional Information	Python Parameter or Design Approach
results in ArcMap	automatically update with the results of the stage elevation where the overflow occurs. The map document will provide layers that are already symbolized with a standard convention.	contain sewage. o Highlighting of overflow manhole. o Label at pump station indicating volume of sewage in the wet well and wet well level. o Labels on assets that contain the volume of asset, volume of sewage in asset, and % full. • Tool should allow user to add results table into ArcMap document, if desired.	other modules as necessary
4. Development of help documentation for end users	User will have help documentation that provides the necessary information for set-up and usage of the geodatabase, toolset, and ArcMap template.	• Need to include information on how to define projection appropriately. • Help documentation should include: o System requirements o Background information o Set-up and installation of toolset o Step-by-step instructions on how to use tool and generate results o Any 'Tips and Tricks' for tool usage (like not removing required layers, etc...)	

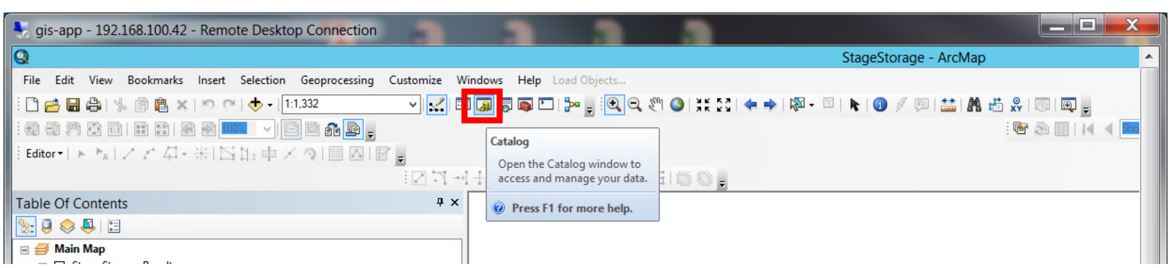
APPENDIX B – TEST STEPS DOCUMENT

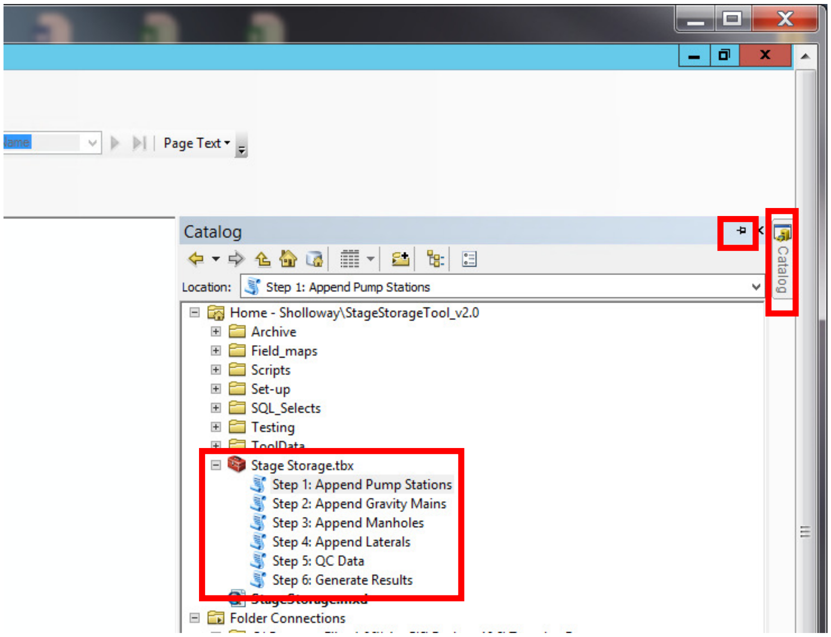
How to run Stage Storage Tools in ArcMap

1. Log-onto Suffolk Rackspace GIS server.
2. Tool is located at S:\GIS\Working\sholloway\StageStorageTool_v2.0
3. Open the StageStorage.mxd shown below

