Maryland Department of the Environment

QAPP:

MARYLAND POINT SOURCE DATA

Project ID:

Effective Date: 2.14.2023

EPA Document Control Number (DCN):

1800 Washington Blvd.
Baltimore, MD 21203

A PROGRAM MANAGEMENT

A1. Approval Sheet

Concurrence

Preparer/Author

Name: Shannon McKenrick Title: Project Lead Organization: Maryland Dept. of the Environment, WSA	Signature: Shannon McKenrick Date: 2/6/2024
Name: Title: Organization:	Signature: Date:

Management/Supervisor

Name: Matthew Stover Title: Division Chief, WQS&A Organization: Maryland Dept. of the Environment, WSA	Signature: Date: 2/6/2024
Name: Title: Organization:	Signature: Date:

Quality Assurance Officer

Name: Najma Khokhar Title: Quality Assurance Coordinator Organization: Maryland Dept. of the Environment, WSA	Signature: Najma Khokhar Date: 2/6/2024

EPA Region 3

Name: Title: R3 Designated Project Manager Organization:	Signature: Date:

Approval

EPA Region 3

Name: Durga Ghosh Title: R3 Delegated Approving Official Organization: CBP / USGS	Signature: Date:

Note: This approval action represents EPA's determination that the document(s) under review comply with applicable requirements of the EPA Region 3 Quality Management Plan [https://www.epa.gov/sites/production/files/2020-06/documents/r3qmp-final-r3-signatures-2020.pdf] and other applicable requirements in EPA quality regulations and policies [https://www.epa.gov/quality]. This approval action does **not** represent EPA's verification of the accuracy or completeness of document(s) under review, and is **not** intended to constitute EPA direction of work by contractors, grantees or subgrantees, or other non-EPA parties.

Revision History

This table shows changes to this controlled document over time. The most recent version is presented in the top row of the table. Previous versions of the document are maintained by Quality Manager.

Document Control Number	History/ Changes	Effective Date
	Version 2 (submitted to CBP on 2/6/2024) Updated Cover and Signatory Pages, added Revision History	2/14/2023
	Version 2 (submitted to CBP on 11/30/2023) Updates to the entire QAPP document to reflect EPA R5 QAPP Guidance	2/14/2023
	Version 1	01/01/2019

Table of Contents

List of Figures

List of Tables

A. Project Management and Information/Data Quality Objectives

A-1. Project Purpose, Problem Definition, and Background

A-1.1. Project Purpose and Problem Definition

The purpose of this Quality Assurance Project Plan (QAPP) is to provide documentation of the Maryland Department of the Environment's (MDE) submission of Point Source (PS) data for the Chesapeake Bay Model. In particular, the QAPP aims to document inventories, data sources, QA/QC protocols, data location, and other aspects of point source data management related to Maryland's annual Point Source Data Submission (PSDS) for the Chesapeake Bay Model. It includes descriptive information about the method for obtaining data for PS facilities such as monthly treated effluent, Biosolids application data, and permitting information for significant facilities.

This QAPP presents the comprehensive data quality control procedures used to identify and correct mistakes with the PSDS such as duplicates, missing data, inaccuracies in the data, and unit errors. These quality control procedures allow the verification and/or/ correction of facility-reported values on an as-needed basis. This serves as an important tool for ensuring PS data submission integrity. Likewise, this QAPP is important to its users (MDE staff, outside researchers, etc.) as it provides the necessary information for accessing, retrieving, and analyzing PS data for Maryland.

Data monitoring and reporting requirements for all Maryland facilities are outlined in individual permits, which are issued by the State of Maryland. This gives Maryland the authority to enforce the requirements of these permits. When significant errors or suspicious data are identified in the facility-reported data, these are reported to MDE's Compliance Program and Permitting Programs to either confirm or correct these values for future data submissions. If there is refusal to comply, it is MDE's permitting programs that have the authority to determine compliance actions.

The charge of maintaining and reporting the State Point Source data for the Chesapeake Bay Model rests with MDE's Watershed Protection, Restoration, and Planning Program (WPRPP) within the Water and Science Administration (WSA), with assistance provided by the Compliance Program and Wastewater Pollution Prevention & Reclamation Program within MDE. WPRPP obtains the Point Source data from several online databases, checks the data for completeness, assesses the data for accuracy, and formats the point source data for annual submission. Sections B-2 and B-4 of the QAPP document the method used to query data and collate the final Point Source Data Submission.

A-1.2. Problem Background

The 1987 Chesapeake Bay Agreement was reached jointly by the Chesapeake Bay Commission, the US-EPA, the District of Columbia, and the States of Delaware, Virginia, Maryland, and Pennsylvania to restore and protect the Chesapeake Bay. This agreement was renewed in 2014. Maryland and other Bay states utilize the Chesapeake Bay Model to plan implementation towards the goals set forth in the Chesapeake Bay Agreement and the Chesapeake Bay Total Maximum Daily Load (TMDL). The Chesapeake Bay Model is updated annually with Point and Nonpoint Source data that is submitted by the states to the Chesapeake Bay Program Office (CBPO). This data is then utilized in the model to develop an Annual Progress scenario that Maryland uses to gauge its progress toward meeting TMDL planning goals in the various modeled sectors.

Through its authority as the regulatory agency for individual wastewater permits, Maryland Department of the Environment requires most facilities to monitor and submit Discharge Monitoring Reports (DMR) monthly. This monthly monitoring data is reported into databases for compliance purposes, where compliance programs may choose to edit the data or make changes due to compliance actions or to ensure data completeness. Descriptions of the data availability through various reporting and compliance databases and the flow of data during the project are described in detail in sections A-1.2.1 through A-1.2.4.

A-1.2.1 - Network Discharge Monitoring Reports (NetDMR) and Integrated Compliance Information System (ICIS) Database

The Integrated Compliance Information System (ICIS) is a data management system that supports the National Pollutant Discharge Elimination System (NPDES) regulations for wastewater. The data stored in ICIS originates from the Network Discharge Monitoring Reports (NetDMR) database. Permitted facilities remotely enter their monitoring data into NetDMR. Spray Irrigation and Rapid Infiltration data is also entered into NetDMR and likewise is available in the ICIS database.

ICIS can be used for tracking permit compliance and enforcement status for the NPDES. ICIS is also available to all federally registered users to review and extract data reports. NetDMR holds downloadable Portable Document Format (PDF) copies of original facilities' print monitoring sheets used for calculations. Both of these are accessed frequently during the Treated Effluent and Irrigation & Infiltration data management protocols described in sections B-2 and B-4. Both databases are used in the functions described in this QAPP.

A.-1.2.2 - Tools for Environmental Management and Protection Organizations (TEMPO) Database

In January of each year, biosolids reports are received annually by MDE's Land Management Administration/Resource Management Program (RMP). This due date and frequency are required by their permits. RMP hand-enters this data into their TEMPO (Tools for Environmental Management and Protection Organizations) database. Once data entry is completed, project staff can access TEMPO to download the sewage sludge reports needed for the PSDS project.

A.-1.2.3 - Point Source Tool

The Point Source Tool (PST) is an online data management system maintained by the CBPO. The PST houses wastewater data extracted from Enforcement and Compliance History Online (ECHO), which in turn receives data from ICIS. In addition, it is used by State submitters to QA/QC data. The submitter views the data and follows a step-by-step analysis of the data made by the tool. Through a series of "checks," the State submitter can use automated corrections for data flagged by the PST. These corrections are based on historical averages. The State submitter may also enter values researched in ICIS or NetDMR. The final products of the work described in this QAPP are Microsoft Excel .xlsx files that are submitted electronically to a staging area maintained by the CBPO via the PST.

A-2. Project Task Description

Due to the nature of the timing of the data revisions in each database described in section A-1.2, there may be data quality issues associated with using only one source of available DMR data. In order for MDE to compile all available Point Source data, this project involves querying the relevant monthly DMR data for the entire model progress year (State Fiscal year - July-June annually) from the databases described in section A-1.2, comparing the available datasets, making data quality decisions and relevant edits to ensure completeness and accuracy, and formatting the data for submission to CBPO. The final product for this work is six .xlsx files which contain monthly effluent data for each facility by type. The final data product includes flow (in MGD) and nutrient and sediment concentrations in mg/L. Biosolids data is provided in countywide totals of tons applied by calendar year.

There are three components to the annual PSDS: 1) Municipal Wastewater Treatment Plants and Industrial Facilities, 2) Spray Irrigation Facilities and 3) Biosolids Applicators. The methods for querying and formatting the data for the annual submission are provided in sections B-2 and B-4 and Appendix A.

A-2.1 - Municipal Wastewater Treatment Plants and Industrial Facilities

The State of Maryland has more than 1,300 municipal and industrial facilities regulated under a NPDES permit. About a quarter of these facilities are municipal wastewater treatment plants, while the rest are industrial facilities. Of these regulated dischargers, 78 facilities are Significant Municipal facilities, with an engineered flow greater than 0.5 MGD. There are only 7 industrial facilities that are significant (with a flow >0.5 MGD). Maryland's PS data reporting and evaluation includes these 85 significant surface discharge municipal and industrial plants which contribute significant nutrient point source pollution in Maryland and the Chesapeake Bay.

Currently, monitoring data from a total of 233 municipal wastewater treatment plants (Significant and Insignificant Municipals) and 7 Significant industrial plants are included in our submission of data collected by Waste Water Treatment Plants (WWTP) (Section B-2). Results are submitted from monitoring at the active outfalls for each facility. Facilities and outfalls reported as no longer active are designated as inactive in the PST. Active outfalls that report no flow in a month report a value of zero (0) for all contaminants in that month. Inventory of treatment facilities is maintained in the PST.

Treated Effluent data for more than 1,000 Insignificant Industrial facilities is prepared in a separate spreadsheet by MDE's Industrial & General Permits Division (IGPD). These values are calculated from design parameters outlined in the facility's NPDES permits. This data is submitted annually to the PST.

A-2.2 - Spray Irrigation Facilities

Spray irrigation refers to nutrient laden wastewater applied to agricultural land, turf grass, or woodland. Rapid infiltration is defined as wastewater discharged to groundwater via rapid infiltration seepage pits. Both types of systems have permitted limits and designs to reduce impact on surface waters.

A list of spray irrigation operators is reviewed annually and their status is verified with MDE's WSA Wastewater Pollution Prevention & Reclamation Program. Spray irrigation operators are identified as Agricultural and Non-agricultural. Non-agricultural operators apply to turf grass and woodlands. If a facility is no longer operating, it is removed from the inventory. All data for Spray Irrigation and Rapid Infiltration facilities is extracted from ICIS. However, no print monitoring reports for these facilities are maintained in NetDMR, limiting the QA/QC process and requiring more substitution of default/average values.

For both of these data sources, permit status can assist in determining whether the facility remains active in the inventory. Coordination with MDE's Compliance Program and Wastewater Pollution Prevention & Reclamation Program may verify or negate the status of the facility. If necessary, facility permits may be accessed online at the: Maryland Wastewater Permits Interactive Search Portal: http://mes-mde.mde.state.md.us/WastewaterPermitPortal/

A-2.3 - Biosolids Application

Biosolids are a result of wastewater treatment that have been processed to remove liquids and harmful pathogens. Maryland issues permits to sites/locations for land application of biosolids (aka – processed sewage sludge from WWTPs) as agricultural fertilizer. Biosolids are tracked by application site, by county, and WWTP source. Although approximately 250 sites are permitted in MD in any given year, it is expected that biosolids are applied to about half of these permitted sites. There is also a holding lagoon in Cedarville, MD, where biosolids are over-wintered from multiple WWTPs. Currently, the over-wintered biosolids are transported to application sites in Maryland and Virginia counties. Each data line of Maryland's submission identifies the total load of applied biosolids (in tons/year) by county.

A-3. Information/Data Quality objectives and performance/acceptance criteria

The work described in this QAPP differs from most environmental information projects in that MDE and project staff are not involved in the generation of any data. Since data is being collected and formatted from various databases, this project assumes that measures of data quality are incorporated into the validation procedures each database utilizes when users enter data. There are two data quality objectives that this work aims to achieve: completeness and accuracy.

The primary objective of this project is to obtain a complete dataset that contains values for each month of the fiscal year for every facility. As there are differing intervals in the data flow between databases described in section A-1.2, as well as different locations where data is stored depending on the facility type, no one tool contains a complete set of Point Source data for the year. Due to this, the work that addresses the completeness objective is accomplished by querying data from the various sources described in section A-1.2 and reconciling those data to fill in fields that are missing data from one source but have values in another. There are also instances where no values are found in any data source (and where a 0 value is not appropriate); in these cases, default values based on permit data are substituted.

Once a complete dataset has been created, the secondary objective for data quality is to assess the data for accuracy. Several data accuracy issues are commonly present - these include errors with calculated fields (such as Total Nitrogen and Total Phosphorus), issues with incorrect units, and others. In addition, in the process of reconciling multiple datasets, it is possible that two disparate values are available for the same field. A series of data quality decisions must be made to assure the accuracy of the values in the final submission. Project staff choose the appropriate values and utilize a variety of quality checks to assure the data is as accurate as possible. The process used to assess the accuracy of the data is described in section B-4.

A-4. QAPP Distribution list

The QAPP and all attachments (including appendices, figures, and references) shall be distributed to the following individuals and organizations:

Table 1. Maryland's Point Source Data Submission QAPP Distribution List.

Name	Project Title	Organization
Shannon McKenrick	Project Manager	Analysis and Informatics Section Water Quality Standards and Analysis Division Maryland Department of the Environment
Bel Martinez da Matta	Technical Lead	Analysis and Informatics Section Water Quality Standards and Analysis Division Maryland Department of the Environment
Najma Khokhar	Quality Assurance Manager	Analysis and Informatics Section Water Quality Standards and Analysis Division Maryland Department of the Environment
Matthew Rowe	Assistant Director - MDE WSA QAPP Approval Authority	Water and Science Administration Maryland Department of the Environment
Ruth Cassilly	Nonpoint Source Policy Analyst	University of Maryland Chesapeake Bay Program Office

Copies of the QAPP and all Appendices and other attachments will be maintained in digital and print within the Maryland Department of the Environment. Details on the retention of records and document revision are in section A-9.

A-5. Project Organization

The work detailed in this QAPP is to be completed by the Analysis and Informatics Section of the Watershed Protection, Restoration and Planning Program (WPRPP) of Maryland Department of the Environment's Water and Science Administration (WSA). MDE does not employ any contractors, subcontractors, or sub-grantees to complete this work. While the Analysis and Informatics Section performs the formatting and assessment work detailed in this QAPP, assistance is provided by the Wastewater Pollution Prevention & Reclamation Program of WSA. The Wastewater Pollution Prevention & Reclamation Program provides context and training for individuals within WPRPP performing the assessment activities described in this document. Additional details about the training project staff received from other MDE programs relating to the completion of the Point Source Data Submission work is in section A-8.

Matthew Rowe, the Assistant Director of the Water and Science Administration (WSA), is the executive and approval authority for this Quality Assurance Project Plan. An organization chart of the relevant individuals and their reporting relationships within MDE's WPRPP is presented in Section A-7.

The Project Manager for the Point Source Data Submission (PSDS) is Shannon McKenrick. The Project Manager is responsible for developing and maintaining the QAPP, serving as liaison with EPA and other programs within MDE, and discussing any issues and results with senior staff. The Project Manager also assists with the final assessment of the data submission and participates in any data editing decisions. The Technical Lead for this work is Bel Martinez da Matta, Statistician in WPRPP's Analysis and Informatics Section. The Technical Lead provides additional support in developing content in the QAPP, maintaining internal SOPs, and conducting most of the technical work described in sections B-2 and B-4.

The project QAM for this work is Najma Khokhar. The QAM provides feedback on the QAPP, assistance with the final product data management and storage, and assures the QAPP meets EPA and MDE standards. Any issues with data viability, quality assurance methods, or documentation are communicated directly to the project manager and/or senior management as needed through e-mail or in-person communications.

The final data product for the PSDS project is utilized by EPA CBP in order to provide annual data to the Chesapeake Bay Model. Other users of the data include WPRPP's Total Maximum Daily Load development program, which utilizes point source data in water quality modeling. The final annualized data that is delivered to the CBP is maintained in an internal network location that others within WPRPP can access for other water quality compliance or modeling projects.

A-6. Project QAM independence

The QAM for the work described in this QAPP is not involved in the processing of the data for submission. They are contacted for assistance by the Project Manager or Technical Lead and are responsible for providing feedback on the Quality Assurance objectives and needs of EPA and MDE. They provide assistance with the management of the final data product and maintaining internal copies of the QAPP. The QAM does not otherwise use or engage with the data product in any way that would present a conflict with the project.

A-7. Project Organizational Chart and Communications

MDE's Analysis and Informatics Section has the primary responsibility for the QAPP and the work described in sections B-2 and B-4. An organization chart showing the reporting relationships and lines of communication is presented in **Figure 1**.

