USGS
science for a changing world

# Methods for Estimating Annual Wastewater Nutrient Loads in the Southeastern United States

Open-File Report 2007–1040

U.S. Department of the Interior
U.S. Geological Survey

# Methods for Estimating Annual Wastewater Nutrient Loads in the Southeastern United States

By Gerard McMahon, Larinda Tervelt, and William Donehoo

Open-File Report 2007–1040

**U.S. Department of the Interior**
**U.S. Geological Survey**

**U.S. Department of the Interior**
DIRK KEMPTHORNE, Secretary

**U.S. Geological Survey**
Mark D. Myers, Director

U.S. Geological Survey, Reston, Virginia: 2007

For product and ordering information:
World Wide Web: *http://www.usgs.gov/pubprod*
Telephone: 1-888-ASK-USGS

For more information on the USGS--the Federal source for science about the Earth, its natural and living resources, natural hazards, and the environment:
World Wide Web: *http://www.usgs.gov*
Telephone: 1-888-ASK-USGS



Suggested citation:
McMahon, Gerard, Tervelt, Larinda, and Donehoo, William, 2007, Methods for estimating annual wastewater nutrient loads in the southeastern United States: U.S. Geological Survey Open-File Report 2007–1040, 81 p.

# Contents



# Figures

## Tables



## Conversion Factors

| Multiply | By | To obtain |
|---|---|---|
| | Length | |
| mile (mi) | 1.609 | kilometer (km) |
| | Flow rate | |
| million gallons per day (MGD) | 0.04381 | cubic meters per second ($m^3/s$) |

Note: The abbreviation MGD will be used in this report as an abbreviation for “million gallons per day” for consistency between the text in the body of the report and language used in the programs included in the attachments.

# Methods for Estimating Annual Wastewater Nutrient Loads in the Southeastern United States

By Gerard McMahon[1], Larinda Tervelt[2], and William Donehoo[2]


## Abstract

This report describes an approach for estimating annual total nitrogen and total phosphorus loads from point-source dischargers in the southeastern United States. Nutrient load estimates for 2002 were used in the calibration and application of a regional nutrient model, referred to as the SPARROW (SPAtially Referenced Regression On Watershed attributes) watershed model. Loads from dischargers permitted under the National Pollutant Discharge Elimination System were calculated using data from the U.S. Environmental Protection Agency Permit Compliance System database and individual state databases. Site information from both state and U.S. Environmental Protection Agency databases, including latitude and longitude and monitored effluent data, was compiled into a project database. For sites with a complete effluent-monitoring record, effluent-flow and nutrient-concentration data were used to develop estimates of annual point-source nitrogen and phosphorus loads. When flow data were available but nutrient-concentration data were missing or incomplete, typical pollutant-concentration values of total nitrogen and total phosphorus were used to estimate load. In developing typical pollutant-concentration values, the major factors assumed to influence wastewater nutrient-concentration variability were the size of the discharger (the amount of flow), the season during which discharge occurred, and the Standard Industrial Classification code of the discharger. One insight gained from this study is that in order to gain access to flow, concentration, and location data, close communication and collaboration are required with the agencies that collect and manage the data. In addition, the accuracy and usefulness of the load estimates depend on the willingness of the states and the U.S. Environmental Protection Agency to provide guidance and review for at least a subset of the load estimates that may be problematic.


## Introduction

One of the greatest challenges in developing regional-scale nutrient water-quality models is the estimation of nutrient inputs used in the model calibration and application (McMahon and others, 2003). Nonpoint-source nutrient input estimates can be mass-based (for example, total annual nutrient mass associated with fertilizers or atmospheric deposition) or area-based (for example, areas of agricultural or urban land). Nonpoint-nutrient input estimates typically are developed for a spatial area of interest, such as a stream reach catchment, a watershed, a county, or for some larger regional unit. Point-source nutrient input estimates, expressed as annual nutrient loads, depend on knowledge of effluent discharge, or flow, and nutrient-concentration data from point-source dischargers in the study area. In the case of both point and nonpoint sources of nutrients, investigators may find it difficult to develop input data estimates that are accurate, that cover the entire study area, and that are compiled using comparable methods across an entire regional study area.

U.S. Geological Survey (USGS) National Water-Quality Assessment (NAWQA) Program and U.S. Environmental Protection Agency (USEPA) researchers have estimated annual total nitrogen (TN) and total phosphorus (TP) loads associated with approximately 3,000 point-source wastewater dischargers in the southeastern United States. These nutrient load estimates are for use in the calibration and application of a regional nutrient model, referred to as the SPARROW (SPAtially Referenced Regression On Watershed attributes; Smith and others, 1997) watershed model. The study area, referred to as Major River Basin-2 (MRB2), includes river basins draining to the South Atlantic and Gulf of Mexico (fig. 1).