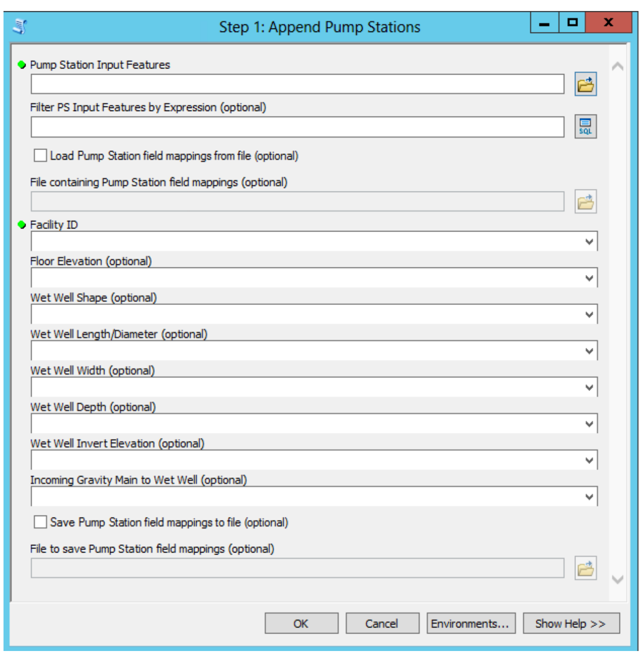


4. When mxd opens, open the ArcCatalog window by selecting the button on the standard toolbar as shown below. Select the Catalog tab on the right of the window and navigate to the Stage Storage Toolbox (Stage Storage.tbx) and double-click. You can 'pin' the Catalog window to the right by selecting the pushpin icon.