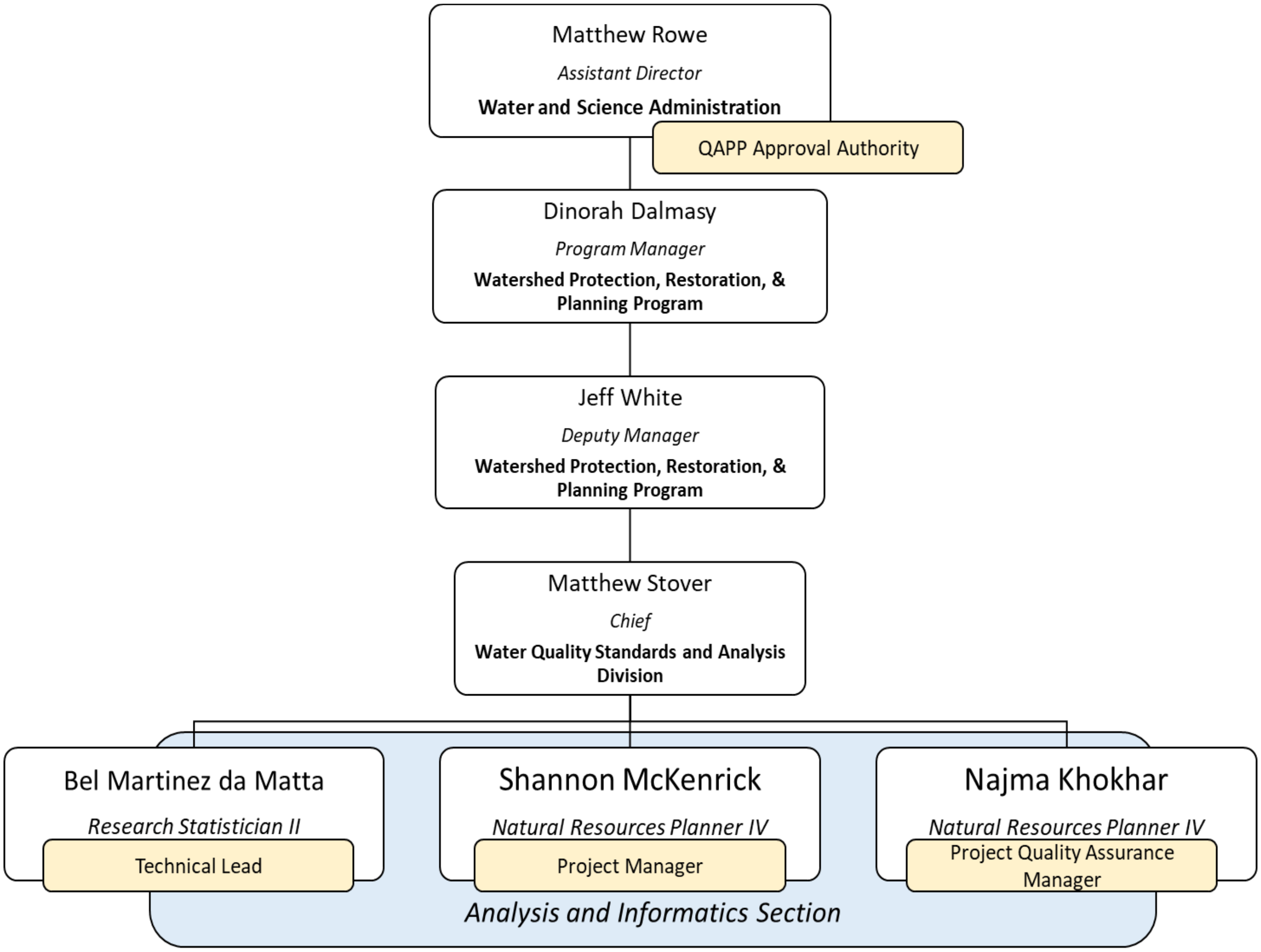

Figure 1. Organization Chart of QAPP Project Staff.

QAPP responsibilities are highlighted in yellow; reporting relationships are represented by lines linking each individual.

The Project Manager and Technical Lead communicate through in-person meetings and emails as necessary during all stages of the project. The Project Manager and the Technical Lead also work simultaneously while developing and revising the QAPP as needed and during the assessment activities described in the QAPP. The Project Manager and Technical Lead are both included in all communications with other programs in WSA and with managers, data users, EPA and senior management regarding the project. Any issues related to the PSDS data, its QAPP, or any other Point Source related data needs are related to managers and data users as needed through e-mails or in-person communication. Senior management is notified of issues with any aspect of the project (including the QAPP) at management's discretion.

If the Project Manager or Technical Lead determines that changes to the QAPP are necessary due to updates in the process or other document requirements, the QAM and WSA management will be notified annually by April 30. Draft QAPP changes will be made available for internal review by managers, the QAM and other internal data managers by July 30, to facilitate a timely submission of QAPP materials to EPA before September 1. Communications between WSA staff and from the Project Manager or Technical Lead to EPA will be accomplished through email.

If issues arise in QAPP development or the methodology described in sections B-2 and B-4, project staff notify managers through e-mail or in-person communication. Managing staff will either make decisions on how to proceed, or choose to elevate issues to senior management or other coordinating WSA programs as needed. If there are issues encountered with the development of the QAPP which require EPA assistance or feedback, the Project Manager and/or Technical Lead will communicate these issues to CBPO QAPP liaison Ruth Cassilly through email, copying the QAM, relevant managers, and/or senior leadership.

During the process of compiling data for the PSDS, project staff may identify issues with data availability or completeness associated with permit circumstances. Once these issues are identified, the relevant WSA permitting staff are contacted by the Project Manager or Technical Lead through email. The points of contact for WSAs permitting staff are provided in **Table 2**.

Table 2. Contact Information for MDE Point Source Personnel.

Name	Title	Division	Program	Data Content
Arno Laud	Chief	Enforcement Division	Compliance Program	Significant Municipal compliance issues
Yen-Der Cheng	Chief	Municipal Surface Discharge Division	Wastewater Pollution Prevention & Reclamation Program	Municipal permit activity
Wallid Saffouri	Program Administrator		Engineering and Capital Projects Program	Municipal WWTP Upgrade Timelines
Thomas Yoo	Chief	Biosolids Division	Land & Materials Administration - Resource Management Program	Biosolids application, Cedarville Lagoon Monitoring Data
Mary Dela Onyemaechi	Chief	Groundwater Discharge Permits Division	Wastewater Pollution Prevention & Reclamation Program	Spray Irrigation Facilities
Jonathan Rice	Chief	Industrial and General Permits Division	Wastewater Pollution Prevention & Reclamation Program	Industrial facilities

A-8. Personnel training and certification

There are no required training or certifications required for the work described in this QAPP. Working knowledge of the databases, data submission format, and method explanations were attained by project staff through the previous version of this QAPP and in-person communications with its author. Project staff were provided training on the work described in this QAPP through shadowing opportunities with staff who completed the PSDS in prior years. Additionally, WPRPP has a number of individuals with institutional knowledge that are able to provide additional context and understanding of the organization's quality standards and policies to project staff.

Project staff received informal on-the-job training facilitated by individuals within the Wastewater Pollution Prevention & Reclamation Program. Regular informal training sessions with staff from other MDE programs are scheduled as needed to provide ongoing support and address emerging needs as the project progresses. Informal training is provided through in-person meetings and communications as well as email correspondence.

A-9. Documents and Records

Once the data reconciliation and quality assessment is completed and all Point Source data is submitted, files of the submitted data (including the exported and intervening data files, the file provided by IGPD, and the file used to upload the Significant Municipal corrections) will all be stored on an MDE server. Additional process files that contain information regarding the source of each value, comments and flags pertaining to quality issues, and a data mismatch log are all maintained on the same MDE internal server *(\\mde-nas\departments\WSA\Point Source Information\PointSource_EDS*). Copies of these files are available to the QAM and may be referenced by internal data users and/or project staff when accessing the final data submission.

The final dataset - a complete fiscal year Point Source dataset - is also available in the Point Source Tool. Data users may utilize the PST to access annual point source treated effluent data as an alternative to data stored on a network location. The fiscal year Point Source data available both internally through MDE's network and externally within the PST will match.

The QAPP document and all related materials (appendices, attachments and referenced files) will be maintained in two locations on an MDE internal server: one in an internal MDE network location and another in a cloud storage folder via Google Drive where QAM maintains all Maryland Department of the Environment QAPP documentation. Additionally, print copies of the QAPP and its attachments are kept at the Project Manager and QAM's discretion at MDE's Baltimore Office.

B. Implementing Environmental Information Operations

B-1. Identification of Project Environmental Information Operations

The environmental operations conducted for the point source data consist of compilation of data submitted to other programs within MDE's Water and Science Administration (WSA) and the Land and Materials Administration (LMA), as well as exports from online databases such as CBPO's PST, ICIS, and TEMPO. Using multiple data sources allows MDE to compare data for completeness and accuracy so that the final data product(s) are representative of the annual Point Source data for the State.

The information compiled from databases and other sources is processed in R for data accuracy checks and calculation of parameters. The next section provides details of data acquisition for each set of point source data.

B-2. Methods for Environmental Information Acquisition

The annual submission of point source data consists of different categories of data: treated effluent, irrigation and infiltration, and biosolids application. Treated effluent data includes data from significant and insignificant municipal wastewater treatment plants and industrial facilities. Not all types of water facilities that generate effluent need to be reported. Groundwater Dischargers, Combined Sewer Operations, and Drinking Water Treatment Plants are the three types of plants that are not included in the model and therefore not included in submissions to the CBPO.

Both municipal and industrial facilities are categorized as significant if their engineered flow is greater than 0.5 MGD, and insignificant if their engineered flow is less than 0.5 MGD. Given the variability of effluent nutrient content and how industrial facilities are categorized within the WPPRP, engineered flow is used instead of nutrient loads in pounds per year as a threshold for significant versus insignificant facilities. This allows the PSDS to align with MDE's internal facility classifications and creates a dataset that users can utilize for other modeling projects within WPRPP. The significant municipal facilities are further categorized as major (engineered flow above 1 MGD) or minor (engineered flow is between 0.5 and 1 MGD).

The frequency data compiled and assessed varies depending on the type of data. Considering the greater contribution of significant municipal facilities to overall nutrient loads, those data are assessed quarterly, with a lag time of about two months (e.g., first-quarter data is compiled and assessed in mid-May) due to the facility's reporting and database update schedule. The remainder of the data is compiled and assessed annually. Except for biosolids application, all point source data annually submitted to the CBPO via the PST corresponds to the previous state fiscal year. Biosolids data is reported on a calendar year basis and submitted annually to MDE. As a result, the data submitted in the PST corresponds to the previous calendar year.

Municipal and significant industrial facilities electronically submit their effluent data from their Monthly Operating Reports (MORs) as Discharge Monitoring Reports (DMRs) into the NetDMR database monthly. The MORs contain facility-generated laboratory results for multiple constituent water quality analyses of effluent samples collected at each plant's active outfall. Requirements on effluent monitoring, analysis, and data reporting are detailed in facilities' discharge permits issued by the Wastewater Pollution Prevention & Reclamation Program (WPPRP), Municipal and Industrial & General Permits Divisions. These divisions receive and review permit applications regarding the design and construction of municipal wastewater and industrial facilities discharging effluents to Maryland's waterways, as well as entities applying spray irrigation and rapid infiltration. They also verify design capacity (flow) and monitoring of potential contaminants. MDE WSA's Compliance Program is responsible for enforcing the requirements of the permits, which includes reporting and remote data entry into NetDMR.

The data entered in NetDMR is transferred to ICIS at least nightly. MDE's Compliance Program's Enforcement Division will manually enter corrections into ICIS as needed for permit compliance. The data in ICIS is transferred daily to US-EPA's ECHO database, which is available to the public. MD's Municipal and Significant Industrial Point Source data is exported weekly from ECHO to CBPO's PST database by the CBPO. Results from effluent monitoring are compiled by WPRPP from both PST exports and ICIS queries.

Point source data for insignificant industrial facilities is based on facilities permit design capacity. MDE's Industrial & General Permits Division (IGPD) compiles a list of the facilities and calculates estimates of flow, concentrations, and loads entering Chesapeake Bay. Only a few of these facilities have design capacity changes each year, which must be reported and corrected in the PST database. Whenever such changes occur, WPRPP receives a memorandum from the IGPD outlining the updates.

Permits to irrigation and infiltration dischargers impacting surface water via land application are issued by MDE's Groundwater Discharge Permits Division (GDPD). The monitoring data required by permitted operations are also entered into NetDMR by facility operators and are migrated to ICIS on a daily basis. WPRPP queries and exports the monitoring data from ICIS, identifying the operations by their Maryland permit ID. A list of active permit IDs is requested from the GDPD and the ICIS query is updated to include new facilities and exclude inactive spray irrigation sites.

MDE requires a Sewage Sludge Utilization Permit for the use of biosolids as well as any product containing this material, which includes treatment, composting, transportation, storage, distribution, or application of these materials. MDE tracks the application of biosolids. Estimates are made of the total load of biosolid nutrients applied to land in each sector by county from any WWTP supplying these materials.

The WWTPs that contribute to biosolids applications annually submit a Generator Report to MDE LMA's Biosolids Division. This division manually enters information into MDE's TEMPO database. This includes the total amounts of biosolids sent for land application and the location where they were applied. In February of each year, the Biosolids Division receives a permit report for the Cedarville Lagoon, which is emptied and cleaned out annually and samples are analyzed to determine the composition of the biosolids to be applied. The report provides the dry tons, wet tons, and percent solids of the material removed from the lagoon and the biosolids land applied in Maryland in the prior year. To estimate the nutrient content of the biosolids generated and applied in Maryland, in addition to the Generator Report and Cedarville Lagoon Sewage Sludge Utilization Permit Annual report, constituent reports of class B and un-stabilized biosolids are exported from TEMPO.

B-3. Integrity of Environmental Information

The Compliance Program in MDE's WSA conducts multimedia inspections to determine compliance of water pollution control and resource protection under State and federal regulations. These include NPDES and State Groundwater discharge permits, erosion and sediment controls, tidal and non-tidal wetlands permits and waterway construction permits. The multimedia inspections are conducted by Environmental Compliance Specialists, Sanitarians or Engineers. NPDES inspections are performed in accordance with MDE's 106 Grant work plan under its delegation by EPA.

NPDES inspections of municipal and industrial wastewater treatment plants are designated as a Compliance Sampling Inspection (CSI) or Compliance Evaluation Inspection (CEI). Program inspectors also conduct Performance Audit Inspections (PAIs) at the contract laboratories for municipal and industrial permittees to verify proper analytical methods are being followed.

PAIs are conducted with detailed emphasis on the laboratory and the self-monitoring program. The inspectors evaluate the analytical performance of the laboratory/laboratories as well as the integrity and quality of the analytical data generated for reporting under the Clean Water Act. The permit is reviewed for all aspects of self-monitoring. Proper sampling and analysis techniques are reviewed: sample preservation, proper holding times for testing, appropriate methodology, record keeping, flow monitoring, proper sample type and frequency, and calculations for Discharge Monitoring Reporting. This process takes the sample from the point and manner of collection, through the preservation, testing and documentation to determine that appropriate data is being conveyed to MDE. Further, this process allows for identification of falsified data, and makes referrals to the Office of the Attorney General for investigation and appropriate compliance action.

MDE also reviews and tracks annual laboratory proficiency testing under the US-EPA DMR/QA Program. In this program, the laboratories of all Major and select Minor NPDES permit holders in Maryland are required to analyze unknown proficiency test (PT) samples provided by an EPA approved external vendor. The participating permittees are required to have their testing laboratories obtain and analyze a PT sample for all NPDES permit-specified constituents, including Whole Effluent Toxicity (WET) PT samples. The vendors or PT Providers grade and report the final results to the laboratory and to the state coordinator. The Compliance Program also reviews and approves the biomonitoring study plans and reviews the data from the subsequent testing to verify compliance with aquatic toxicity standards established by NPDES permits and Maryland law and regulation.

B-4. Quality Control

Since State Fiscal Year (SFY) 2019, WPRPP has been using the PST to QA/QC all Municipal and Industrial point source data. The exported data is reviewed for completeness and corrections are made as needed. An overview of the QA/QC process for treated effluents is outlined in **Table 3**.

Table 3. QA/QC Process for Point Source Facility Category.

	MUNICIPALS		**INDUSTRIALS**	
	Sig.	**Insig.**	**Sig.**	**Insig.**
Update facility inventories in PST during the year	X	X	X	X
Run facility-check in PST and make corrections	X	X	X	X
Export data	X	X	X	X
Compare PST data with ICIS report exports in R	X	X	X	
Upload corrected data in PST	X	X	X	X
Run data-checks	X	X	X	X
Notify CBP of prepared data in PST	X	X	X	X

B-4.1 - Significant and Insignificant Municipal Facilities

For both significant and insignificant municipal facilities, the QC process starts by exporting the corresponding DMR data from the PST, followed by exporting ICIS query reports of each individual parameter: flow, biochemical oxygen demand (BOD5), dissolved oxygen (DO), total suspended solids (TSS), nitrogen ammonia (NH3), nitrite plus nitrate (NO23), total organic nitrogen (TON), orthophosphate (PO4), and total phosphorus (TP). The data from both databases are aggregated and compared using R (Appendices A and B), and data discrepancies are verified using NetDMR data and MOR attachments. Based on the reason for the discrepancy and which database contained the correct value, the R code replaces the data accordingly and additional checks are done. Common data discrepancies result from the PST containing a parameter's weekly average instead of the monthly average. All data discrepancies and corrections are documented in a separate Excel file, and any data entry errors identified in the process are provided to MDE WSA's Compliance Program so they can correct them in ICIS or notify the facility when applicable.

In the case of insignificant municipal facilities, same as with the other parameters, total nitrogen (TN) values from PST and ICIS are compared and discrepancies are verified. For significant facilities, however, after the initial data check, TN is calculated from the sum of its constituents (NH3+NO23+TON), and the result is compared with the PST value. If the relative percent difference between calculated TN and PST data is above 30%, NetDMR and ICIS are consulted to determine the appropriate value and identify potential data entry errors.

The last steps in the QA/QC process for municipal facilities are to check if the concentration of PO4 is greater than the concentration of TP and to calculate total organic phosphorus (TOP). The R code first runs the comparison of PO4 and TP concentrations and, upon flagging the occurrences where PO4 is greater than TP, WPRPP verifies the facility's permit for parameter analysis frequency requirements, and reviews NetDMR for data entry errors. WPRPP maintains a list of facility monitoring requirements and, if PO4 is required to be analyzed less frequently than TP, TP concentrations are considered correct and PO4 concentrations are updated to match TP concentrations. Unless there is a typo or a parameter is only available in the MOR attachment on NetDMR requiring the data to be manually entered, all data corrections are done in R. Parameters source columns (e.g., FLOWSource) are updated to reflect the origin of the data, which could be ICIS, NetDMR, MOR, average of available data, or the reason why the parameter value is missing (e.g., no discharge, failed to sample, below detection limit).

This QA/QC process occurs quarterly for significant municipal facilities, and annually for insignificant facilities. Significant municipal facility quarterly results for a given fiscal year are appended after the QA/QC process is completed each quarter and the data is maintained in an MDE internal network drive until it is time to submit them in the PST. Once all municipal facility data have been QA/QCed and are in their final version, the final tables are uploaded to the PST by December 1, and the previous data is overwritten. All PST data checks are conducted to validate the data and final corrections are done if needed.