---

[1] U.S. Geological Survey, Raleigh, North Carolina.

[2] U.S. Environmental Protection Agency, Atlanta, Georgia.


LOCATION OF THE STUDY AREA IN THE SOUTHEASTERN UNITED STATES

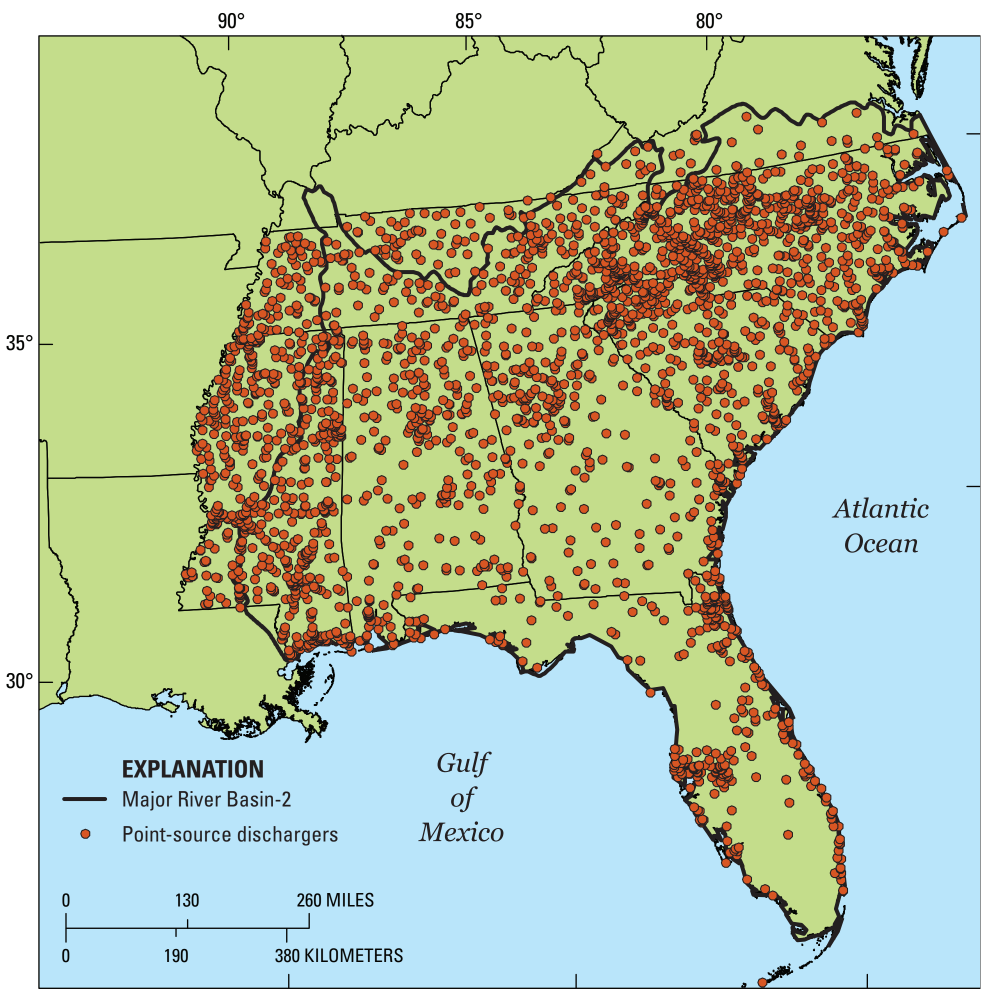


**Figure 1.** Major River Basin-2 study area and point-source dischargers.

## Purpose and Scope

This report presents a replicable method, based on explicit assumptions, for estimating the annual loads of total nitrogen and total phosphorus in wastewater discharges from National Pollutant Discharge Elimination System (NPDES) regulated facilities in the southeastern United States. This methodology is intended to be applicable in studies in other parts of the country. The spatial scope of the study included individual point-source dischargers in Alabama, Florida, Georgia, Mississippi, North Carolina, South Carolina, and Tennessee (fig. 1). The temporal scale of the project included the development of annual point-source nutrient estimates for 2000–2002, a time period coincident with the SPARROW model calibration period.