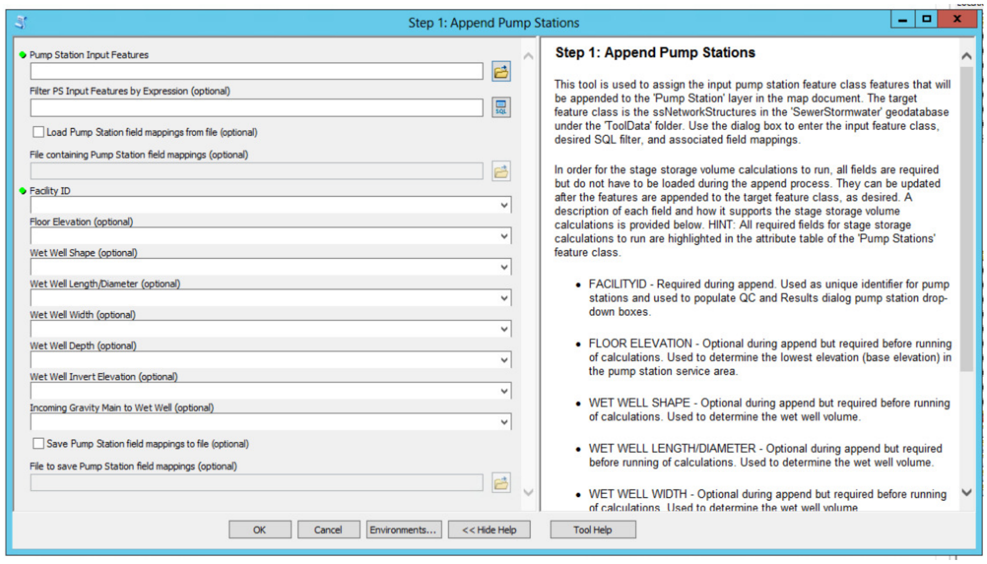




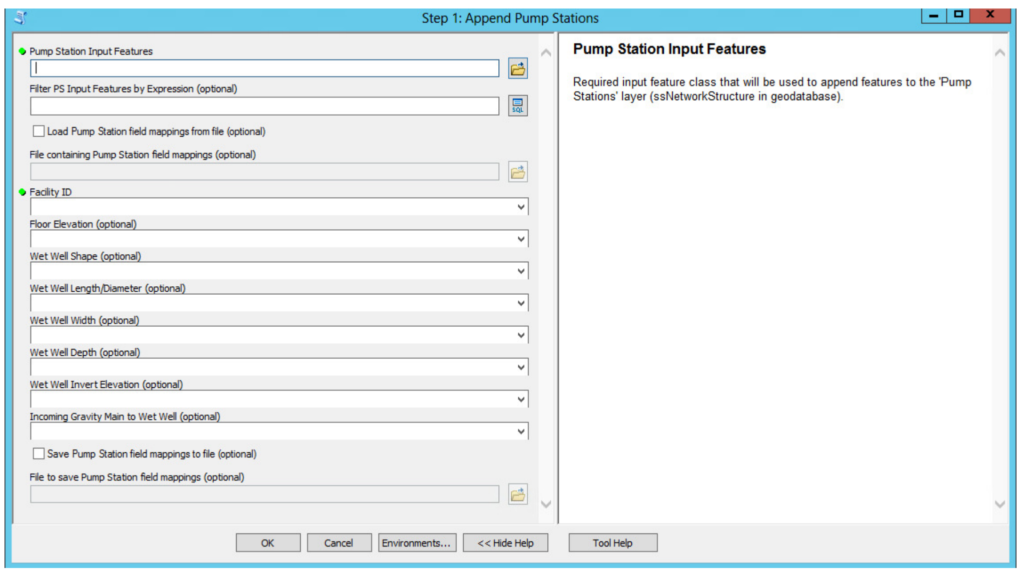
5. Now you are ready to run the tools!
    a. Step 1 – 4 will append the required features into the SewerStormwater.mdb that is located in the ToolData folder. This is the database that the stage storage tool links to.
    b. Step 5 will run QCs on the appended data and return results of features with issues
    c. Step 6 will run the total volume calculations and will return a table, graph, and will update the 'Stage Storage Results' layers in the table of contents
6. To append all the required data:
    a. Pump Stations – run Step 1: Append Pump Stations by double-clicking the script icon and the python dialog will open below



b. The dialog box should open as shown below.



i. To determine what is needed in each parameter, just place the cursor in the parameter and the help window will update for that parameter.



ii. To see overall help for the tool again, just click anywhere in the grey area on the dialog window.

iii. To close the Help, just select 'Hide Help'. To show Help again, just select 'Show Help'.

c. Select the input features that will populate the pump station feature class by navigating to the GIS database already provided here: S:\GIS\Working\sholloway\StageStorageTool_v2.0\Testing.

i. Note: Data from previously run volumes has been provided in order to be able to compare the results from this tool to the original tool. The only difference is that the laterals have been added to the geodatabases from the previous runs because this tool now calculates lateral volumes based off of the laterals in the GIS. So some difference in volumes is expected, but should equal the total lateral volume for that station.

1. Haney will be testing PS 004 data that was run on 05/12/2014 which is located in the geodatabase at S:\GIS\Working\sholloway\StageStorageTool_v2.0\Testing\Haney\004_StageData_20140512.mdb
2. Quinones will be testing PS 120 data that was run on 04/02/2015 which is located in the geodatabase at S:\GIS\Working\sholloway\StageStorageTool_v2.0\Testing\Quinones\120_Stage_Data_20150402.mdb

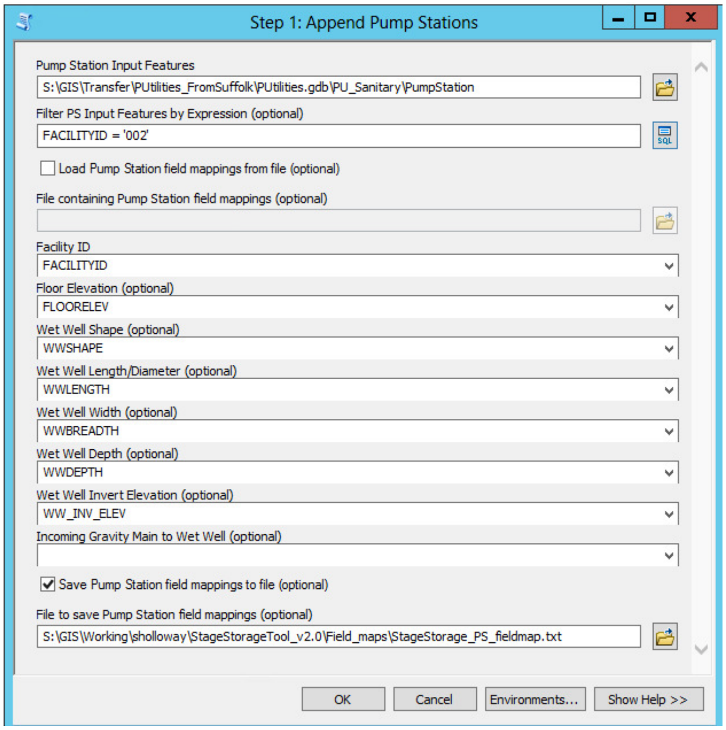
ii. In order to bring in only the Pump Station you want to run, you can then enter a filter in the 'Filter PS Input Features by Expression' parameter. An example selection has been provided in the 'SQL_Selects' folder called 'PumpStation_Select.exp' that can be loaded and changed for the desired pump station.