B-4.2 - Significant Industrial Facilities

The same process conducted for insignificant municipal facilities is conducted for significant industrial facilities (Appendix C). Data is exported from the PST and ICIS for all of the parameters (flow, BOD5, DO, TSS, NH3, NO23, TON, PO4, and TP) except TOP (calculated from TP and PO4). Data from these two sources are then exported and aggregated to compare database content and fill in missing values. Corrections are made as needed and the parameter source column in the PST is updated to reflect the origin of the data. Not all parameters in the PST are required under every facility's permit; therefore, default values are provided instead. After the data is QA/QCed, the final version is uploaded into the PST and run through the PST auto-check process to fill in missing data and balance values.

B-4.3 – Insignificant Industrial Facilities

In the case of insignificant industrial facilities, the previous year's data is copied by CBPO to create a current year dataset. The values in the data set are calculated by the IGPD for new facilities or those that had design changes. Those are incorporated into the dataset before uploading the table into the PST and running the tool's data checks.

B-4.4 – Spray Irrigation

Spray irrigation data queried from ICIS is processed in R (Appendix D) for unit conversions and monthly and annual average calculations. Facilities are not required to report TP data, therefore a default value of 3.15 mg/L is provided for all facilities that had discharges during the fiscal year. Total monthly flow is calculated from the average daily flow in million gallons per day (MGD) and the number of days in each month. All month's flows are added to calculate the annual flow in gallons per year (gal/yr) spray irrigated on each site and the mean of TN monthly average concentrations is calculated to obtain TN annual average concentration in mg/L. The final spray irrigation data is formatted according to the PST template and uploaded into the PST by December 1.

B-4.5 - Biosolids

Biosolids data submitted annually to CBPO are calculated from the individual facilities estimated dry tons and nutrient contents in their biosolids, and the amount of biosolids applied to agricultural land, marginal land reclamation, distributed and marketed, and used or disposed of in a landfill across the state. The calculations are done in R (Appendix E) and the necessary input files are Class B and Un-Stabilized Sewage Sludge Constituents reports, and the Generator report. Additionally, since the Generator Report does not include information on what is removed from the Cedarville Lagoon for land application, the dry tons of biosolids and where they were applied are obtained from the Cedarville Lagoon's Sewage Sludge Utilization Permit Annual Report provided by MDE LMA's Biosolids Division and are incorporated in the calculations.

The constituents' reports are used to estimate percent dry content, ammonium (NH4), nitrate nitrogen (NO3), total Kjeldahl nitrogen (TKN), TON (from TKN – NH4), TN

(TON+NH4+NO3), TP, PO4 (TP÷2), and TOP (TP÷2) in biosolids generated at each MD county. Those averages are used to calculate tons of dry content and loads of nutrients applied in each county during the calendar year. The amount of biosolids imported into MD that is distributed and marketed is divided by the number of counties applying biosolids and incorporated into their total tons of biosolids used prior to calculating nutrient loads. The final biosolids data is then formatted according to the PST template provided and is uploaded into the PST by December 1. It is important to note that, since biosolids data is reported to MDE once per year and on a calendar year basis, for each fiscal year submission to CPBO, MDE submits the previous calendar year data on biosolids (e.g., FY2023 submission will contain calendar year 2022 biosolids application in Maryland).

B-5. Instruments/Equipment calibration, testing, inspection, and maintenance

The nature of this project does not involve the use of specialized equipment requiring calibration, testing, or maintenance, as it primarily relies on the use of data pulled from online databases and formatting in data management software. This project requires querying and formatting data from a variety of sources. The only equipment required for this project is a desktop computer with internet access, Microsoft Excel, and an installation of the R statistics software.

B-6. Inspection/acceptance of supplies and services

Given that this project does not entail the utilization of specialized equipment and does not involve contractors or subcontractors, there is no need for inspection of external supplies or services.

B-7. Environmental Information Management

The pathways from data generation by wastewater facilities into the databases referenced in sections A.-1.2.1 through A.-1.2.3 are presented in **Figure 2**. Documentation of the methods for obtaining the data for this project are presented in sections B-2 and B-4.

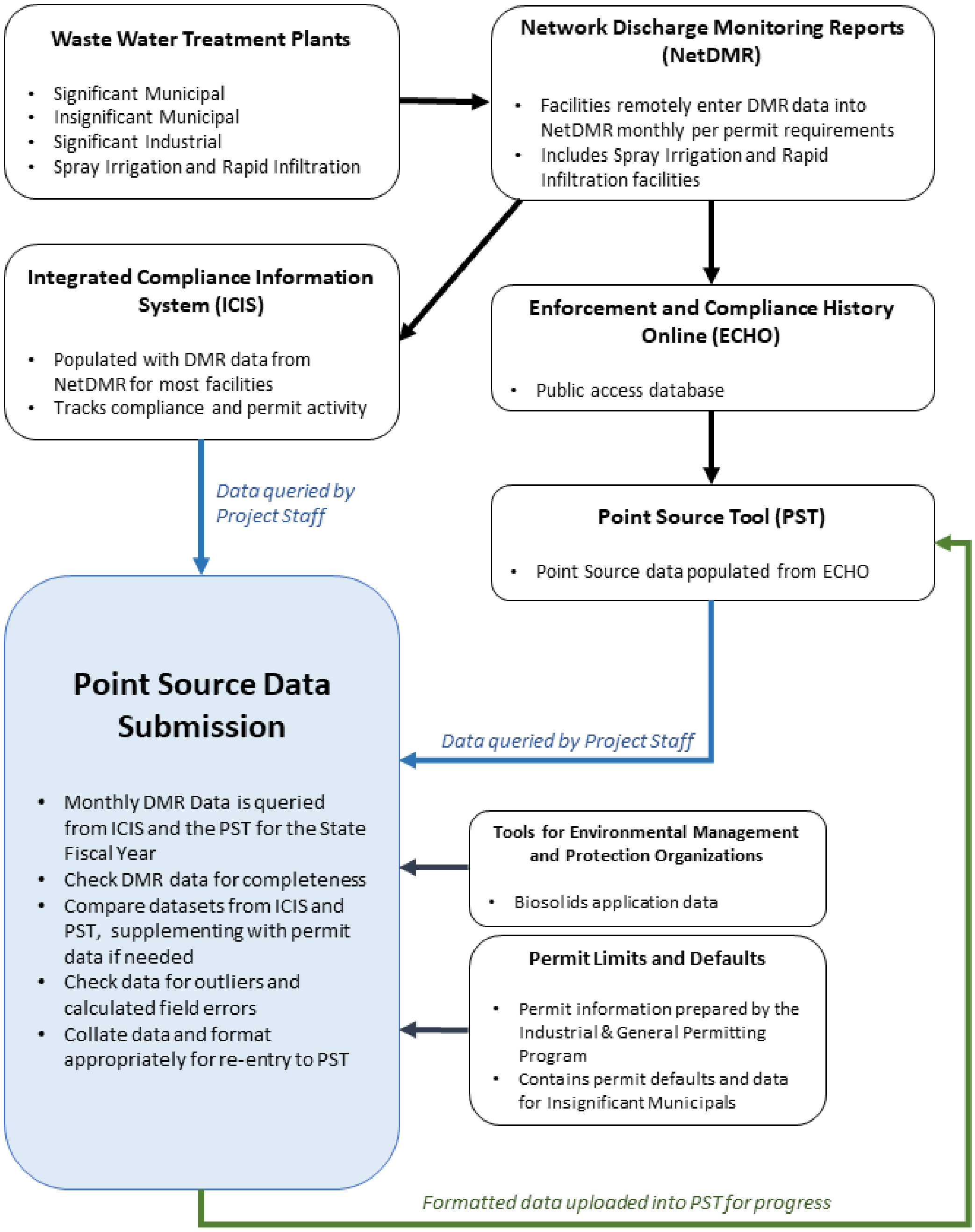

Figure 2. Maryland Point Source Data Flow.

Blue arrows indicate data sources for the Point Source Data Submission (represented with green arrow).

The majority of the work performed for this project involves processing, compiling and analyzing data for the indicators described in Section A-3. During the process of assessing the data for completeness and accuracy, any errors are corrected at the time of detection and documented in the intermediary files and a data mismatch log. Many of these errors relate to incomplete fields or incorrect units. Calculation of Total Nitrogen and Total Phosphorus is also checked and corrected if necessary. Details on the corrective actions taken to ensure the PSDS is complete and accurate are documented in Sections B-2 and B-4.

There is no specialized equipment required for this project. The process requires a computer with internet access, Microsoft Excel, and an installation of R (Version 1.2.1093 or later). Annotated R code for each step of the process is documented in Appendix A through Appendix D.

C. Assessment and Oversight

C-1. Assessment and Response Actions

C-1.1. Assessment Actions

In addition to the assessments conducted during the quality control process described in section B4 for data accuracy and completeness, the Technical Lead conducts annual assessments aimed at evaluating the representativeness of each facility's data. The comprehensive evaluation involves comparing full datasets from each facility with the corresponding data from the previous year. This comparative analysis serves to identify any values that fall outside the established typical range for that facility, signifying potential anomalies. Additionally, the interquartile range of each facility's data is calculated to determine if an observation is an outlier. Outlier assessment involves the systematic examination of data points that deviate significantly from the expected or normal distribution within a given dataset. By implementing outlier detection techniques, WPRPP aims to pinpoint and scrutinize data values that may be indicative of unusual or unexpected conditions.

Any data entry errors and issues with facilities identified during the quality control in section B4 and the above-described assessment are documented in an Excel file and relevant files (e.g., MOR attachments, ICIS reports) are saved on an internal server. Data entry errors are communicated to the Compliance Program immediately within one month of data review completion. Issues that require contacting the facility are referred to the Wastewater Pollution Prevention & Reclamation Program, Compliance Program, and/or the Biosolids Division as appropriate.

C-1.2. Response Actions

As mentioned in section A-7, since WPRPP does not hold regulatory authority over permit compliance and reporting, personnel from other programs listed in Table 2 are contacted in the event non-conformities are identified and corrective and response actions are necessary.

C-2. Oversight and Reports to Management

While compiling data for the PSDS, the project team may discover issues related to data availability or completeness linked to permit circumstances. Upon identification of these issues, the Project Manager or Technical Lead communicates with the pertinent point source permitting staff via email. Table 2 contains the contact details for the points of contact for compliance related issues.

The most common issue that is communicated to staff is errors with incomplete data, incorrect units or decimal placement, and errors with calculated fields. These errors may be unrelated to permit compliance but are necessary to fix for future data submissions. After finalizing the PSDS, project staff send a list of these inconsistencies to the appropriate WSA staff. The list contains the facility, a description of the issue, and examples of the problems encountered. It may also include an explanation of how the issue impacts the data submission. Permitting staff utilize this list to contact facilities and notify facility reporters of errors with their DMRs.

D. Environmental Information Review and Usability Determination

D-1. Environmental Information Review

Data validation is accomplished for this project through the use of the Chesapeake Bay Model's Point Source Tool (PST). When a dataset is submitted into the PST, the web tool performs a series of validation checks on the completeness of the data. The individual submitting the data reviews the information and systematically analyzes it using the tool's step-by-step process. Utilizing this sequence of "checks," the State submitter has the option to apply automated corrections to data identified by the PST.

Once data has been submitted and all of the PST checks are complete, the project is considered complete, and there is no need for additional review by project staff.

D-2. Useability Determination

Data that is gathered and formatted in accordance with the method described in this QAPP is submitted to CBPO in order to be utilized in a progress scenario of the Chesapeake Bay Model. Data is provided in the precise formatting and temporal coverage that matches the needs of the progress scenario and Chesapeake Bay Modeling team. Project staff do not make determinations on the usability of the data - provided the PSDS is in the correct format and contains all of Maryland's currently operating facilities, the data is considered usable for the model.

If issues with data usability do arise, they are found by the Chesapeake Bay Modeling team. These issues may be related to the submission format, calculation errors, or other unforeseen issues with data entry. These issues will be communicated to project staff by CBPO or other parties related to the use of the data in the Chesapeake Bay Model.

APPENDIX A - Municipal Significant Quarterly Reports

2022-06-28

Intro

This code provides the steps to compare the information available in the Point Source tool and ICIS and generate a report for every quarter regarding pollutant levels from municipal wastewater treatment facilities with significant discharges.

```
# if any of the packages has not been installed yet, remove "#" preceding function
#install.packages("readxl")
#install.packages("tidyverse")
#install.packages("openxlsx")

library("readxl")
library("tidyverse")
library("openxlsx")
```

Editable code

Some of the code needs to be edited each quarter. The information that needs editing is:

startPeriod → The day, month and year when the quarter starts. Use:

Q1	20XX-07-01
Q2	20XX-10-01
Q3	20XX-01-01
Q4	20XX-04-01

endPeriod → The day, month and year when the quarter ends. Use:

Q1	20XX-09-30
Q2	20XX-12-31
Q3	20XX-03-31
Q4	20XX-06-30

year -> fiscal year data corresponds to

quarter -> fiscal year quarter data corresponds to

```
#Set up period dates below and run it:

startPeriod <- "2023-07-01"
endPeriod <- "2023-09-30"
year <- "2024"
quarter <- "Q1"
```

Data

The files to be imported include the table exported from the PS tool ("PS TOOL TABLE.xlsx"), and ICIS reports for each constituent (BOD5, DO, FLOW, NH3, NO23, PO4, TON, TP, TSS), named as "`constituent` SIG ICIS". Once files are downloaded from the web applications, save them in the 'Input_files' folder of the project in their respective years and fiscal year quarter.

```
# Set working directory to project location, not to the file location.
# This can be checked by clicking on the arrow to the right of the `Knit` icon >
# 'Knit Directory' and make sure 'Project Directory' is selected
getwd()

PS_data <- read_excel(
  paste0("Input_data/",year,"/Quarterly_Reports/", quarter, "/PS TOOL TABLE.xlsx"),
  sheet = 2)
```

Formatting ICIS tables

The first step of the process is to format each table obtained from ICIS to have the columns equivalent to the PS TOOL TABLE. This parameters are split due to a slightly different processing needed: BOD5, NH3, TP and TSS PS tool values had some instances where maximum weekly average information was pulled from NetDMR instead of maximum monthly average.

The process will repeat through a for loop for every parameter.

```
# create parameter list
parameter1 <- c("FLOW",  "DO",  "NO23", "PO4", "TON")
parameter2 <- c("BOD5","NH3","TP", "TSS")

ICIS_data <- list()

# for loop to generate each parameter table (ICIS)
for(i in parameter1){
  ICIS_data[[i]] <- read_excel(paste0("Input_data/",year,"/Quarterly_Reports/",
                                      quarter,"/", i," ", "SIG ICIS.xlsx"),
                               sheet=1)
  ICIS_data[[i]] <- ICIS_data[[i]][-c(5,8)]
  names(ICIS_data[[i]]) <- c("NPDES ID","Facility","Outfall","Period",
                             "Description",i,"NODI Code","NODI Description")
  ICIS_data[[i]] <- drop_na(ICIS_data[[i]], i)
  ICIS_data[[i]][6] <- apply(ICIS_data[[i]][6], 2,
                      function(x) as.numeric(as.character(gsub("<","",x))))
  ICIS_data[[i]] <- ICIS_data[[i]] %>%
  group_by(`NPDES ID`,`Facility`,`Outfall`,
           `Period`,
           `Description`) %>%
  summarize(across(where(is.numeric), ~.x[!is.na(.x)][1]),
            across(where(is.character),~.x[!is.na(.x)][1]))
}


# for loop to generate other set of parameters tables
for(j in parameter2){
  ICIS_data[[j]] <- read_excel(paste0("Input_data/",year,"/Quarterly_Reports/",
                                      quarter,"/", j," ", "SIG ICIS.xlsx"),
```

```
                              sheet=1)
  ICIS_data[[j]] <- ICIS_data[[j]][-c(5,8,10)]
  names(ICIS_data[[j]]) <- c("NPDES ID","Facility","Outfall","Period",
                             "Description",j,paste0(j,"_weekly"),"NODI Code",
                             "NODI Description")
  ICIS_data[[j]] <- ICIS_data[[j]] %>%
    mutate(Description = "MX AV/MO AVG")
  ICIS_data[[j]][c(6,7)] <- apply(ICIS_data[[j]][c(6,7)], 2,
                     function(x) as.numeric(as.character(gsub("<","",x))))
  ICIS_data[[j]] <- ICIS_data[[j]] %>%
  group_by(`NPDES ID`,`Facility`,`Outfall`,
          `Period`,
          `Description`) %>%
  summarize(across(where(is.numeric), ~.x[!is.na(.x)][1]),
           across(where(is.character),~.x[!is.na(.x)][1]))
  }
```

Select period data from PS TOOL TABLE

The next step is to filter the data for the quarter of interest so ICIS and PS tool information can be joined for comparison.

```
PS_data <- filter(PS_data, Period >= startPeriod & Period <= endPeriod)

PS_list <- list()
parameter <- c("FLOW","BOD5","DO","NH3","NO23","PO4","TP","TSS","TON")

for(i in parameter){
  f <- c("NPDES ID", "Facility", "Outfall", "Period", i)
  PS_list[[i]] <- select(PS_data, all_of(f))
  PS_list[[i]]
}
```

Combining both ICIS and PS tables for each individual parameter

```
merged_list <- list()

for (i in parameter1){
 levels <- c("NPDES ID","Outfall","Period")
 merged_list[[i]] <- full_join(PS_list[[i]], ICIS_data[[i]], by=levels)
 merged_list[[i]] <- merged_list[[i]] %>% mutate(new.column = "ICIS VALUE")
 names(merged_list[[i]])[11] <- paste(i,"_SOURCE", sep="")
 merged_list[[i]] <- merged_list[[i]] %>% mutate(
   ifelse(!is.na(merged_list[[i]][5])&is.na(merged_list[[i]][8]),"FALSE",
          ifelse(merged_list[[i]][5] == merged_list[[i]][8],"", "FALSE")))
 names(merged_list[[i]])[12] <- "PSTool.x == ICIS.y"
 merged_list[[i]][12][is.na(merged_list[[i]][12])] <- "CHECK"
}

for (i in parameter2){
 levels <- c("NPDES ID","Outfall","Period")
 merged_list[[i]] <- full_join(PS_list[[i]], ICIS_data[[i]], by=levels)
 merged_list[[i]] <- merged_list[[i]] %>% mutate(new.column = "ICIS VALUE")
 names(merged_list[[i]])[12] <- paste(i,"_SOURCE", sep="")
 merged_list[[i]] <- merged_list[[i]] %>%
   mutate(ifelse(merged_list[[i]][5] != merged_list[[i]][8],
```

```
                 ifelse(merged_list[[i]][5] != merged_list[[i]][8] &
                          merged_list[[i]][5] == merged_list[[i]][9],"WEEKLY",
                        ifelse(is.na(merged_list[[i]][8]) &
                                 merged_list[[i]][5] == merged_list[[i]][9],
                               "WEEKLY","FALSE")),""))
 names(merged_list[[i]])[13] <- "PSTool.x == ICIS.y"
 merged_list[[i]][13][is.na(merged_list[[i]][13])] <- "CHECK"}

# reorder list
merged_list <- merged_list[c(1,6,2,7,3,4,8,9,5)]
```

Writing edited tables into one Excel file

Once the ICIS and PS TOOL TABLE information were merged by parameter, the tables have to be written in separate tabs within an excel file.

```
#first, the replacement table needs to be added to merged_list

final_list <- append(merged_list,list(PS_data),after = 0)
names(final_list)[1] <- "ReplacementTable"

OUT <- createWorkbook()

# add each dataframe from the final_list to a tab in the workbook,
# naming them by parameter

lapply(names(final_list), function(x){
  addWorksheet(OUT,x)
  writeData(OUT, x, x = final_list[[x]])})

saveWorkbook(OUT, paste0("Output_data/",year, "/Quarterly_Reports/", quarter,
                         "/",quarter,"_",year,"_SigMun_ReplacementTable.xlsx"),
             overwrite=TRUE)
```

STOP HERE AND CHECK OUTPUT EXCEL FILE!!!