## Use of Point-Source Nutrient Load Estimates in SPARROW

The SPARROW model, which is calibrated using nonlinear regression methods, uses a mass-balance conceptual approach to account for the introduction, transport, and processing of nutrients at the scale of individual reaches in a stream network. The model enables reach-specific predictions of nutrient load and concentration, as well as the apportioning of shares of this load among the nutrient sources (Smith and others, 1997). Geographic information system (GIS) data-management capabilities are used to manage information about nutrient sources, in-stream nutrient flux measured at monitoring sites, and landscape characteristics at the scale of individual stream reaches and reach catchments. The GIS uses the topological characteristics of the stream network (for example, Reach File 1 (RF1; U.S. Geological Survey, 2003) or National Hydrography Dataset (NHD; U.S. Geological Survey, 2006), such as connectivity and flow direction, to account for location and connectivity of individual stream reaches, thus enabling the introduction, transport, and processing of nutrients within a watershed to be accounted for in a spatially explicit manner. SPARROW specifies that in-stream nutrient flux is a function of a nonlinear relation between reach-specific nutrient inputs (for example, point sources, atmospheric deposition, agricultural inputs), and landscape and in-stream nitrogen processing. The statistical basis of SPARROW provides an objective means of specifying a relation between nutrient flux and the sources and losses of nutrients that occur within the network of stream reaches and reach catchments. The SPARROW modeling approach uses information derived from the stream-reach network about the spatial relation among nutrient fluxes, sources, landscape characteristics, and stream characteristics. The processing and delivery of nutrients to downstream water bodies are estimated as a function of the location, magnitude, and interactions of nutrient sources with the terrestrial and aquatic properties of the river basin. Estimates of the magnitude and proportional share of various point and nonpoint sources can be made for an individual stream reach or for the entire basin.

## Challenges in Developing Point-Source Nutrient Load Estimates

Implementation of a SPARROW model requires information about nutrient sources, including point-source nutrient loads, at the stream-reach scale. Researchers would like to have data for each point-source discharger in the model study area, including regular and accurate entries for flow and nutrient concentration. Ideally, these data would be reported frequently enough to represent variability in facility operating conditions and in factors, such as seasonality and size and type of discharger, that influence nutrient discharge concentrations. Unfortunately, the ideal point-source data set typically does not exist for all dischargers in a region the size of this study area.

Point-source dischargers are required to monitor and report information about pollutants as defined in facility-specific NPDES discharge permits that are required by the Clean Water Act (United States Code, Amended 2002). These monitoring data are required to be stored in the USEPA Permit Compliance System (PCS) database. Typically, these data are collected as part of a self-monitoring program defined in the discharger's NPDES permit. The data are transferred to a state database by means of a facility-specific Discharge Monitoring Report (DMR), from which the data are transferred to regional USEPA PCS databases.

The reasons are varied why the PCS and state databases may not provide complete, up-to-date information for all facilities that may be discharging nutrients. Although the data structure of the USEPA PCS database is consistent nationwide, reporting requirements under the Clean Water Act vary for individual dischargers. Permit-specific discharge-concentration limits and reporting requirements for a constituent discharged by a facility, such as nitrogen or phosphorus, may vary among dischargers in a region as will the data entered into the PCS database. For example, not all dischargers may be required to monitor for total nitrogen during a particular time period; thus, data may be available for flow but not for total nitrogen concentration (or the concentrations of nitrogen constituents that would be combined to estimate total nitrogen), even if nitrogen is in the effluent. Generally, if a state does not have water-quality standards for a specific constituent in the watershed where a facility is located, the facility's permit may not include a limit for that constituent, and the monitoring data from that facility generally will not include information for the constituent.

Completeness, accuracy, and currency of the monitoring data may be affected by errors in data entry and data transfer. Data entry errors occur in two ways. Data may be entered incorrectly; for example, a monthly average flow of 0.10 million gallons per day (MGD) may be entered as 10 MGD. Also, data reported from the laboratory inadvertently may not be entered into the database. Another possibility is that updated data may not be transferred to the PCS database. For example, improved site-location information may be collected as part of a permit-renewal process by using global positioning system instruments to develop more accurate facility latitude and longitude data. Although these data may be entered in state-maintained site files, the data may not be transferred to the PCS database.