iii. The first time running the tool, field mappings will not be available, so map each field as shown in the figure below. If desired, save the field mappings once they are complete to a desired location for later use.

1. Note: Field maps can be used over and over again for any pump station as long as the same feature class with the same schema is selected for the input features.
2. Note: The text file to be saved must already exist. So create a blank text file before pointing to it in tool.

iv. An example of a completed append dialog is shown below.

1. Note: For Suffolk, Incoming Gravity Main to the Wet Well is not available in the Pump Station feature class, so that field is not mapped and left blank. There are two options for this field:
    a. Can update the values once appended into the geodatabase (option that will be selected for this process).
    b. Can create a field in the incoming GIS feature class that stores this data.

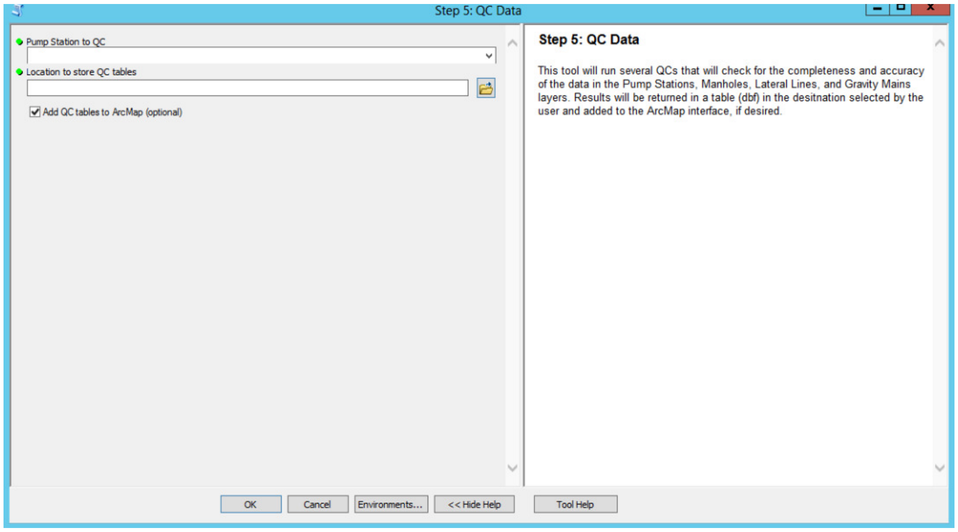


d. Finish all the remaining Step 2 – 4 using the same procedures as outlined for pump stations above. When all appends complete, should be able to zoom to all layers and see your data in the map. Figure below shows example once all appends complete for PS 002.

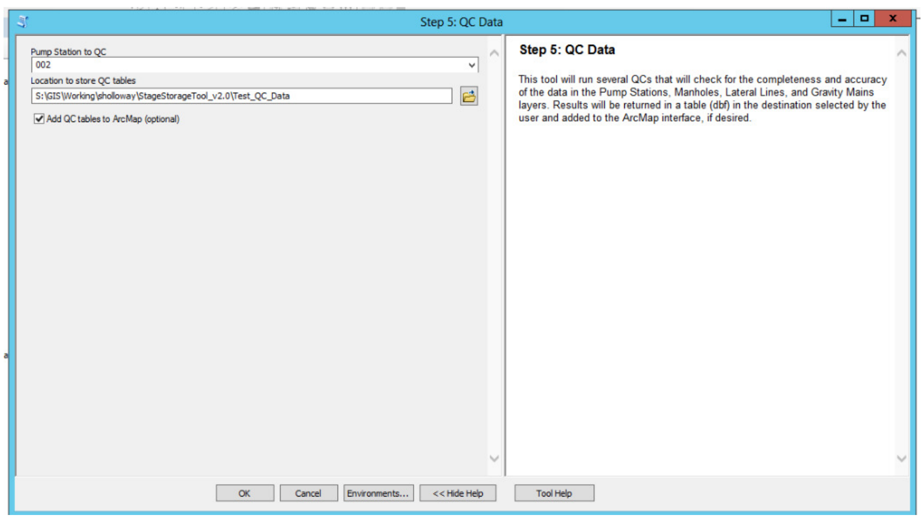
   i. HINT: Remember to filter data coming in!! There are sample select expressions in the 'SQL_Select' folder that can be adjusted for desired pump station