QA/QC the output file to make sure everything looks good. Create a log of the issues found (file SigMun_Data_Mismatches_Log_QXFY20XX). Change column L or M (if parameter goes through the Weekly check) if flagged data was checked and no issues with PS tool data was found (CHECKED). Any value or NODI code change should be written in red to identify manually edited/entered data. If the reason for data mismatch could not be identified, highlight the cells in yellow and consult permitting or compliance for clarification. Make sure facilities with data different than zero do not have a qualifier code (should be deleted, otherwise further code will replace value with zero). If the mismatch assessment concluded that ICIS value is correct and PS tool value needs to be replaced, no manual change is necessary because the R code will take care of that issue in the upcoming code chunks.

Delete additional facilities (bottom rows contain facilities only present in ICIS report outputs, no facility name in PS tool column B and usually an atypical outfall number) and any potential duplicates before starting the next step.

Adjusting PS tool data

Clean the environment before starting this next step.

In this step, based on the content of the QAQCed file, the information on the replacement table will be updated.

If NODI Code is not NA: value replaced with zero and SOURCE column changed to NODI description. If check column has "WEEKLY", "CHECK" or "FALSE", value in the replacement table should be replaced with ICIS value.

```
# Set up period dates again below

startPeriod <- "2023-07-01"
endPeriod <- "2023-09-30"
year <- "2024"
quarter <- "Q1"

# Load the adjusted data
read_excel_allsheets <- function(filename) {
  sheets <- excel_sheets(filename)
  x <-    lapply(sheets, function(X) readxl::read_excel(filename, sheet = X))
  names(x) <- sheets
  x
}
corrected_list <- read_excel_allsheets(paste0("Output_data/",year,
                                              "/Quarterly_Reports/",quarter,
                                               "/",quarter,"_",year,
                                               "_SigMun_ReplacementTable.xlsx"))
names(corrected_list) <- c("ReplacementTable","flow","bod5","do","nh3","no23",
                           "po4","tp","tss","ton")
list2env(corrected_list,envir=.GlobalEnv)

# FLOW
flow <- flow %>%
  mutate(FLOW.x = ifelse(is.na(`PSTool.x == ICIS.y`),FLOW.x,
                       ifelse(`PSTool.x == ICIS.y` == "WEEKLY",FLOW.y,
                              ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                       !is.na(FLOW.y),FLOW.y,
                                     ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                       !is.na(FLOW.y),
                                       FLOW.y,FLOW.x)))),
         FLOW.x= ifelse(is.na(`NODI Code`),FLOW.x,0),
         FLOW_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                              FLOW_SOURCE),
         check=ifelse(FLOW.x==FLOW.y,"","FALSE"))



# BOD5
bod5 <- bod5 %>%
  mutate(BOD5.x = ifelse(is.na(`PSTool.x == ICIS.y`),BOD5.x,
                       ifelse(`PSTool.x == ICIS.y`== "WEEKLY", BOD5.y,
                              ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                       !is.na(BOD5.y),BOD5.y,
                                     ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                       !is.na(BOD5.y),
```

```
                                          BOD5.y,BOD5.x)))),
         BOD5.x = ifelse(is.na(`NODI Code`),BOD5.x,0),
         BOD5_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                              BOD5_SOURCE),
         check=ifelse(BOD5.x==BOD5.y,"","FALSE"))

# DO
do <- do %>%
  mutate(DO.x = ifelse(is.na(`PSTool.x == ICIS.y`),DO.x,
                       ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                !is.na(DO.y),DO.y,
                              ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                       !is.na(DO.y),
                                     DO.y,DO.x))),
         DO.x = ifelse(is.na(`NODI Code`),DO.x,0),
         DO_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                            DO_SOURCE),
         check=ifelse(DO.x==DO.y,"", "FALSE"))

#NH3
nh3 <- nh3 %>%
  mutate(NH3.x = ifelse(is.na(`PSTool.x == ICIS.y`), NH3.x,
                        ifelse(`PSTool.x == ICIS.y`== "WEEKLY", NH3.y,
                               ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                        !is.na(NH3.y),NH3.y,
                                      ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                        !is.na(NH3.y),
                                        NH3.y,NH3.x)))),
         NH3.x = ifelse(is.na(`NODI Code`),NH3.x,0),
         NH3_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                             NH3_SOURCE),
         check=ifelse(NH3.x==NH3.y,"","FALSE"))

#NO23
no23 <- no23 %>%
  mutate(NO23.x = ifelse(is.na(`PSTool.x == ICIS.y`),NO23.x,
                         ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                  !is.na(NO23.y),NO23.y,
                                ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                         !is.na(NO23.y),
                                       NO23.y,NO23.x))),
         NO23.x = ifelse(is.na(`NODI Code`),NO23.x,0),
         NO23_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                              NO23_SOURCE),
         check=ifelse(NO23.x==NO23.y,"", "FALSE"))

#PO4
po4 <- po4 %>%
  mutate(PO4.x = ifelse(is.na(`PSTool.x == ICIS.y`),PO4.x,
                        ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                 !is.na(PO4.y),PO4.y,
                               ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                        !is.na(PO4.y),
                                      PO4.y,PO4.x))),
         PO4.x = ifelse(is.na(`NODI Code`),PO4.x,0),
         PO4_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                             PO4_SOURCE),
         check=ifelse(PO4.x==PO4.y,"", "FALSE"))
```

```
#TP
tp <- tp %>%
  mutate(TP.x = ifelse(is.na(`PSTool.x == ICIS.y`),TP.x,
                      ifelse(`PSTool.x == ICIS.y`== "WEEKLY", TP.y,
                              ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                        !is.na(TP.y),TP.y,
                                     ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                        !is.na(TP.y),
                                        TP.y,TP.x)))),
         TP.x = ifelse(is.na(`NODI Code`),TP.x,0),
         TP_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                            TP_SOURCE),
         check=ifelse(TP.x==TP.y,"","FALSE"))

#TSS
tss <- tss %>%
  mutate(TSS.x = ifelse(is.na(`PSTool.x == ICIS.y`),TSS.x,
                       ifelse(`PSTool.x == ICIS.y`== "WEEKLY", TSS.y,
                               ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                        !is.na(TSS.y),
                                      TSS.y,
                                      ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                        !is.na(TSS.y),
                                        TSS.y,TSS.x)))),
         TSS.x = ifelse(is.na(`NODI Code`),TSS.x,0),
         TSS_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                             TSS_SOURCE),
         check=ifelse(TSS.x==TSS.y,"","FALSE"))

#TON
ton <- ton %>%
  mutate(TON.x = ifelse(is.na(`PSTool.x == ICIS.y`),TON.x,
                        ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                 !is.na(TON.y),TON.y,
                               ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                        !is.na(TON.y),
                                        TON.y,TON.x))),
         TON.x = ifelse(is.na(`NODI Code`),TON.x,0),
         TON_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                             TON_SOURCE),
         TON_SOURCE = ifelse(TON.x<0,"Negative value corrected",
                             TON_SOURCE),
         TON.x = ifelse(TON.x<0,0,TON.x),
         check=ifelse(TON.x==TON.y,"","FALSE"))

# Replace values in replacement table and put all dataframes in a list

FinalTable <- ReplacementTable %>%
  mutate(FLOW=flow$FLOW.x,
         FLOWSource=flow$FLOW_SOURCE,
         BOD5=bod5$BOD5.x,
         BOD5Source=bod5$BOD5_SOURCE,
         DO=do$DO.x,
         DOSource=do$DO_SOURCE,
         NH3=nh3$NH3.x,
         NH3Source=nh3$NH3_SOURCE,
         NO23=no23$NO23.x,
         NO23Source=no23$NO23_SOURCE,
         PO4=po4$PO4.x,
```

```
         PO4Source=po4$PO4_SOURCE,
         TP=tp$TP.x,
         TPSource=tp$TP_SOURCE,
         TSS=tss$TSS.x,
         TSSSource=tss$TSS_SOURCE,
         TON=ton$TON.x,
         TONSource=ton$TON_SOURCE
         )

df_list <- list(ReplacementTable,FinalTable, flow,bod5,do,nh3,no23,po4,
                tp,tss,ton)
names(df_list) <-c("ReplacementTable","FinalTable","FLOW","BOD5","DO","NH3",
                   "NO23","PO4","TP","TSS","TON")

# Save file
OUT <- createWorkbook()

lapply(names(df_list), function(x){
  addWorksheet(OUT,x)
  writeData(OUT, x, x = df_list[[x]])})

saveWorkbook(OUT, paste0("Output_data/",year, "/Quarterly_Reports/", quarter,
                         "/",quarter,"_",year,
                         "_SigMun_ReplacementTable_Final.xlsx"),
             overwrite=TRUE)
```

QA/QC the output file to make sure everything looks good and there are no outstanding issues in each constituent's tab.

Calculated values

Total Nitrogen

Total nitrogen levels are calculated by adding the concentrations of NH3, TON and NO23. After doing the first quality control check, the code below will combine those data.

```
# Load table in R and calculate field
Final_Table <- read_excel(
  paste0("Output_data/",year, "/Quarterly_Reports/", quarter,"/",
         quarter,"_",year,"_SigMun_ReplacementTable_Final.xlsx"),
  sheet = 2)

Final_Table_TN <- Final_Table[c(4,5,8,10,14,15,20)]

Final_Table_TN <- filter(Final_Table_TN, Period >= startPeriod &
                           Period <= endPeriod) %>%
  mutate(TN = NH3+TON+NO23,
         TN_Source = ifelse(!is.na(TN), "Calculated as NH3+NO23+TON",
                            "ICIS VALUE"),
         TN_PS = ReplacementTable$TN,
         Check = ifelse(TN==TN_PS,"",FALSE),
         PCT_difference = as.numeric(ifelse(Check=="FALSE",
           round(((TN-TN_PS)/((TN+TN_PS)/2))*100,digits=3),"")))

# Write resulting table in a new tab of the Excel file
wb <- loadWorkbook(paste0("Output_data/",year, "/Quarterly_Reports/", quarter,
                          "/",quarter,"_",year,
```

```
                          "_SigMun_ReplacementTable_Final.xlsx"))

addWorksheet(wb,"TN")

DF <- data.frame(Final_Table_TN)

writeData(wb,"TN",DF)

saveWorkbook(wb,
             paste0("Output_data/",year, "/Quarterly_Reports/", quarter,"/",
                    quarter,"_",year,"_SigMun_ReplacementTable_Final.xlsx"),
             overwrite = TRUE)
```

QA/QC the resulting tab in the file (TN tab) to make sure everything looks good. Carefully look into data records where relative percent difference between calculated TN and TN provided by the facility was above 30%. Take notes in the "SigMun_Data_Mismatches_Log_QXFY20XX" and share cases that need ICIS editing with compliance.

PO4 over Total Phosphorus

In some instances, PO4 concentrations recorded are higher than total phosphorus concentrations. To adjust this, we run a code to replace the TP with PO4 concentrations when [PO4 > [TP].However, before that replacement occurs, those instances of PO4 > TP are reviewed for data entry errors or cases when the facility needs to be contacted by compliance.

```
# Load table in R and add the check columns
PO4_Over_TP <- read_excel(
  paste0("Output_data/",year, "/Quarterly_Reports/", quarter,"/",
         quarter,"_",year,"_SigMun_ReplacementTable_Final.xlsx"),
  sheet = 2)

PO4_Over_TP <- PO4_Over_TP[c(4,5,8,10,17,18,28)]

PO4_Over_TP <- filter(PO4_Over_TP, Period >= startPeriod &
                        Period <= endPeriod) %>%
      mutate(PO4_Over_TP = ifelse(PO4 > TP, TP, PO4),
             PO4_Source = ifelse(PO4 > TP, "EQUALLED TO TP", PO4Source))

# Write resulting table in a new tab of the Excel file

wb <- loadWorkbook(
  paste0("Output_data/",year, "/Quarterly_Reports/", quarter,"/",quarter,"_",
         year,"_SigMun_ReplacementTable_Final.xlsx"))

addWorksheet(wb,"PO4 Over TP")

DF <- data.frame(PO4_Over_TP)

writeData(wb,"PO4 Over TP",DF)

saveWorkbook(wb,
             paste0("Output_data/",year, "/Quarterly_Reports/", quarter,"/",
              quarter,"_",year,"_SigMun_ReplacementTable_Final.xlsx"),
             overwrite = TRUE)
```

QA/QC the resulting tab in the file (PO4 Over TP tab) to make sure everything looks good. Carefully look into data records where PO4 concentration is higher than TP. In most cases, TP is sampled more frequently and that is why the results differ. Check Discharge permits to see if that is the case. Take notes of any incorrect data "SigMun_Data_Mismatches_Log_QXFY20XX" and share cases that need ICIS editing with compliance.

Replace PO4 correct values

Before calculating TOP from TP and PO4, we have to replace the PO4 values we corrected from the previous step (by changing the cases where PO4 was higher than TP to TP concentrations) into the FinalTable.

```
Final_Table <- read_excel(
  paste0("Output_data/",year, "/Quarterly_Reports/", quarter,"/",
         quarter,"_",year,"_SigMun_ReplacementTable_Final.xlsx"),
  sheet = 2)

# added the part below because sometimes changes are made in TN or PO4
ed_TN <- read_excel(
  paste0("Output_data/",year, "/Quarterly_Reports/", quarter,"/",
         quarter,"_",year,"_SigMun_ReplacementTable_Final.xlsx"),
  sheet = 12)

ed_PO4 <- read_excel(
  paste0("Output_data/",year, "/Quarterly_Reports/", quarter,"/",
         quarter,"_",year,"_SigMun_ReplacementTable_Final.xlsx"),
  sheet = 13)

# Edit final table
Final_Table <- Final_Table %>%
  mutate(TN = ed_TN$TN,
         TNSource = ed_TN$TN_Source,
         PO4 = ed_PO4$PO4_Over_TP,
         PO4Source = ed_PO4$PO4_Source)

# Replace "FinalTable" tab with this one

wb  <- loadWorkbook(
  paste0("Output_data/",year, "/Quarterly_Reports/", quarter,"/",
         quarter,"_",year,"_SigMun_ReplacementTable_Final.xlsx"))
writeData(wb, sheet="FinalTable", Final_Table)
saveWorkbook(wb,paste0("Output_data/",year, "/Quarterly_Reports/", quarter,"/",
                       quarter,"_",year,"_SigMun_ReplacementTable_Final.xlsx"),
             overwrite = TRUE)
```

Total Organic Phosphorus

Last tab that needs to be added to the file is the Total Organic Phosphorus (TOP), calculated by subtracting PO4 from TP.

```
# Load table in R and include TOP column with appropriate control checks
Final_Table_TOP <- read_excel(
  paste0("Output_data/",year, "/Quarterly_Reports/", quarter,"/",
   quarter,"_",year,"_SigMun_ReplacementTable_Final.xlsx"),
```

```
  sheet = 2)

Final_Table_TOP <- Final_Table_TOP[c(4,5,8,10,17,18)]

Final_Table_TOP <- filter(Final_Table_TOP, Period >= startPeriod &
                            Period <= endPeriod) %>%
    mutate(TOP = TP-PO4,
           TOP_Source = "Calculated as TP - PO4")

# Write final table into new tab of the Excel file
wb <- loadWorkbook(
  paste0("Output_data/",year, "/Quarterly_Reports/", quarter,"/",
         quarter,"_",year,"_SigMun_ReplacementTable_Final.xlsx"))

addWorksheet(wb,"TOP")

DF <- data.frame(Final_Table_TOP)

writeData(wb,"TOP",DF)

saveWorkbook(wb,
             paste0("Output_data/",year, "/Quarterly_Reports/", quarter,"/",
                    quarter,"_",year,"_SigMun_ReplacementTable_Final.xlsx"),
             overwrite = TRUE)
```

QA/QC the resulting tab in the file (TOP tab) to make sure everything looks good.

Adding final values to the PS table

```
Final_Table <- Final_Table %>%
  mutate(TOP = Final_Table_TOP$TOP,
         TOPSource = Final_Table_TOP$TOP_Source
         )

# Replace "FinalTable" tab with this one

wb  <- loadWorkbook(
  paste0("Output_data/",year, "/Quarterly_Reports/", quarter,"/",
         quarter,"_",year,"_SigMun_ReplacementTable_Final.xlsx"))
writeData(wb, sheet="FinalTable", Final_Table)
saveWorkbook(wb,paste0("Output_data/",year, "/Quarterly_Reports/", quarter,"/",
                       quarter,"_",year,"_SigMun_ReplacementTable_Final.xlsx"),
             overwrite = TRUE)
```

APPENDIX B - Municipal Insignificant Annual Report

2023-11-20

Intro

This code provides the steps to compare the information available in the Point Source tool and ICIS and generate a report for each fiscal year regarding pollutant levels from municipal wastewater treatment facilities with insignificant discharges.

```
# if any of the packages has not been installed yet, remove "#" preceding function
#install.packages("readxl")
#install.packages("tidyverse")
#install.packages("openxlsx")

library("readxl")
library("tidyverse")
library("openxlsx")
```

Editable code

Some of the code needs to be edited each fiscal year. The information that needs editing is:

fy –> fiscal year

```
#Set up period fiscal year below and remove # preceding the code to run it:

fy <- 2023
```

Data

The files to be imported include the table exported from the PS tool (and renamed as "PS TOOL TABLE INSIG MUN.xlsx"), and ICIS reports for each constituent (BOD5, DO, FLOW, NH3, NO23, TN, PO4, TON, TP, TSS), which should already be named appropriately due to the name of the reports themselves. Once files are downloaded from the web applications, save them in the 'Input_files' folder of the project in their respective year and annual reports folder.

```
# Set working directory to project location, not to the file location.
getwd()

PS_data <- read_excel(
  paste0("Input_data/",fy,"/Annual_Reports/PS TOOL TABLE INSIG MUN.xlsx"),
  sheet = 2)
```

Formatting ICIS tables

The first step of the process is format each table obtained from ICIS to have the columns equivalent to PS TOOL TABLE.The parameters are split due to a slightly different processing needed: BOD5, TP, TSS, and TN PS tool values had some instances where the weekly average information was pulled from NetDMR instead of the monthly average.

The process will repeat through a for loop for every parameter.

```r
# create parameter lists
parameter1 <- c("FLOW", "DO", "NH3", "NO23", "PO4", "TON")
parameter2 <- c("BOD5","TP", "TSS", "TN") # added weekly vs. monthly check


ICIS_data <- list()

# for loop to generate each parameter table (ICIS)
for(i in parameter1){
  ICIS_data[[i]] <- read_excel(paste0("Input_data/",fy,"/Annual_Reports/", i,
                                      " ", "INSIG ICIS FY.xlsx"), sheet=1)
  ICIS_data[[i]] <- ICIS_data[[i]][-c(5,8)]
  names(ICIS_data[[i]]) <- c("NPDES ID","Facility","Outfall","Period",
                              "Description",i,"NODI Code","NODI Description")
  ICIS_data[[i]] <- drop_na(ICIS_data[[i]], i)
  ICIS_data[[i]][6] <- apply(ICIS_data[[i]][6], 2,
                        function(x) as.numeric(as.character(gsub("<","",x))))
  ICIS_data[[i]][6][is.na(ICIS_data[[i]][6])] <- 0
  ICIS_data[[i]]
}

# for loop to generate other set of parameters tables
for(j in parameter2){
  ICIS_data[[j]] <- read_excel(paste0("Input_data/",fy,"/Annual_Reports/", j,
                                      " ", "INSIG ICIS FY.xlsx"), sheet=1)
  ICIS_data[[j]] <- ICIS_data[[j]][-c(5,8,10)]
  names(ICIS_data[[j]]) <- c("NPDES ID","Facility","Outfall","Period",
                              "Description",j,paste0(j,"_weekly"),"NODI Code",
                              "NODI Description")
  ICIS_data[[j]] <- ICIS_data[[j]] %>%
    mutate(Description = "MX AV/MO AVG")
  ICIS_data[[j]][c(6,7)] <- apply(ICIS_data[[j]][c(6,7)], 2,
                        function(x) as.numeric(as.character(gsub("<","",x))))
  ICIS_data[[j]] <- ICIS_data[[j]] %>%
  group_by(`NPDES ID`,`Facility`,`Outfall`,
           `Period`,
           `Description`) %>%
  summarize(across(where(is.numeric), ~.x[!is.na(.x)][1]),
            across(where(is.character),~.x[!is.na(.x)][1]))
  }
```

Split PS tool by constituents

In the case of municipal insignificant, there is no need to select a specific period of the PS tool table since the entire fiscal year is used in the comparison. A list of dataframes is created containing each constituent information in a different dataframe.

```r
PS_list <- list()

parameter <- c("FLOW","BOD5","DO","NH3","NO23","TN","PO4","TP","TSS","TON")

for (i in parameter){
  f <- c("NPDES ID", "Facility", "Outfall", "Period", i)
  PS_list[[i]] <- select(PS_data, all_of(f))
  PS_list[[i]]
}
```

Combining both ICIS and PS tables for each individual parameter

```
merged_list <- list()

for (i in parameter1){
 levels <- c("NPDES ID","Outfall","Period")
 merged_list[[i]] <- full_join(PS_list[[i]], ICIS_data[[i]], by=levels)
 merged_list[[i]] <- merged_list[[i]] %>% mutate(new.column = "ICIS VALUE")
 names(merged_list[[i]])[11] <- paste(i,"_SOURCE", sep="")
 merged_list[[i]] <- merged_list[[i]] %>% mutate(
   ifelse(!is.na(merged_list[[i]][5])&is.na(merged_list[[i]][8]),"FALSE",
          ifelse(merged_list[[i]][5] == merged_list[[i]][8],"", "FALSE")))
 names(merged_list[[i]])[12] <- "PSTool.x == ICIS.y"
 merged_list[[i]][12][is.na(merged_list[[i]][12])] <- "CHECK"
}

for (i in parameter2){
 levels <- c("NPDES ID","Outfall","Period")
 merged_list[[i]] <- full_join(PS_list[[i]], ICIS_data[[i]], by=levels)
 merged_list[[i]] <- merged_list[[i]] %>% mutate(new.column = "ICIS VALUE")
 names(merged_list[[i]])[12] <- paste(i,"_SOURCE", sep="")
 merged_list[[i]] <- merged_list[[i]] %>%
   mutate(ifelse(merged_list[[i]][5] != merged_list[[i]][8],
                 ifelse(merged_list[[i]][5] != merged_list[[i]][8] &
                          merged_list[[i]][5] == merged_list[[i]][9],"WEEKLY",
                        ifelse(is.na(merged_list[[i]][8]) &
                                 merged_list[[i]][5] == merged_list[[i]][9],
                               "WEEKLY","FALSE")),""))
 names(merged_list[[i]])[13] <- "PSTool.x == ICIS.y"
 merged_list[[i]][13][is.na(merged_list[[i]][13])] <- "CHECK"}

# reorder list
merged_list <- merged_list[c(1,7,2,3,4,10,5,8,9,6)]
```

Writing edited tables into one Excel file

Once the ICIS and PS TOOL TABLE information were merged by parameter, the tables have to be written in separate tabs within an excel file.

```
#first, the replacement table needs to be added to merged_list

final_list <- append(merged_list,list(PS_data),after = 0)
names(final_list)[1] <- "Replacement Table"

OUT <- createWorkbook()

# add each dataframe from the final_list to a tab in the workbook, naming them by parameter
lapply(names(final_list), function(x){
  addWorksheet(OUT,x)
  writeData(OUT, x, x = final_list[[x]])})

saveWorkbook(OUT, paste0("Output_data/",fy,
                         "/Annual_Reports/FY",fy,"_InsigMun_ReplacementTable.xlsx"))
```

STOP HERE AND CHECK OUTPUT EXCEL FILE!!!

QA/QC the output file to make sure everything looks good. Create a log of the issues found (InsMun_Data_Mismatches_Log_FY20XX). Change column L or M (if parameter goes through the Weekly check) if flagged data was checked and no issues with the PS tool data was found (CHECKED), and if data is typically not reported (NO REPORTING). Any manual entries in the Excel file should be written in read to identify manually edited data. If the reason for data mismatch could not be identified, highlight the cells in yellow and consult permitting or compliance for clarification. If mismatch assessment concluded that ICIS value is correct and PS tool value needs to be replaced, no manual change is necessary because the R code will take care of that issue in the upcoming code chunks.

Delete additional facilities (bottom rows contain facilities only present in ICIS report outputs, no facility name in PS tool column B and usually an atypical outfall number) and any potential duplicates before starting the next step.

Adjusting PS tool data

Clean the environment before starting this next step and set the fiscal year again.

In this step, based on the content of the QAQCed file, the information on the replacement table will be updated.

If the NODI Code is not NA: the value replaced with zero and the SOURCE column is changed into the NODI description. If check column has "WEEKLY", "CHECK" or "FALSE", the value in the replacement table should be replaced with the ICIS value.

```
# Set up period again

fy <- 2023

# Load the adjusted data
read_excel_allsheets <- function(filename) {
  sheets <- excel_sheets(filename)
  x <-    lapply(sheets, function(X) readxl::read_excel(filename, sheet = X))
  names(x) <- sheets
  x
}

corrected_list <- read_excel_allsheets(
  paste0("Output_data/",fy,"/Annual_Reports/FY",fy,"_InsigMun_ReplacementTable.xlsx"))

names(corrected_list) <- c("ReplacementTable","flow","bod5","do","nh3","no23",
                           "tn","po4","tp","tss","ton")

list2env(corrected_list,envir=.GlobalEnv)

# FLOW
flow <- flow %>%
  subset(., `PSTool.x == ICIS.y` != "DELETE"|is.na(`PSTool.x == ICIS.y`))%>%
  mutate(FLOW.x = ifelse(is.na(`PSTool.x == ICIS.y`),FLOW.x,
                         ifelse(`PSTool.x == ICIS.y`== "CHECK" & !is.na(FLOW.y),
                                FLOW.y,
                                ifelse(`PSTool.x == ICIS.y`== "FALSE" &
```

```
                                  !is.na(FLOW.y),FLOW.y,FLOW.x))),
         FLOW.x= ifelse(is.na(`NODI Code`),FLOW.x,0),
         FLOW_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                              FLOW_SOURCE),
         check=ifelse(FLOW.x==FLOW.y,"","FALSE"))



# BOD5
bod5 <- bod5 %>%
  subset(., `PSTool.x == ICIS.y` != "DELETE"|is.na(`PSTool.x == ICIS.y`))%>%
  mutate(BOD5.x = ifelse(is.na(`PSTool.x == ICIS.y`),BOD5.x,
                         ifelse(`PSTool.x == ICIS.y`== "WEEKLY", BOD5.y,
                                ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                         !is.na(BOD5.y),BOD5.y,
                                       ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                         !is.na(BOD5.y),
                                         BOD5.y,BOD5.x)))),
         BOD5.x = ifelse(is.na(`NODI Code`),BOD5.x,0),
         BOD5_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                              BOD5_SOURCE),
         check=ifelse(BOD5.x==BOD5.y,"","FALSE"))

# DO
do <- do %>%
  subset(., `PSTool.x == ICIS.y` != "DELETE"|is.na(`PSTool.x == ICIS.y`))%>%
  mutate(DO.x = ifelse(is.na(`PSTool.x == ICIS.y`),DO.x,
                         ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                  !is.na(DO.y),DO.y,
                                ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                         !is.na(DO.y),
                                         DO.y,DO.x))),
         DO.x = ifelse(is.na(`NODI Code`),DO.x,0),
         DO_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                                   DO_SOURCE),
         check=ifelse(DO.x==DO.y,"", "FALSE"))

# NH3
nh3 <- nh3 %>%
  subset(., `PSTool.x == ICIS.y` != "DELETE"|is.na(`PSTool.x == ICIS.y`))%>%
  mutate(NH3.x = ifelse(is.na(`PSTool.x == ICIS.y`), NH3.x,
                        ifelse(`PSTool.x == ICIS.y`== "WEEKLY", NH3.y,
                                ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                         !is.na(NH3.y),NH3.y,
                                       ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                         !is.na(NH3.y),
                                         NH3.y,NH3.x)))),
         NH3.x = ifelse(is.na(`NODI Code`),NH3.x,0),
         NH3_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                             NH3_SOURCE),
         check=ifelse(NH3.x==NH3.y,"","FALSE"))

# NO23
no23 <- no23 %>%
  subset(., `PSTool.x == ICIS.y` != "DELETE"|is.na(`PSTool.x == ICIS.y`))%>%
  mutate(NO23.x = ifelse(is.na(`PSTool.x == ICIS.y`),NO23.x,
                         ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                  !is.na(NO23.y),NO23.y,
                                ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                         !is.na(NO23.y),
```

```
                                        NO23.y,NO23.x))),
         NO23.x = ifelse(is.na(`NODI Code`),NO23.x,0),
         NO23_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                              NO23_SOURCE),
         check=ifelse(NO23.x==NO23.y,"", "FALSE"))

# TN
tn <- tn %>%
  subset(., `PSTool.x == ICIS.y` != "DELETE"|is.na(`PSTool.x == ICIS.y`))%>%
  mutate(TN.x = ifelse(is.na(`PSTool.x == ICIS.y`),TN.x,
                        ifelse(`PSTool.x == ICIS.y`== "WEEKLY", TN.y,
                                ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                        !is.na(TN.y),
                                      TN.y,
                                      ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                        !is.na(TN.y),
                                        TN.y,TN.x)))),
         TN.x = ifelse(is.na(`NODI Code`),TN.x,0),
         TN_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                             TN_SOURCE),
         check=ifelse(TN.x==TN.y,"","FALSE"))

# PO4
po4 <- po4 %>%
  subset(., `PSTool.x == ICIS.y` != "DELETE"|is.na(`PSTool.x == ICIS.y`))%>%
  mutate(PO4.x = ifelse(is.na(`PSTool.x == ICIS.y`),PO4.x,
                         ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                 !is.na(PO4.y),PO4.y,
                                ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                        !is.na(PO4.y),
                                        PO4.y,PO4.x))),
         PO4.x = ifelse(is.na(`NODI Code`),PO4.x,0),
         PO4_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                             PO4_SOURCE),
         check=ifelse(PO4.x==PO4.y,"", "FALSE"))

# TP
tp <- tp %>%
  subset(., `PSTool.x == ICIS.y` != "DELETE"|is.na(`PSTool.x == ICIS.y`))%>%
  mutate(TP.x = ifelse(is.na(`PSTool.x == ICIS.y`),TP.x,
                       ifelse(`PSTool.x == ICIS.y`== "WEEKLY", TP.y,
                               ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                        !is.na(TP.y),TP.y,
                                     ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                        !is.na(TP.y),
                                        TP.y,TP.x)))),
         TP.x = ifelse(is.na(`NODI Code`),TP.x,0),
         TP_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                            TP_SOURCE),
         check=ifelse(TP.x==TP.y,"","FALSE"))

# TSS
tss <- tss %>%
  subset(., `PSTool.x == ICIS.y` != "DELETE"|is.na(`PSTool.x == ICIS.y`))%>%
  mutate(TSS.x = ifelse(is.na(`PSTool.x == ICIS.y`),TSS.x,
                        ifelse(`PSTool.x == ICIS.y`== "WEEKLY", TSS.y,
                                ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                        !is.na(TSS.y),
                                      TSS.y,
```

```
                                       ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                         !is.na(TSS.y),
                                         TSS.y,TSS.x)))),
         TSS.x = ifelse(is.na(`NODI Code`),TSS.x,0),
         TSS_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                             TSS_SOURCE),
         check=ifelse(TSS.x==TSS.y,"","FALSE"))

# TON
ton <- ton %>%
  subset(., `PSTool.x == ICIS.y` != "DELETE"|is.na(`PSTool.x == ICIS.y`))%>%
  mutate(TON.x = ifelse(is.na(`PSTool.x == ICIS.y`),TON.x,
                        ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                 !is.na(TON.y),TON.y,
                               ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                        !is.na(TON.y),
                                        TON.y,TON.x))),
         TON.x = ifelse(is.na(`NODI Code`),TON.x,0),
         TON_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                             TON_SOURCE),
         TON_SOURCE = ifelse(TON.x<0,"Negative value corrected",
                             TON_SOURCE),
         TON.x = ifelse(TON.x<0,0,TON.x),
         check=ifelse(TON.x==TON.y,"","FALSE"))


###############################################################################

# Replace values in replacement table and put all dataframes in a list

FinalTable <- ReplacementTable %>%
  mutate(FLOW=flow$FLOW.x,
         FLOWSource=flow$FLOW_SOURCE,
         BOD5=bod5$BOD5.x,
         BOD5Source=bod5$BOD5_SOURCE,
         DO=do$DO.x,
         DOSource=do$DO_SOURCE,
         NH3=nh3$NH3.x,
         NH3Source=nh3$NH3_SOURCE,
         NO23=no23$NO23.x,
         NO23Source=no23$NO23_SOURCE,
         TN=tn$TN.x,
         TNSource=tn$TN_SOURCE,
         PO4=po4$PO4.x,
         PO4Source=po4$PO4_SOURCE,
         TP=tp$TP.x,
         TPSource=tp$TP_SOURCE,
         TSS=tss$TSS.x,
         TSSSource=tss$TSS_SOURCE,
         TON=ton$TON.x,
         TONSource=ton$TON_SOURCE
         )

# Final edited table is included in the Excel output
df_list <- list(ReplacementTable,FinalTable, flow,bod5,do,nh3,no23,tn,po4,
                tp,tss,ton)
names(df_list) <-c("ReplacementTable","FinalTable","FLOW","BOD5","DO","NH3",
                   "NO23","TN","PO4","TP","TSS","TON")
```

```
# Save file
OUT <- createWorkbook()

lapply(names(df_list), function(x){
  addWorksheet(OUT,x)
  writeData(OUT, x, x = df_list[[x]])})

saveWorkbook(OUT, paste0("Output_data/",fy, "/Annual_Reports/FY", fy,
                         "_InsigMun_ReplacementTable_Final.xlsx"),
             overwrite=TRUE)
```

QA/QC the output file to make sure everything looks good and there are no outstanding issues in each constituent's tab.