   ii. Note: The gravity mains, manholes, and laterals have an 'In-Service' option. If a field cannot be mapped to update that from the GIS, you can just select the option to set all gravity mains to In-Service.

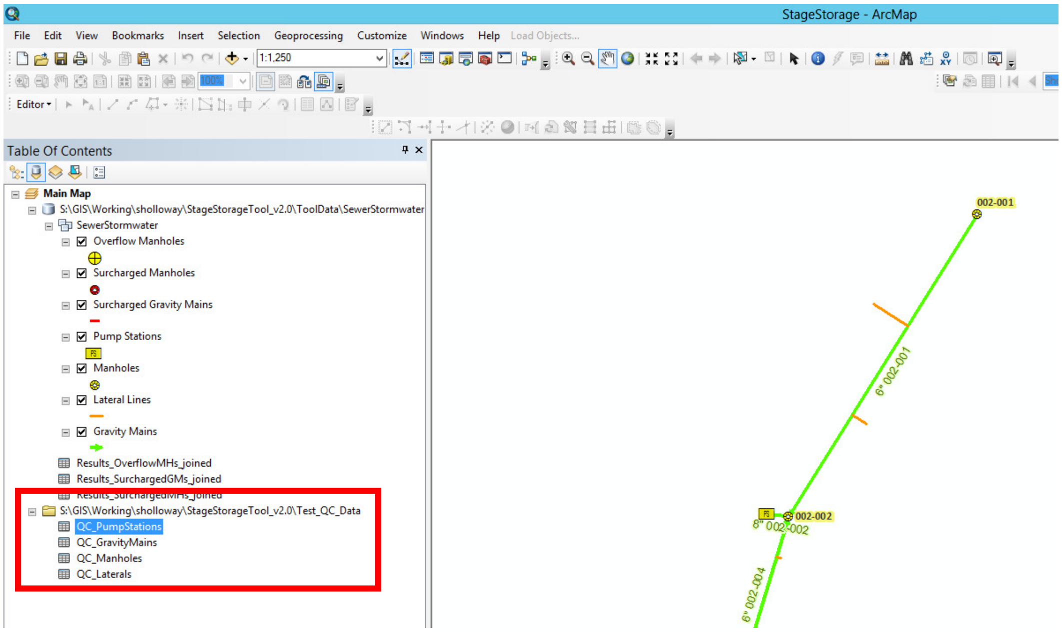
7. Once all data is appended, it's time to run the QCs to ensure the data is complete and valid before running the volume calculations. To run the QCs, just double-click Step 5: QC Data script and the dialog below should appear.



8. Run the QC by selecting the desired pump station, location to store output tables, and if tables should be added directly to ArcMap as shown below. Click on each parameter or grey area in dialog to update help information as desired.



9. Once QC run, tables will show up in 'Source' tab of table of contents as shown below (if that option was selected) or can be added by the 'Add Data' button.