Calculated values

Missing TN replaced with NH3+NO23+TON

In cases where facility does not report TN, but reports one or more nitrogen constituents (NH3, NO23, TON), we replace the missing TN value with calculated value and Source gets updated.

```
# Load table in R and calculate field
Final_Table <- read_excel(
  paste0("Output_data/",fy, "/Annual_Reports/FY", fy,
                         "_InsigMun_ReplacementTable_Final.xlsx"),
  sheet = 2)

Final_Table_TN <- Final_Table[c(4,5,8,10,14,15,20,16)]

Final_Table_TN <- Final_Table_TN %>%
  mutate(TN_calculated = NH3+NO23+TON,
         TN_missing = ifelse(TN ==0 & (NH3!=0|TON!=0|NO23!=0),
                             NH3+NO23+TON,""),
         TN_final = pmax(TN,TN_calculated),
         TN_Source = ifelse(pmax(TN,TN_calculated)==TN,"ICIS VALUE",
                            "Calculated as NH3+NO23+TON"))

# Write resulting table in a new tab of the Excel file
wb <- loadWorkbook(paste0("Output_data/",fy, "/Annual_Reports/FY", fy,
                         "_InsigMun_ReplacementTable_Final.xlsx"))

addWorksheet(wb,"TNcalculated")

DF <- data.frame(Final_Table_TN)

writeData(wb,"TNcalculated",DF)

saveWorkbook(wb,
             paste0("Output_data/",fy, "/Annual_Reports/FY", fy,
                         "_InsigMun_ReplacementTable_Final.xlsx"),
             overwrite = TRUE)
```

QA/QC the resulting tab in the file (TNcalculated tab) to make sure everything looks good. Carefully look into data records where reported TN is much different from calculated. Take notes in the "InsigMun_Data_Mismatches_Log_FY20XX" and share cases that need ICIS editing with compliance.

PO4 over Total Phosphorus

In some instances, PO4 concentrations recorded are higher than total phosphorus concentrations due to the frequency each is required to be analyzed. Typically, TP is sampled at a higher frequency than PO4 and should reflect more accurately the data. To adjust the instances where PO4 concentration is higher than TP, we run a code to replace the TP with PO4 concentrations when [PO4] >[TP].

```
# Load table in R and add the check columns
PO4_Over_TP <- read_excel(paste0("Output_data/",fy, "/Annual_Reports/FY", fy,
                         "_InsigMun_ReplacementTable_Final.xlsx"),
  sheet = 2)

PO4_Over_TP <- PO4_Over_TP[c(4,5,8,10,17,18,28)]

PO4_Over_TP <- PO4_Over_TP %>%
      mutate(PO4_Over_TP = ifelse(PO4 > TP, TP, PO4),
             PO4_Source = ifelse(PO4 > TP, "EQUALLED TO TP", PO4Source))

# Write resulting table in a new tab of the Excel file

wb <- loadWorkbook(paste0("Output_data/",fy, "/Annual_Reports/FY", fy,
                         "_InsigMun_ReplacementTable_Final.xlsx"))

addWorksheet(wb,"PO4 Over TP")

DF <- data.frame(PO4_Over_TP)

writeData(wb,"PO4 Over TP",DF)

saveWorkbook(wb,paste0("Output_data/",fy, "/Annual_Reports/FY", fy,
                         "_InsigMun_ReplacementTable_Final.xlsx"),
             overwrite = TRUE)
```

QA/QC the resulting tab in the file (PO4 Over TP tab) to make sure everything looks good. Carefully look into data records where PO4 concentration is higher than TP. In most cases, TP is sampled more frequently and that is why the results differ. Check Discharge permits to see if that is the case. Take notes of any incorrect data "InsigMun_Data_Mismatches_Log_FY20XX" and share cases that need ICIS editing with compliance.

Replace PO4 and TN correct values

Before calculating TOP from TP and PO4, we have to replace the PO4 values we corrected from the previous step (by changing the cases where PO4 was higher than TP to TP concentrations) into the FinalTable.

```
Final_Table <- read_excel(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                                 "_InsigMun_ReplacementTable_Final.xlsx"),
                          sheet = 2)
```

```
# added the part below because sometimes changes are made in TN or PO4
ed_TN <- read_excel(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                           "_InsigMun_ReplacementTable_Final.xlsx"),
                    sheet = 13)

ed_PO4 <- read_excel(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                            "_InsigMun_ReplacementTable_Final.xlsx"),
                     sheet = 14)

# Edit final table
Final_Table <- Final_Table %>%
  mutate(
    TN = ed_TN$TN_final,
    TNSource = ed_TN$TN_Source,
    PO4 = ed_PO4$PO4_Over_TP,
    PO4Source = ed_PO4$PO4_Source)

# Replace "FinalTable" tab with this one

wb  <- loadWorkbook(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                           "_InsigMun_ReplacementTable_Final.xlsx"))

writeData(wb, sheet="FinalTable", Final_Table)

saveWorkbook(wb,paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                           "_InsigMun_ReplacementTable_Final.xlsx"),
             overwrite = TRUE)
```

Total Organic Phosphorus

Last tab that needs to be added to the file is the Total Organic Phosphorus (TOP), calculated by subtracting PO4 from TP.

```
# Load table in R and include TOP column with appropriate control checks
Final_Table_TOP <- read_excel(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                                     "_InsigMun_ReplacementTable_Final.xlsx"),
                              sheet = 2)

Final_Table_TOP <- Final_Table_TOP[c(4,5,8,10,17,18)]

Final_Table_TOP <- Final_Table_TOP %>%
    mutate(TOP = TP-PO4,
           TOP_Source = "Calculated as TP - PO4")

# Write final table into new tab of the Excel file
wb <- loadWorkbook(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                          "_InsigMun_ReplacementTable_Final.xlsx"))

addWorksheet(wb,"TOP")

DF <- data.frame(Final_Table_TOP)

writeData(wb,"TOP",DF)

saveWorkbook(wb,
```

```
            paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                   "_InsigMun_ReplacementTable_Final.xlsx"),
            overwrite = TRUE)
```

QA/QC the resulting tab in the file (TOP tab) to make sure everything looks good.

Adding final values to the PS table

```
top <- read_excel(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_InsigMun_ReplacementTable_Final.xlsx"),
                  sheet = 15)

Final_Table <- read_excel(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                                 "_InsigMun_ReplacementTable_Final.xlsx"),
                          sheet = 2) %>%
  mutate(TOP = top$TOP,
         TOPSource = top$TOP_Source
         )

# Replace "FinalTable" tab with this one

wb  <- loadWorkbook(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                           "_InsigMun_ReplacementTable_Final.xlsx"))

writeData(wb, sheet="FinalTable", Final_Table)

saveWorkbook(wb,paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                       "_InsigMun_ReplacementTable_Final.xlsx"),
             overwrite = TRUE)
```

APPENDIX C - Significant Industrial Annual Reports

2023-11-14

Intro

This code provides the steps to compare the information available in the Point Source tool and ICIS and generate a report for each fiscal year regarding pollutant levels in treated wastewater discharges from industrial facilities with significant discharges.

```
# if any of the packages has not been installed yet, remove "#" preceding function
#install.packages("readxl")
#install.packages("tidyverse")
#install.packages("openxlsx")

library("readxl")
library("tidyverse")
library("openxlsx")
```

Editable code

Some of the code needs to be edited each fiscal year. The information that needs editing is:

fy –> fiscal year

```
#Set up period fiscal year below:

fy <- 2023
```

Data

The files to be imported include the table exported from the PS tool (and renamed as "PS TOOL TABLE IND SIG.xlsx"), and ICIS reports for each constituent (BOD5, DO, FLOW, NH3, NO23, PO4, TN, TON, TP, TSS), which should already be named appropriately due to the name of the reports themselves. Once files are downloaded from the web applications, save them in the 'Input_files' folder of the project in their respective year and annual reports folder.

```
# Set working directory to project location, not to the file location.
getwd()

PS_data <- read_excel(
  paste0("Input_data/",fy,"/Annual_Reports/PS TOOL TABLE IND SIG.xlsx"),
  sheet = 2)
```

Formatting ICIS tables

The first step of the process is format each table obtained from ICIS to have the columns equivalent to PS TOOL TABLE.

The process will repeat through a for loop for every parameter.

```
# create parameter lists
parameter1 <- c("FLOW",  "DO", "NH3", "NO23", "PO4", "TON")
```

```
parameter2 <- c("BOD5","TP", "TSS", "TN") # added weekly vs. monthly check


ICIS_data <- list()

# for loop to generate each parameter table (ICIS)
for(i in parameter1){
  ICIS_data[[i]] <- read_excel(paste0("Input_data/",fy,"/Annual_Reports/", i,
                                       " ", "IND SIG ICIS.xlsx"), sheet=1)
  ICIS_data[[i]] <- ICIS_data[[i]][-c(5,8)]
  names(ICIS_data[[i]]) <- c("NPDES ID","Facility","Outfall","Period",
                             "Description",i,"NODI Code","NODI Description")
  ICIS_data[[i]] <- drop_na(ICIS_data[[i]], i)
  ICIS_data[[i]][6] <- apply(ICIS_data[[i]][6], 2,
                    function(x) as.numeric(as.character(gsub("<","",x))))
  ICIS_data[[i]][6][is.na(ICIS_data[[i]][6])] <- 0
  ICIS_data[[i]]
}

# for loop to generate other set of parameters tables
for(j in parameter2){
  ICIS_data[[j]] <- read_excel(paste0("Input_data/",fy,"/Annual_Reports/", j,
                                       " ", "IND SIG ICIS.xlsx"), sheet=1)
  ICIS_data[[j]] <- ICIS_data[[j]][-c(5,8,10)]
  names(ICIS_data[[j]]) <- c("NPDES ID","Facility","Outfall","Period",
                             "Description",j,paste0(j,"_weekly"),"NODI Code",
                             "NODI Description")
  ICIS_data[[j]] <- ICIS_data[[j]] %>%
    mutate(Description = "MX AV/MO AVG")
  ICIS_data[[j]][c(6,7)] <- apply(ICIS_data[[j]][c(6,7)], 2,
                    function(x) as.numeric(as.character(gsub("<","",x))))
  ICIS_data[[j]] <- ICIS_data[[j]] %>%
  group_by(`NPDES ID`,`Facility`,`Outfall`,
           `Period`,
           `Description`) %>%
  summarize(across(where(is.numeric), ~.x[!is.na(.x)][1]),
            across(where(is.character),~.x[!is.na(.x)][1]))
  }
```

Split PS tool by constituents

In the case of municipal insignificant, there is no need to select a specific period of the PS tool table since the entire fiscal year is used in the comparison. A list of dataframes is created containing each constituent information in a different dataframe.

```
PS_list <- list()

parameter <- c("FLOW","BOD5","DO","NH3","NO23","PO4","TP","TSS","TON","TN")

for (i in parameter){
  f <- c("NPDES ID", "Facility", "Outfall", "Period", i)
  PS_list[[i]] <- select(PS_data, all_of(f))
  PS_list[[i]]
}
```

Combining both ICIS and PS tables for each individual parameter

```
merged_list <- list()


# with Weekly issues for certain parameters
for (i in parameter1){
 levels <- c("NPDES ID","Outfall","Period")
 merged_list[[i]] <- full_join(PS_list[[i]], ICIS_data[[i]], by=levels)
 merged_list[[i]] <- merged_list[[i]] %>% mutate(new.column = "ICIS VALUE")
 names(merged_list[[i]])[11] <- paste(i,"_SOURCE", sep="")
 merged_list[[i]] <- merged_list[[i]] %>% mutate(
   ifelse(!is.na(merged_list[[i]][5])&is.na(merged_list[[i]][8]),"FALSE",
          ifelse(merged_list[[i]][5] == merged_list[[i]][8],"", "FALSE")))
 names(merged_list[[i]])[12] <- "PSTool.x == ICIS.y"
 merged_list[[i]][12][is.na(merged_list[[i]][12])] <- "CHECK"
}

for (i in parameter2){
 levels <- c("NPDES ID","Outfall","Period")
 merged_list[[i]] <- full_join(PS_list[[i]], ICIS_data[[i]], by=levels)
 merged_list[[i]] <- merged_list[[i]] %>% mutate(new.column = "ICIS VALUE")
 names(merged_list[[i]])[12] <- paste(i,"_SOURCE", sep="")
 merged_list[[i]] <- merged_list[[i]] %>%
   mutate(ifelse(merged_list[[i]][5] != merged_list[[i]][8],
                 ifelse(merged_list[[i]][5] != merged_list[[i]][8] &
                          merged_list[[i]][5] == merged_list[[i]][9],"WEEKLY",
                        ifelse(is.na(merged_list[[i]][8]) &
                                 merged_list[[i]][5] == merged_list[[i]][9],
                               "WEEKLY","FALSE")),""))
 names(merged_list[[i]])[13] <- "PSTool.x == ICIS.y"
 merged_list[[i]][13][is.na(merged_list[[i]][13])] <- "CHECK"}

# reorder list
merged_list <- merged_list[c(1,7,2,3,4,10,5,8,9,6)]
```

Writing edited tables into one Excel file

Once the ICIS and PS TOOL TABLE information were merged by parameter, the tables have to be written in separate tabs within an excel file.

```
#first, the replacement table needs to be added to merged_list

final_list <- append(merged_list,list(PS_data),after = 0)
names(final_list)[1] <- "Replacement Table"

OUT <- createWorkbook()

# add each dataframe from the final_list to a tab in the workbook, naming them by parameter
lapply(names(final_list), function(x){
  addWorksheet(OUT,x)
  writeData(OUT, x, x = final_list[[x]])})

saveWorkbook(OUT, paste0("Output_data/",fy,
                         "/Annual_Reports/FY",fy,"_SigInd_ReplacementTable.xlsx"))
```

STOP HERE AND CHECK OUTPUT EXCEL FILE!!!

QA/QC the output file to make sure everything looks good. Create a log of the issues found. Change column L if flagged data was checked and no issues with PS tool data was found (CHECKED), and if data was not reported (NOT REPORTED). Any value or NODI code change should be written in red to identify manually edited data. If the reason for data mismatch could not be identified, highlight the cells in yellow. If mismatch assessment concluded that ICIS value is correct and PS tool value needs to be replaced, no manual change is necessary because the R code will take care of that issue in the upcoming code chunks.

Adjusting PS tool data

Clean the environment before starting this next step.

In this step, based on the output from the previous steps, the information on the replacement table will be updated.

If NODI Code is not NA: value replaced with zero and SOURCE column changed to NODI description. If check column has "WEEKLY", "CHECK" or "FALSE", value in the replacement table should be replaced with ICIS value.

```
# Set up period again

fy <- 2023

# Load the adjusted data
read_excel_allsheets <- function(filename) {
  sheets <- excel_sheets(filename)
  x <-    lapply(sheets, function(X) readxl::read_excel(filename, sheet = X))
  names(x) <- sheets
  x
}

corrected_list <- read_excel_allsheets(
  paste0("Output_data/",fy,"/Annual_Reports/FY",fy,"_SigInd_ReplacementTable.xlsx"))

names(corrected_list) <- c("ReplacementTable","flow","bod5","do","nh3","no23",
                           "tn","po4","tp","tss","ton")

list2env(corrected_list,envir=.GlobalEnv)

# FLOW
flow <- flow %>%
  subset(., `PSTool.x == ICIS.y` != "DELETE"|is.na(`PSTool.x == ICIS.y`))%>%
  mutate(FLOW.x = ifelse(is.na(`PSTool.x == ICIS.y`),FLOW.x,
                         ifelse(`PSTool.x == ICIS.y`== "CHECK" & !is.na(FLOW.y),
                                FLOW.y,
                                ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                         !is.na(FLOW.y),FLOW.y,FLOW.x))),
         FLOW.x= ifelse(is.na(`NODI Code`),FLOW.x,0),
         FLOW_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                              FLOW_SOURCE),
         check=ifelse(FLOW.x==FLOW.y,"","FALSE"))
```

```
# BOD5
bod5 <- bod5 %>%
  mutate(BOD5.x = ifelse(is.na(`PSTool.x == ICIS.y`),BOD5.x,
                         ifelse(`PSTool.x == ICIS.y`== "WEEKLY", BOD5.y,
                                ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                         !is.na(BOD5.y),BOD5.y,
                                       ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                         !is.na(BOD5.y),
                                         BOD5.y,BOD5.x)))),
         BOD5.x = ifelse(is.na(`NODI Code`),BOD5.x,0),
         BOD5_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                              BOD5_SOURCE),
         check=ifelse(BOD5.x==BOD5.y,"","FALSE"))

# DO
do <- do %>%
  mutate(DO.x = ifelse(is.na(`PSTool.x == ICIS.y`),DO.x,
                         ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                  !is.na(DO.y),DO.y,
                                ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                         !is.na(DO.y),
                                         DO.y,DO.x))),
         DO.x = ifelse(is.na(`NODI Code`),DO.x,0),
         DO_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                                   DO_SOURCE),
         check=ifelse(DO.x==DO.y,"", "FALSE"))

#NH3
nh3 <- nh3 %>%
  mutate(NH3.x = ifelse(is.na(`PSTool.x == ICIS.y`), NH3.x,
                        ifelse(`PSTool.x == ICIS.y`== "WEEKLY", NH3.y,
                                ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                         !is.na(NH3.y),NH3.y,
                                       ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                         !is.na(NH3.y),
                                         NH3.y,NH3.x)))),
         NH3.x = ifelse(is.na(`NODI Code`),NH3.x,0),
         NH3_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                             NH3_SOURCE),
         check=ifelse(NH3.x==NH3.y,"","FALSE"))

#NO23
no23 <- no23 %>%
  mutate(NO23.x = ifelse(is.na(`PSTool.x == ICIS.y`),NO23.x,
                         ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                  !is.na(NO23.y),NO23.y,
                                ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                         !is.na(NO23.y),
                                         NO23.y,NO23.x))),
         NO23.x = ifelse(is.na(`NODI Code`),NO23.x,0),
         NO23_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                              NO23_SOURCE),
         check=ifelse(NO23.x==NO23.y,"", "FALSE"))

#TN
tn <- tn %>%
  subset(., `PSTool.x == ICIS.y` != "DELETE"|is.na(`PSTool.x == ICIS.y`))%>%
  mutate(TN.x = ifelse(is.na(`PSTool.x == ICIS.y`),TN.x,
                        ifelse(`PSTool.x == ICIS.y`== "WEEKLY", TN.y,
```

```
                                ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                         !is.na(TN.y),
                                       TN.y,
                                       ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                         !is.na(TN.y),
                                         TN.y,TN.x)))),
         TN.x = ifelse(is.na(`NODI Code`),TN.x,0),
         TN_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                             TN_SOURCE),
         check=ifelse(TN.x==TN.y,"","FALSE"))

#PO4
po4 <- po4 %>%
  mutate(PO4.x = ifelse(is.na(`PSTool.x == ICIS.y`),PO4.x,
                         ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                  !is.na(PO4.y),PO4.y,
                                ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                         !is.na(PO4.y),
                                         PO4.y,PO4.x))),
         PO4.x = ifelse(is.na(`NODI Code`),PO4.x,0),
         PO4_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                             PO4_SOURCE),
         check=ifelse(PO4.x==PO4.y,"", "FALSE"))

#TP
tp <- tp %>%
  mutate(TP.x = ifelse(is.na(`PSTool.x == ICIS.y`),TP.x,
                       ifelse(`PSTool.x == ICIS.y`== "WEEKLY", TP.y,
                               ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                         !is.na(TP.y),TP.y,
                                      ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                         !is.na(TP.y),
                                         TP.y,TP.x)))),
         TP.x = ifelse(is.na(`NODI Code`),TP.x,0),
         TP_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                            TP_SOURCE),
         check=ifelse(TP.x==TP.y,"","FALSE"))

#TSS
tss <- tss %>%
  mutate(TSS.x = ifelse(is.na(`PSTool.x == ICIS.y`),TSS.x,
                        ifelse(`PSTool.x == ICIS.y`== "WEEKLY", TSS.y,
                                ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                         !is.na(TSS.y),
                                       TSS.y,
                                       ifelse(`PSTool.x == ICIS.y`== "FALSE" &
                                         !is.na(TSS.y),
                                         TSS.y,TSS.x)))),
         TSS.x = ifelse(is.na(`NODI Code`),TSS.x,0),
         TSS_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                             TSS_SOURCE),
         check=ifelse(TSS.x==TSS.y,"","FALSE"))

#TON
ton <- ton %>%
  mutate(TON.x = ifelse(is.na(`PSTool.x == ICIS.y`),TON.x,
                         ifelse(`PSTool.x == ICIS.y`== "CHECK" &
                                  !is.na(TON.y),TON.y,
                                ifelse(`PSTool.x == ICIS.y`== "FALSE" &
```

```
                                          !is.na(TON.y),
                                          TON.y,TON.x))),
         TON.x = ifelse(is.na(`NODI Code`),TON.x,0),
         TON_SOURCE = ifelse(!is.na(`NODI Code`),`NODI Description`,
                             TON_SOURCE),
         TON_SOURCE = ifelse(TON.x<0,"Negative value corrected",
                             TON_SOURCE),
         TON.x = ifelse(TON.x<0,0,TON.x),
         check=ifelse(TON.x==TON.y,"","FALSE"))


#########

# Replace values in replacement table and put all dataframes in a list

FinalTable <- ReplacementTable %>%
  mutate(FLOW=flow$FLOW.x,
         FLOWSource=flow$FLOW_SOURCE,
         BOD5=bod5$BOD5.x,
         BOD5Source=bod5$BOD5_SOURCE,
         DO=do$DO.x,
         DOSource=do$DO_SOURCE,
         NH3=nh3$NH3.x,
         NH3Source=nh3$NH3_SOURCE,
         NO23=no23$NO23.x,
         NO23Source=no23$NO23_SOURCE,
         TN=tn$TN.x,
         TNSource=tn$TN_SOURCE,
         PO4=po4$PO4.x,
         PO4Source=po4$PO4_SOURCE,
         TP=tp$TP.x,
         TPSource=tp$TP_SOURCE,
         TSS=tss$TSS.x,
         TSSSource=tss$TSS_SOURCE,
         TON=ton$TON.x,
         TONSource=ton$TON_SOURCE
         )

df_list <- list(ReplacementTable,FinalTable, flow,bod5,do,nh3,no23,tn,po4,
                tp,tss,ton)
names(df_list) <-c("ReplacementTable","FinalTable","FLOW","BOD5","DO","NH3",
                   "NO23","TN","PO4","TP","TSS","TON")



# Save file
OUT <- createWorkbook()

lapply(names(df_list), function(x){
  addWorksheet(OUT,x)
  writeData(OUT, x, x = df_list[[x]])})

saveWorkbook(OUT, paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"),
             overwrite=TRUE)
```

QA/QC the output file to make sure everything looks good and there are no outstanding issues in each constituent's tab.

Calculated values

Nitrogen Constituents

If facility did not report TN or its constituents (NH3, NO23, TON), we calculate those values based on the following formula: TN = 0.17xNH3 + 0.48xNO23 + 0.35xTON.

```
# Load table in R and calculate field
Final_Table <- read_excel(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"),
                         sheet = 2)

Final_Table_TN <- Final_Table[c(4,5,8,10,14,15,20,16,25,26,31,27)]

Final_Table_TN <- Final_Table_TN %>%
  mutate(
    NH3_calc = ifelse(is.na(NH3)&is.na(NO23),TN*0.17,
                        ifelse(is.na(NH3)&!is.na(NO23),(TN-NO23)*0.33, NH3)),
    NO23_calc = ifelse(is.na(NO23),TN*0.48,NO23),
    TON_calc = ifelse(is.na(TON)&is.na(NO23),TN*0.35,
                        ifelse(is.na(TON)&!is.na(NO23),(TN-NO23)*0.67,TON)),
    TN_calc = NH3_calc+NO23_calc+TON_calc,
    NH3_SOURCE = ifelse(is.na(NH3)&is.na(NO23),
                        "Used Formula TN = 0.17 NH3 + 0.35 TON + 0.48 NO23",
                        ifelse(is.na(NH3)&!is.na(NO23),"NH3=(TN-NO23)*0.33",
                               NH3Source)),
    NO23_SOURCE = ifelse(is.na(NO23),
                         "Used Formula TN = 0.17 NH3 + 0.35 TON + 0.48 NO23",
                         NO23Source),
    TON_SOURCE = ifelse(is.na(TON)&is.na(NO23),
                        "Used Formula TN = 0.17 NH3 + 0.35 TON + 0.48 NO23",
                        ifelse(is.na(TON)&!is.na(NO23),"TON=(TN-NO23)*0.67",
                               TONSource)),
    TN_final = pmax(TN,TN_calc, na.rm=TRUE),
    TN_SOURCE = ifelse(pmax(TN,TN_calc)==TN,TNSource,
                        "Calculated as NH3+NO23+TON"),
    TN_SOURCE = ifelse(is.na(TN_SOURCE),"Calculated as NH3+NO23+TON",
                       TN_SOURCE)
        )

# Write resulting table in a new tab of the Excel file
wb <- loadWorkbook(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"))

addWorksheet(wb,"Nitrogen_calc")

DF <- data.frame(Final_Table_TN)

writeData(wb,"Nitrogen_calc",DF)

saveWorkbook(wb,
             paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"),
             overwrite = TRUE)
```

PO4 over Total Phosphorus

In some instances, PO4 concentrations recorded are higher than total phosphorus concentrations due to the frequency each is required to be analyzed. Typically, TP is sampled at a higher frequency than PO4 and should reflect more accurately the data. To adjust the instances where PO4 concentration is higher than TP, we run a code to replace the TP with PO4 concentrations when [PO4] >[TP].

```
# Load table in R and add the check columns
PO4_Over_TP <- read_excel(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"),
  sheet = 2)

PO4_Over_TP <- PO4_Over_TP[c(4,5,8,10,17,18,28)]

PO4_Over_TP <- PO4_Over_TP %>%
      mutate(PO4_Over_TP = ifelse(PO4 > TP, TP, PO4)) %>%
      mutate(PO4_Source = ifelse(PO4 > TP, "EQUALLED TO TP", PO4Source))

# Write resulting table in a new tab of the Excel file

wb <- loadWorkbook(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"))

addWorksheet(wb,"PO4 Over TP")

DF <- data.frame(PO4_Over_TP)

writeData(wb,"PO4 Over TP",DF)

saveWorkbook(wb,
             paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"),
             overwrite = TRUE)
```

QA/QC the resulting tab in the file (PO4 Over TP tab) to make sure everything looks good.

Correct Nitrogen and PO4 values

```
Final_Table <- read_excel(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"),
                          sheet = 2)

# added the part below because sometimes changes are made in TN or PO4
ed_TN <- read_excel(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"),
                    sheet = 13)

ed_PO4 <- read_excel(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"),
                     sheet = 14)

# Edit final table
Final_Table <- Final_Table %>%
  mutate(
```

```
    NH3 = ed_TN$NH3_calc,
    NH3Source = ed_TN$NH3_SOURCE,
    NO23 = ed_TN$NO23_calc,
    NO23Source = ed_TN$NO23_SOURCE,
    TON = ed_TN$TON_calc,
    TONSource = ed_TN$TON_SOURCE,
    TN = ed_TN$TN_final,
    TNSource = ed_TN$TN_SOURCE,
    PO4 = ed_PO4$PO4_Over_TP,
    PO4Source = ed_PO4$PO4_Source)

# Replace "FinalTable" tab with this one

wb  <- loadWorkbook(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"))

writeData(wb, sheet="FinalTable", Final_Table)

saveWorkbook(wb,paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"),
             overwrite = TRUE)
```

Phosphorus Constituents

```
# Load table in R and include TOP column with appropriate control checks
Final_Table <- read_excel(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"),
                          sheet = 2)

Final_Table_TOP <- Final_Table[c(4,5,8,10,17,18,28,29)]

Final_Table_TOP <- Final_Table_TOP %>%
    mutate(
      PO4_calc = ifelse(is.na(PO4),TP*0.59,PO4),
      PO4_SOURCE = ifelse(is.na(PO4), "PO4 = TP*0.59", PO4Source),
      TOP = TP-PO4_calc,
      TOP_SOURCE = "Calculated as TP - PO4")

# Write final table into new tab of the Excel file
wb <- loadWorkbook(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"))

addWorksheet(wb,"TOP")

DF <- data.frame(Final_Table_TOP)

writeData(wb,"TOP",DF)

saveWorkbook(wb,
             paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"),
             overwrite = TRUE)
```

QA/QC the resulting tab in the file (TOP tab) to make sure everything looks good.

Adding final values to the PS table

```
top <- read_excel(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"),
                  sheet = 15)

Final_Table <- read_excel(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"),
                          sheet = 2) %>%
  mutate(
    PO4 = top$PO4_calc,
    PO4Source = top$PO4_SOURCE,
    TOP = top$TOP,
    TOPSource = top$TOP_SOURCE
        )

# Replace "FinalTable" tab with this one

wb  <- loadWorkbook(paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"))

writeData(wb, sheet="FinalTable", Final_Table)

saveWorkbook(wb,paste0("Output_data/",fy, "/Annual_Reports/FY",fy,
                         "_SigInd_ReplacementTable_Final.xlsx"),
             overwrite = TRUE)
```

APPENDIX D - Spray Irrigation Annual Reports

2023-10-26

Intro

This code uses data exported from ICIS and calculates total monthly flow and annual averages, as well as annual total nitrogen average concentrations for annual reporting to the Chesapeake Bay Program. Facilities are not required to report total phosphorus data, therefore the default value of 3.15 mg/L is reported instead.An updated list of active spray irrigation facilities is provided annually by the Groundwater Discharge Permits Division and NPDES IDs list is adjusted in the ICIS query to reflect the most up to date information.

```
# if any of the packages has not been installed yet, remove "#" preceding function
#install.packages("readxl")
#install.packages("tidyverse")
#install.packages("openxlsx")

library("readxl")
library("tidyverse")
library("openxlsx")

#turn off scientific notation
options(scipen = 999)
```

Editable Code

Some of the code needs to be edited each year. After this code chunk, no more editing of the code is necessary.

```
#Set up period dates below and run it:

year <- "2023"   # fiscal year
past_year <- "2022"  # previous fiscal year
```

Data Import

Only one file needs to be imported: ICIS SPRAY IRRIGATION.xlsx, which contains FLOW, TN, and TP reports. Once files is downloaded from ICIS, save it in the 'Input_files' folder of the project in its respective fiscal year, in the 'Annual_Reports' folder.

```
# Set working directory to project location, not to the file location.
# This can be checked by clicking on the arrow to the right of the `Knit` icon >
# 'Knit Directory' and make sure 'Project Directory' is selected

getwd()

read_excel_allsheets <- function(filename) {
  sheets <- excel_sheets(filename)
  x <-    lapply(sheets, function(X) readxl::read_excel(filename, sheet = X, skip=3))
  names(x) <- c("SI_FLOW", "SI_TN","SI_TP")
```

```
  x
}

arrange_list <- function(DF) {
  DF %>% arrange(`NPDES ID`,`Facility Name`,`Perm Feature ID`,
           `Monitoring Period End Date`,
           `Statistical Base Short Desc`)
}

SI_list <- read_excel_allsheets(
  paste0("Input_data/",year,"/Annual_Reports/ICIS SPRAY IRRIGATION.xlsx"))%>%
  map(.,arrange_list)

list2env(SI_list,envir=.GlobalEnv)
```

Formatting Tables

Facilities report their data using different statistics (e.g., monthly average, annual average, cumulative total, etc.) and, sometimes more than one. Monthly average is the prioritized statistical base whenever facilities report data in that format. A file named `STAT_CODES_SI.xlsx` was created from the data submitted in 2022 and only needs to be edited if inactive facilities must be removed from the list or new facilities have to be included. If there is a new facility with a spray irrigation permit, the statistical base short description the facility uses to report the parameter needs to be assessed and added to the file.

In this step, the flow unit is adjusted from gal/d to MGD, when applicable, and the total monthly flow is calculated. Additionally, if TN record has a qualifier code representing no discharge (NODI codes C - No Discharge or 2 - Operation Shutdown) associated with it, the result column is filled with zero, otherwise with NA (e.g., 9 - Conditional Monitoring). The step to fill with NAs is to have averages more representative of the actual concentrations in a facility's discharge over the fiscal year.

```
# FLOW
code_flow <- read_excel(
  paste0("Input_data/",year,"/Annual_Reports/STAT_CODES_SI.xlsx"),sheet=1)

SI_FLOW <- code_flow %>%
  inner_join(SI_FLOW,by=c("NPDES ID","Facility Name",
                          "Statistical Base Short Desc"))%>%
  .[,c(1,2,4:6,3,7:12)]%>%
  mutate(
    FLOW_MGD = ifelse(is.na(`Q2 DMR Value`),`Q1 DMR Value`,`Q2 DMR Value`),
    FLOW_MGD = as.numeric(ifelse(grepl("NODI",FLOW_MGD),0,FLOW_MGD)),
    FLOW_MGD_correct = ifelse(is.na(`Q1 DMR Value Unit Short Desc`),FLOW_MGD,
                              ifelse(`Q1 DMR Value Unit Short Desc`=="gal/d",
                              FLOW_MGD/1000000,FLOW_MGD)),
    FLOW_MGD_correct = ifelse(is.na(`Q2 DMR Value Unit Short Desc`),FLOW_MGD_correct,
                              ifelse(`Q2 DMR Value Unit Short Desc`=="gal/d",
                              FLOW_MGD/1000000,FLOW_MGD_correct)),
    DAYS = days_in_month(`Monitoring Period End Date`),
    FLOW_MGM = FLOW_MGD_correct*DAYS
  )
```

```
# TN
code_tn <- read_excel(
  paste0("Input_data/",year,"/Annual_Reports/STAT_CODES_SI.xlsx"),sheet=2)

SI_TN <- code_tn %>%
  inner_join(SI_TN,by=c("NPDES ID","Facility Name",
                        "Statistical Base Short Desc"))%>%
  .[,c(1,2,4:6,3,7:14)]%>%
  mutate(
    TN = ifelse(is.na(`C1 DMR Value`),`C2 DMR Value`,ifelse(
      is.na(`C1 DMR Value`)&is.na(`C2 DMR Value`),`C3 DMR Value`,
      `C1 DMR Value`)),
    TN = as.numeric(ifelse(TN=="NODI C"|TN=="NODI 2",0,
                           ifelse(TN=="NODI 9",NA,TN)))
  )

# TP not necessary since reporting is not required. Values will be filled with
# default value of 3.15 unless there was no discharge at the facility.
```

Final Tables

Monthly flow is added for total annual flow and total nitrogen is averaged for the average annual concentrations. The calculated values are appended to the list of facilities from the previous year. Details on new facilities are filled in the output Excel spreadsheet.

```
facility_info <- read_excel(paste0("Input_data/",past_year,
                                   "/Annual_Reports/Maryland FY ",past_year,
                                   " Spray Irrigation Table.xlsx"),
                            sheet=2)%>%
  rename(Notes=...18)

annual_flow <- SI_FLOW %>%
  group_by(`NPDES ID`,`Facility Name`)%>%
  summarise(TOTAL_FLOW=sum(FLOW_MGM))%>%
  mutate(FLOW_GY=TOTAL_FLOW*1000000)

avg_annual_tn <- SI_TN %>%
  group_by(`NPDES ID`,`Facility Name`)%>%
  summarise(AVG_TN = mean(TN,na.rm = TRUE))

spray_irrigation <- facility_info %>%
  full_join(.,annual_flow, by=join_by(`NPDES ID / FACILITY NAME`==`NPDES ID`))%>%
  full_join(.,avg_annual_tn, by=join_by(`NPDES ID / FACILITY NAME`==`NPDES ID`))

spray_irr_final <- spray_irrigation %>%
  mutate(
    Year = year,
    `Volume (gal/yr)`= FLOW_GY,
    `Volume (MGY)`= TOTAL_FLOW,
    `TN (mg/l)` = AVG_TN,
    `TP (mg/l)` = ifelse(`Volume (MGY)`== 0|is.na(`Volume (MGY)`),0,3.15)
  ) %>%
  .[,c(1:18)]

data_fields <- colnames(spray_irr_final[,c(1:4,6:9,11:17)])
```

```
ReadMe <- data.frame(`Data Field`=data_fields,
                     Required=c("Yes","Yes","Yes","No","Yes","Yes","Yes","Yes",
                                "No","No","No","No","No","Yes","Yes"),
                     Description=c("Year spray irrigation was applied",
                                   "State Identifier",
                                   "Values are Ag or Urban or RIB (rapid infiltration system).
Was liquid sprayed on agricultural crops (ag) or turfgrass (urban) or into a rapid infiltratio
n basin",
                                   "Used to identify the source of the spray irrigation. Not u
sed by the CBP but good for tracking purposes",
                                   "Location spray irrigation was applied. Can be submitted as
latitude+longitude or Land River Segment", NA,NA,
                                   "Total volume sprayed on to the land in gallons per year",
                                   "Nutrient data should be supplied as mg of nutrient per lit
er of spray irrigation. If TN and TP are supplied, individual nutrient species will be calcula
ted using defaults.",
                                   NA,NA,NA,NA,NA,NA))
```

Save Output

The output will contain a ReadMe tab, the final Spray Irrigation table, flow calculations, total annual flow by site, the adjusted TN values, and average annual TN.

```
spray_irr_list <- list(ReadMe,spray_irr_final,SI_FLOW,annual_flow,
                       SI_TN,avg_annual_tn)
tabs.names <- c("ReadMe","Spray Irrigation","FLOW Report Calculations",
                "TOTAL ANNUAL FLOW","TN Report Adjusted","AVG ANNUAL TN")
names(spray_irr_list) <- tabs.names

OUT <- createWorkbook()


# add each dataframe from 'spray_irr_list' to a tab in the workbook
lapply(names(spray_irr_list), function(x){
  addWorksheet(OUT,x)
  writeData(OUT, x, x = spray_irr_list[[x]])})

saveWorkbook(OUT, paste0("Output_data/",year,"/Annual_Reports/Maryland FY ",
year," Spray Irrigation Table.xlsx"))
```

Once output is saved, open in Excel and format to make it look nicer:

- ReadMe tab: merge cells C6:C8 and middle align, merge cells C10:C16 and middle align, adjust column widths to display all the text, wrap text in column C and align left, make first row text bold and add light gray background, and add borders.
- Spray Irrigation: adjust width of all columns to display text, make first row text bold, and format columns P and Q to display only two decimal places. Verify if any adjustments in the values need to be made, particularly based on the `Notes` column.
- Rest of the tabs: adjust columns widths to display all information and format date columns as short dates.