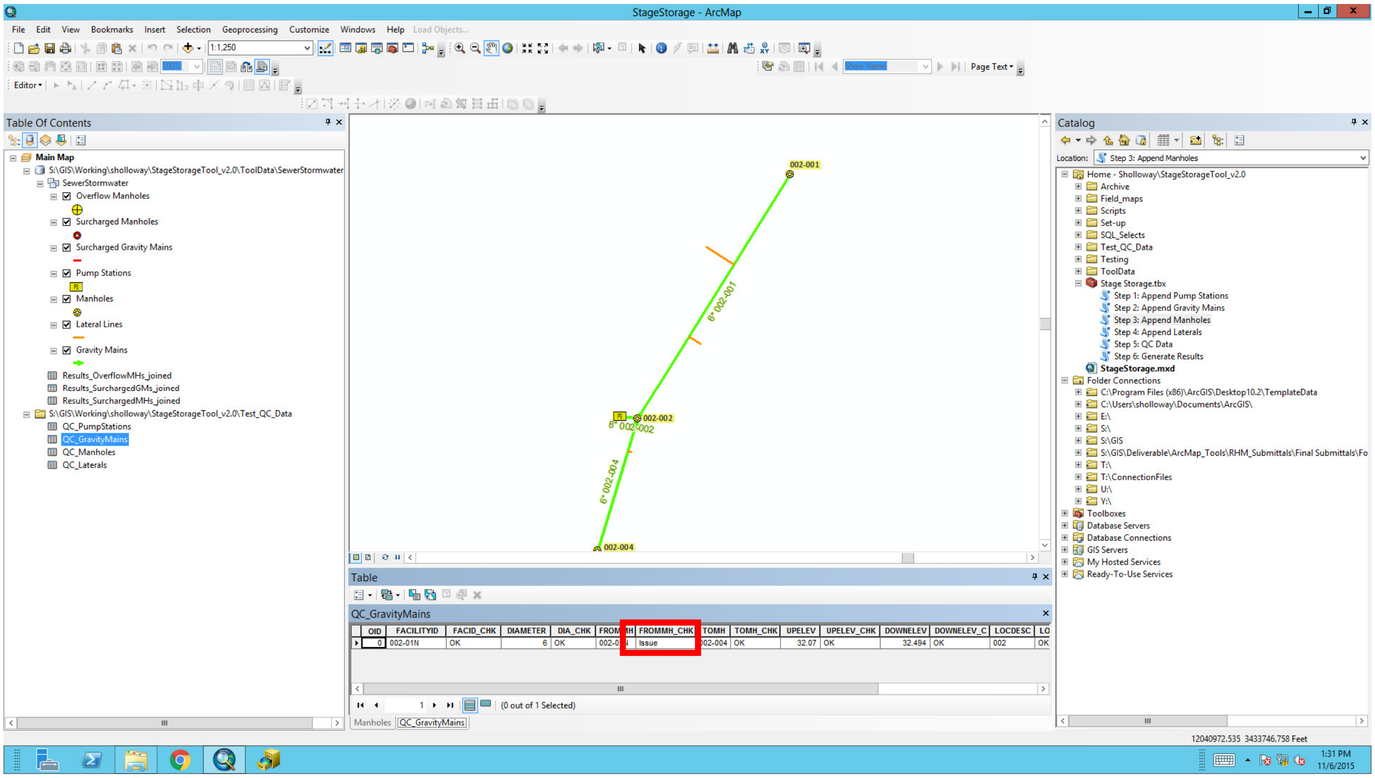


10. Right-click on each table and view attributes (see example below). Only failing records for each feature class will show up. If the field is good, it will say "OK" in the fieldname_CHK column. If it has an issue, it will say "Issue". Most of the issues will be missing data and/or connectivity issues. To address issues, you can do the following:

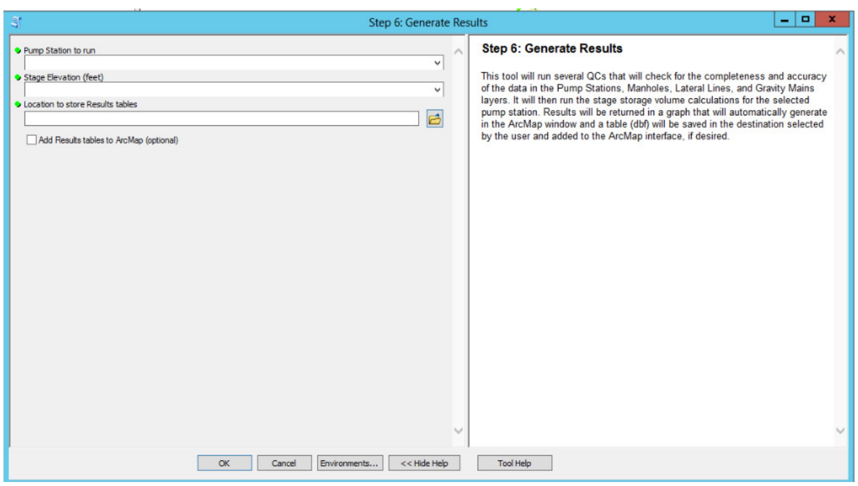
    a. For missing values, you can edit and/or calculate the values directly in the stage storage geodatabase or you can fix your datasource and reappend the features, then rerun the QCs.

    b. For connectivity issues, you can edit or fix issues directly in the stage storage geodatabase or you can fix your data source and re-append the features, then rerun the QCs.

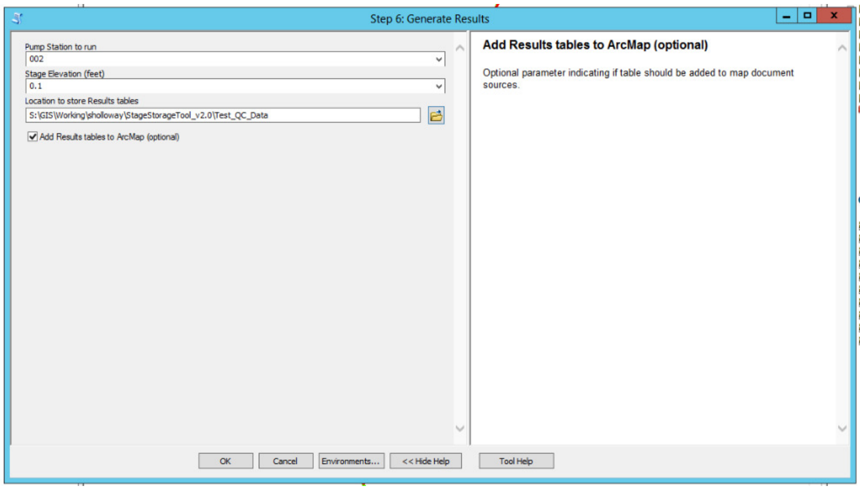
        i. HINT: Sometimes the connectivity issues is just bringing in the correct types of features and simple adjustment to the select query used during the append will resolve any issues. For example, in the QC_GravityMains shown below, there was an issue with the FromMH_CHK, which when I went back to look, I did not correctly append that manhole because it had a type of node. Fixed select statement, reappended, and the issue went away.



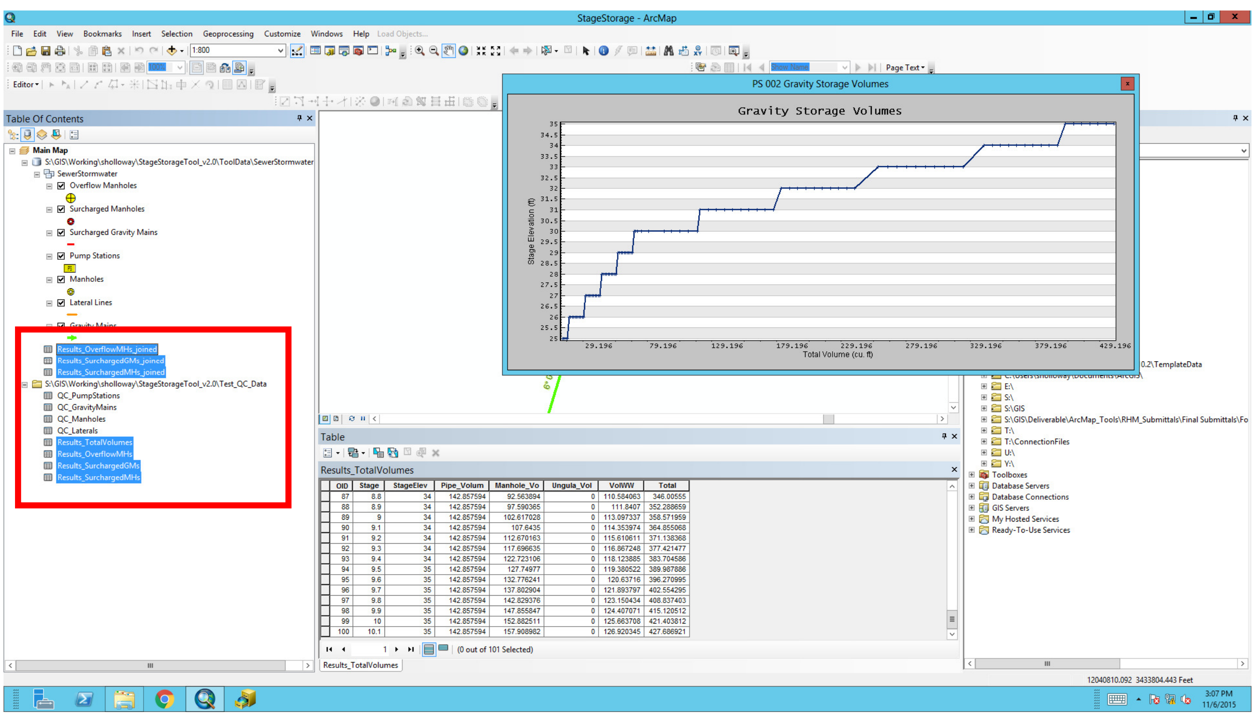
11. Once all QC's come back clear, it's time to run results! To run the results, just double-click ;Step 6: Generate Results' script and the dialog below should appear