Make final adjustments or calculations in case information is missing. If a facility has flow above zero but no information on TN, TN should be calculated as the average of all facilities reporting Once everything looks correct, re-upload the final version and format table to match PS tool spray irrigation template.

Final Version for Submission

```
SI_final <- read_excel(paste0("Output_data/",year,
                              "/Annual_Reports/Maryland FY ",
                              year," Spray Irrigation Table.xlsx"),
                       sheet=2)%>%
  .[,c(1:4,6:9,11:17)]


write.xlsx(SI_final, paste0("Output_data/",year,
                              "/Annual_Reports/Maryland FY ",year,
                            " Spray Irrigation Table Final Submission.xlsx"))
```

APPENDIX E - Biosolids Calendar Year Annual Reports

2023-10-26

Intro

This code prepares biosolids data for annual submission to CBP. Because reporting to the respective MDE program happens on a calendar year basis and only annually, we are unable to provide data for the entire fiscal year. Instead, we have been submitting the previous calendar year data during each fiscal year (e.g., FY2023 submission will have data for calendar year 2022).

```
# if any of the packages has not been installed yet, remove "#" preceding function
#install.packages("readxl")
#install.packages("tidyverse")
#install.packages("openxlsx")
#install.packages("janitor")

library("readxl")
library("tidyverse")
library("openxlsx")
library("janitor")

#turn off scientific notation
options(scipen = 999,digits=10)

# Opposite of %in% function
`%not_in%` <- purrr::negate(`%in%`)
```

Editable Code

Some of the code needs to be edited each year. The information that needs editing is:

fy -> Fiscal year we are reporting.

cy -> Calendar year data represents

past.cy -> Previous calendar year submission

```
#Set up period dates below and run it:

fy <- "2023"   # fiscal year
cy <- "2022"   # calendar year or past fiscal year
past.cy <- "2021"  # previous calendar year submission

# Enter the name of the last WWTP in each Constituent report

lastB <- "Wicomico Shores WWTP" # Class B biosolids
lastUnst <- "Worton-Butlertown WWTP" # Un-Stabilized biosolids
```

The Sewage Sludge Utilization Permit Annual Report from Cedarville Lagoon is provided by the Land and Materials Administration, Biosolids Division in PDF format. The code below should be updated with the calendar year PDF information so the data can be used to calculate biosolids applied. Table 2 of the report contains the abbreviation if the county where biosolids were applied, and the wet and dry tons. Make sure to include all the information from the report.

```
cdville <- data.frame(county.applied = c("Prince George's, MD"),
                      dry.tons.applied = c(0.65))
```

Data Import

There are a few reports that must be downloaded from Maryland's TEMPO database as Excel files:

- In-State Generator Report -> `BIOSOLIDS_In_State.xls`
- Out-of-State Generator Report -> `BIOSOLIDS_Out_of_State.xls`
- Constituents Report Class B and Un-Stabilized -> `BIOSOLIDS_ConstituentsB.xls` and `BIOSOLIDS_ConstituentsUnst.xls`

All downloaded files should be saved in the `Input_data` folder, under the correct fiscal year and in `Annual_Reports` following the naming conventions above.

```
# Set working directory to project location, not to the file location.
# This can be checked by clicking on the arrow to the right of the `Knit` icon >
# 'Knit Directory' and make sure 'Project Directory' is selected

getwd()
```

Calculation Table

Using the Constituents report, we create a dataset to estimate percent wet versus dry and nutrient content, and average that per County. These averages, alongside the Sewage Sludge Utilized data, will be used to calculate discharged nutrient from biosolids per MD county.

The format of exported Excel files need adjusting in R. The vectors created in the code chunks below are strategic to take care of that.

```
# Import file and fix column names and data misalignment

constb <- read_excel(paste0("Input_data/",fy,
                            "/Annual_Reports/BIOSOLIDS_ConstituentsB.xls"),
                     skip=8) %>%
  remove_empty(which="rows")%>%
  rename("pH"= "...14", "...15"="pH",
         "TK\n\nmg/kg"="...24","...25"="TK\n\nmg/kg",
         "Nitrate\nNitrogen\nmg/kg"="...26","...27"="Nitrate\nNitrogen\nmg/kg",
         "Total\nCadmium\nmg/kg"="...29",
         "Total\nCopper\nmg/kg"="Total\nCadmium\nmg/kg",
         "...31"="Total\nCopper\nmg/kg",
         "Total\nNickel\nmg/kg"="...32",
         "Total\nLead\nmg/kg"="Total\nNickel\nmg/kg",
         "...35"="Total\nLead\nmg/kg",
         "Total\nZinc\nmg/kg"="...36",
         "Total\nMercury\nmg/kg"="Total\nZinc\nmg/kg",
```

```
        "Total\nArsenic\nmg/kg"="Total\nMercury\nmg/kg",
        "...40"="Total\nArsenic\nmg/kg",
        "Total\nMolybdenum\nmg/kg"="...41","...42"= "Total\nMolybdenum\nmg/kg",
        "Total\nSelenium\nmg/kg"="...44",
        "PCBs\n\nmg/kg"="Total\nSelenium\nmg/kg",
        "CaCO3\n\n%"="PCBs\n\nmg/kg","...50"="CaCO3\n\n%"
        )

# Vector to get empty columns
unnamed <- colnames(constb)%>%
  .[grepl('[0-9]',.)] %>%
      gsub("[^0-9]", "", .)%>%
  as.integer

# Filter out empty columns leaving only those with data
constb <- constb[,-unnamed]

# Get the row number of the last facility
nrow.lastB <- min(which(constb$Plant==lastB))

# Remove first row and end rows with annotation
constb <- constb %>%
  fill(everything(),.direction = "down") %>%
  .[c(2:nrow.lastB),]

# Function to retain odd numbered rows
row_odd <- seq_len(nrow(constb))%%2
constb.final <- constb[row_odd == 1,]
```

CHECK OUTPUT TO SEE IF EVERYTHING LOOKS GOOD. If TEMPO export format changes, the code above would need adjustments!

Repeat the same process for Un-Stabilized Biosolids facilities.

```
# Import file and fix column names and data misalignment

constunst <- read_excel(paste0("Input_data/",fy,
                               "/Annual_Reports/BIOSOLIDS_ConstituentsUnst.xls"),
                        skip=8) %>%
  remove_empty(which="rows")%>%
  rename("pH"= "...14", "...15"="pH",
        "TK\n\nmg/kg"="...24","...25"="TK\n\nmg/kg",
        "Nitrate\nNitrogen\nmg/kg"="...26","...27"="Nitrate\nNitrogen\nmg/kg",
        "Total\nCadmium\nmg/kg"="...29",
        "Total\nCopper\nmg/kg"="Total\nCadmium\nmg/kg",
        "...31"="Total\nCopper\nmg/kg",
        "Total\nNickel\nmg/kg"="...32",
        "Total\nLead\nmg/kg"="Total\nNickel\nmg/kg",
        "...35"="Total\nLead\nmg/kg",
        "Total\nZinc\nmg/kg"="...36",
        "Total\nMercury\nmg/kg"="Total\nZinc\nmg/kg",
        "Total\nArsenic\nmg/kg"="Total\nMercury\nmg/kg",
        "...40"="Total\nArsenic\nmg/kg",
        "Total\nMolybdenum\nmg/kg"="...41","...42"= "Total\nMolybdenum\nmg/kg",
        "Total\nSelenium\nmg/kg"="...44",
        "PCBs\n\nmg/kg"="Total\nSelenium\nmg/kg",
        "CaCO3\n\n%"="PCBs\n\nmg/kg","...50"="CaCO3\n\n%"
```

```
        )

# Vector to get empty columns
unnamed <- colnames(constunst)%>%
  .[grepl('[0-9]',.)] %>%
      gsub("[^0-9]", "", .)%>%
  as.integer

# Filter out empty columns leaving only those with data
constunst <- constunst[,-unnamed]

# Get the row number of the last facility
nrow.lastUnst <- min(which(constunst$Plant==lastUnst))

# Remove first row and end rows with annotation
constunst <- constunst %>%
  fill(everything(),.direction = "down") %>%
  .[c(2:nrow.lastUnst),]

# Function to retain odd numbered rows
row_odd <- seq_len(nrow(constunst))%%2
constunst.final <- constunst[row_odd == 1,]
```

CHECK OUTPUT TO SEE IF EVERYTHING LOOKS GOOD. If TEMPO export format changes, the code above would need adjustments!

Now merge the data and perform calculations

```
const.all <- bind_rows(constb.final,constunst.final)%>%
  arrange(County,Plant)

const.county <- const.all %>%
  group_by(County)%>%
  summarise_at(c(4:7,9),mean,na.rm = TRUE)%>%
  rename(`Dry Content Average (%)`= `Solid\nContent\n%`,
       `NH4_N (%)` = `NH4\n\n%`,
       `NO3_N (mg/kg)` = `Nitrate\nNitrogen\nmg/kg`,
       `TKN (%)` = `TKN\n\n%`,
       `TP (mg/kg)` = `TP\n\nmg/kg`)%>%
  .[,c(1,2,4,6,3,5)]
```

Calculating Generated and Used

Using the generators report in state, we estimate the amount of biosolids applied in MD. The imported and distributed in MD from the out of State report is used to check if all imported biosolids have been accounted for. Once this check is added to the table, the imported amount is divided across the counties with biosolids application and factored into their total before calculating final nutrient loads per year.

```
# Import Sewage Sludge Generated and Utilized Report

# Biosolid amount from Out of State
out_state <- read_excel(paste0("Input_data/",fy,
                               "/Annual_Reports/BIOSOLIDS_Out_of_State.xls"),
```

```
                        skip=3)%>%
  rename("Facility"="...1")%>%
  .[!is.na(.$Facility),]%>%
  .[.$Facility=="Total",11]

in_state <- read_excel(paste0("Input_data/",fy,"/Annual_Reports/BIOSOLIDS_In_State.xls"), rang
e="A12:AB67") %>%
  remove_empty(which="rows")%>%
  clean_names()%>%
  rename("imported_from_another_county_in_md" = "x10",
         "hauled_out_of_state" = "imported_from_another_county_in_md",
         "hauled_to_another_wwtp_in_county" = "hauled_out_of_state",
         "agricultural_land_in_county" = "hauled_to_another_wwtp_in_county",
         "x14" = "agricultural_land_in_county",
         "marginal_land_reclamation_in_county" = "x15",
         "x16" = "marginal_land_reclamation_in_county",
         "distributed_marketed" = "x17",
         "x18" = "distributed_marketed",
         "landfill_utilization_disposal_in_county" = "x19",
         "x20" = "landfill_utilization_disposal_in_county",
         "incinerated_in_county" = "x21",
         "x22" = "incinerated_in_county",
         "storage_in_county" = "x23",
         "x24" = "storage_in_county",
         "other_type" = "x25",
         "x26" = "other_type")

unnamed <- colnames(in_state)%>%
  .[grepl('[0-9]',.)] %>%
      gsub("[^0-9]", "", .)%>%
  as.integer

in_state <- in_state[,-unnamed]%>%
  clean_names()

# Simplify table

bio_raw <- in_state %>%
  rename(generated=total_generated,
         imported_to_md = imported_into_md_from_out_of_state,
         county_export = exported_to_another_county_in_md,
         county_import = imported_from_another_county_in_md,
         houled_out = hauled_out_of_state,
         county_haul = hauled_to_another_wwtp_in_county,
         ag_county = agricultural_land_in_county,
         land_rec_county = marginal_land_reclamation_in_county,
         dist_mrkt = distributed_marketed,
         ldfill_disposal = landfill_utilization_disposal_in_county,
         incinerated = incinerated_in_county,
         storage = storage_in_county,
         other = other_type)%>%
  mutate(across(2:14,~ as.numeric(.x)))

bio_raw[is.na(bio_raw)] <- 0

# Estimating generated vs. used biosolids
bio_calc <- bio_raw %>%
  mutate(
```

```
    incoming = generated+imported_to_md+county_import,
    used_export = county_export+houled_out+county_haul+ag_county+
      land_rec_county+dist_mrkt+ldfill_disposal+incinerated+storage+other,
    end_balance = as.numeric(incoming-used_export,digits=2),
    used = ag_county+land_rec_county+dist_mrkt+ldfill_disposal)

# Number of counties that applied biosolids
using.counties <- sum(bio_calc$used != 0)-1

# Biosolids imported and marketed in MD are equally distributed to the
# counties applying.

bio_calc <- bio_calc %>%
  mutate(used_all = ifelse(used != 0, used+(.$dist_mrkt[22]/using.counties),
                           used)
  )
```

Calculating Nutrients in Biosolids Applied in Maryland

Now that we have the calculation table with average dry tons, and nutrient content per county, and the total wet tons of biosolids applied in Maryland, we can create the final output to submit to CBP. Values multiplying each new variable converts the result to the appropriate unit: tons to pounds per year, and mg per kg to pounds per year.

```
cnty.fips <- read_excel("Templates/BiosolidsTemplate_PSTool.xlsx", sheet = 2)

final.calc <- merge(const.county,bio_calc[,c(1,19)],
                    by.x="County", by.y="county")%>%
  mutate(
    Year = cy,
    State = "MD",
    NPDES = NA,
    County = paste0(County,", MD"),
    County = gsub("'", '', County),
    Amount = used_all*(`Dry Content Average (%)`/100),
    Amount = ifelse(County == cdville$county.applied, #add Cedarville Lagoon data
                    Amount+cdville$dry.tons.applied,
                    Amount),
    Unit = "tons",
    `Dry or Wet` = "Dry",
    `Moisture Content %` = NA,
    NH4 = Amount*(`NH4_N (%)`/100)*2000,
    NO3 = Amount*`NO3_N (mg/kg)`*(907.185/453592),
    TKN = Amount*(`TKN (%)`/100)*2000,
    TON = TKN - NH4,
    TP = Amount*`TP (mg/kg)`*(907.185/453592),
    PO4 = TP/2,
    TOP = TP/2,
    TN = NH4+NO3+TON
    ) %>%
  left_join(.,cnty.fips, by = join_by(County == `County Name`))%>%
  rename(`Total Annual Wet Applied (tons)`= used_all,
         `County Applied (FIPS)` = `FIPS Code`)%>%
  .[, c(8:10,1,23,7,2,11:14,3,15,4,16,5,17,18,20:22,6,19)]

final.calc <- final.calc[!is.na(final.calc$`County Applied (FIPS)`),]
```

```
# Save all tables in one file
biosolids_list <- list(final.calc,const.all,const.county,cdville)
names(biosolids_list) <- c("FINAL TABLE","CALCULATION TABLE",
                           "COUNTIES AVERAGES","CEDARVILLE LAGOON")

OUT <- createWorkbook()

# add each dataframe from 'biosolids_list' to a tab in the workbook
lapply(names(biosolids_list), function(x){
  addWorksheet(OUT,x)
  writeData(OUT, x, x = biosolids_list[[x]])})

saveWorkbook(OUT, paste0("Output_data/",fy,"/Annual_Reports/Maryland CY ",
cy," Biosolids Table.xlsx"))
```

Check output before generating a file that matches the submission format.

Once the output is saved, Cedarville Lagoon report details should be added, and any additional editing for better visualization of the information is done:

- Copy the format in "CEDARVILLE LAGOON" tab from the previous reporting year and fill in with the current reporting year's information - SSU Permit Annual Report provided by LMA (Thomas Yoo and Nazeeh Freij).

After the editing is finished and if nothing needs to be corrected, run the last portion of the code to output the file matching the PS Tool template format.

Create Final Output

```
final.format <- read_excel(paste0("Output_data/",fy,
                                  "/Annual_Reports/Maryland CY ",cy,
                                  " Biosolids Table.xlsx"), sheet=1) %>%
  .[,c(1:3,5,8:11,13,15,18:21,23)]

write.xlsx(final.format, paste0("Output_data/",fy,"/Annual_Reports/Maryland CY",
           cy," Biosolids Final Submission.xlsx"))
```