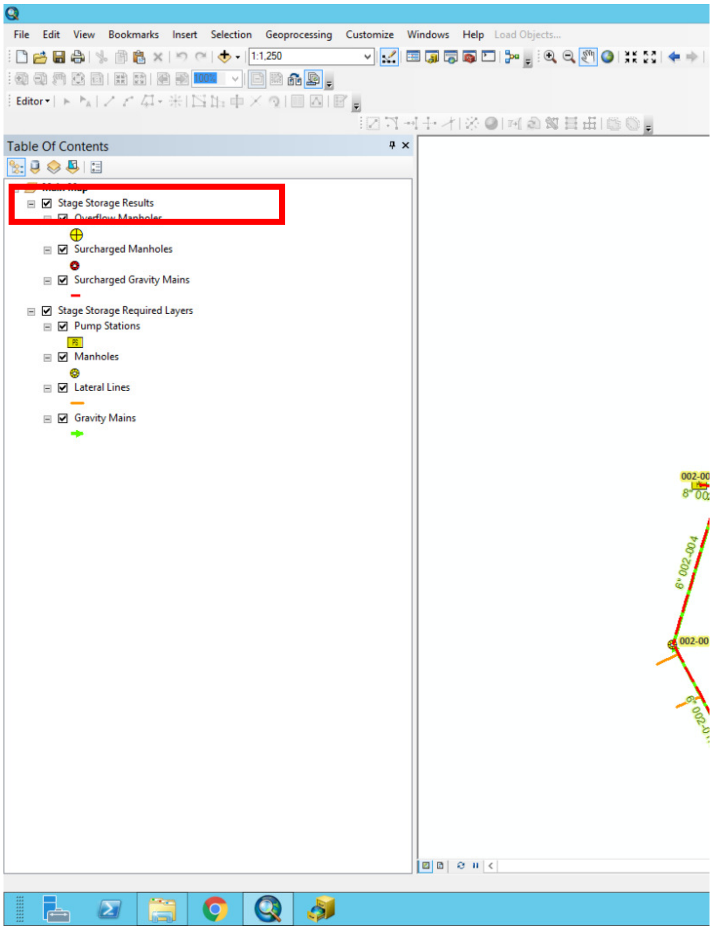
12. Run the results by selecting the desired pump station, location to store output tables, and if tables should be added directly to ArcMap as shown below. Click on each parameter or grey area in dialog to update help information as desired.



13. As calculations are running, a dialog will indicate the progress. When complete, the graph, Results_TotalVolumes, Results_OverflowMHs, Results_SurchargedMHs, Results_SurchargedGMs tables will automatically add to the map document. As well, the joined tables that update the 'Stage Storage Results' group will also be automatically updated. This is so you can see the results of the current run easily but save old results too, if desired.



14. To view results on map, just turn on the group layer as shown below (turned off by default).



15. If desired, you can right-click on the graph and export results to desired format. You can also copy and paste results volumes into Excel if desired.

Prerequisites:

- ArcGIS Desktop Advanced 10.2 or higher
- Python 2.7 or higher (installed in C:\Python27\ArcGIS10.2 folder)
- Win32com client and adodbapi that are installed in C:\Python27\ArcGIS10.2\Lib\site-packages
- Microsoft Access 2003 or higher