## Acknowledgments

The authors thank the following individuals for assistance in compiling data and providing advice on preparing load estimates: Mike Templeton and Jeanne Phillips of the North Carolina Division of Water Quality; Dale Stoudemire, Larry Turner, Angela Murray, and Mike Montebello of the South Carolina Department of Health and Environmental Control; Bill Noell of the Georgia Department of Natural Resources; Wayne Magley of the Florida Division of Water Resource Management; Lynn Sisk of the Alabama Department of Environmental Management; Dusty Myers of the Mississippi Department of Environmental Quality; Bruce Evans of the Tennessee Division of Water Pollution Control; and Doug Moyers of the USGS Virginia Water Science Center. Thanks also are extended to Anne Hoos and Tammy Ivahnenko of the USGS and Steve Rubin of the USEPA for providing editorial and scientific reviews that substantially improved the quality of this report.

# Methods of Investigation

The method for estimating point-source loads presented in this report extends the procedures developed at the USEPA Gulf of Mexico Program for studies of the Mississippi River basin during the late 1990s (U.S. Environmental Protection Agency, 1998a, b). Parallels and points of divergence can be found in the USEPA methods and the methods presented here.

A census approach was used in both methods to estimate loads from individual dischargers rather than using statistical approaches to estimate aggregate nutrient loads for the region of interest. For this study, the census approach was used because of the need to develop accurate, reach-scale estimates of nutrient inputs for calibration and application of SPARROW models.

Both methods are based on multiple sources of data, primarily the USEPA PCS database and state databases maintained independently of the PCS. The USEPA method also relied on treatment level information from the USEPA Clean Watersheds Needs Survey database and was able to deploy field investigators to examine individual discharger NPDES permit renewal applications and DMRs. In the method developed for this study, preference was given to using state data over PCS data when duplicate records were available for the same discharger, outfall, and date. It was assumed that data updates were made in the state databases first, with no assurance that the data updates were transferred to the PCS.

Both methods include calculated nutrient loads only for facilities that reported flow data. No effort was made to synthesize or estimate flow data for dischargers that did not report flow values; in other words, it was assumed that if no flow data were reported by a facility in the PCS for a given period, no discharge occurred from that facility for the given reporting period.

Both estimation efforts subset the data in the PCS or state databases to focus on certain dischargers thought to have nutrients in their effluent, using the 1987 Standard Industrial Classification SIC codes (U.S. Census Bureau, 2005). The current project included 54 SIC codes to identify dischargers that regularly had nutrients reported in their effluent, as indicated by the data (table 1).

The development of both methods for estimating point-source loads used typical pollutant concentrations (TPC) to populate monitoring records that include flow data but are missing nutrient data. The TPCs used in the USEPA studies (U.S. Environmental Protection Agency, 1998a, b) were drawn from studies conducted by the National Oceanic and Atmospheric Administration (NOAA; Percy Pacheco, National Oceanic and Atmospheric Administration, written commun., February 2006).

To develop the TPC data used in this study, nutrient concentration data for MRB2 dischargers were summarized by using a combination of factors thought to influence variability in discharge nutrient concentrations, including SIC codes, the magnitude of facility discharges, and seasons. The TPC data were matched to discharger records with missing concentration data. Additional national-scale TPC data, based on SIC codes, were obtained from a USEPA database and combined with regional data to populate remaining missing concentration values.

The method developed to estimate annual point-source nutrient loads is a six-step procedure:

1. Develop a site file by using PCS and state databases.
2. Retrieve monitoring data from PCS and state databases.
3. Import data into a common database.
4. Prepare data for load estimation.
5. Estimate annual nutrient load in wastewater discharge.
6. Quality assure estimates of nutrient load in wastewater discharges.

Each of these steps is described in detail below.

## Developing a Site File by Using the Permit Compliance System and State Databases

For each discharger, the permit number, name, SIC code, and location (latitude and longitude) were retrieved from the PCS by the USEPA Region 4 database manager. This information was supplemented with data obtained from the States of Alabama, Mississippi, North and South Carolina, and Tennessee containing the most recent available location information. SIC codes associated with effluents containing nutrients were identified based on the USEPA methods (U.S. Environmental Protection Agency, 1998a, b) and a review of nutrient-concentration data for permitted dischargers in the study area. These data were merged to produce a site-file listing of approximately 5,000 dischargers in the study area with a reasonable probability of having nutrients in their effluents.

**Table 1.** Standard Industrial Classification (SIC) codes (U.S. Census Bureau, 2005).

| SIC code | SIC description |
|---|---|
| 211 | Beef Cattle Feedlots |
| 213 | Hogs |
| 251 | Broiler, Fryers, and Roaster Chickens |
| 252 | Chicken Eggs |
| 253 | Turkey and Turkey Eggs |
| 254 | Poultry Hatcheries |
| 259 | Poultry and Eggs, NEC |
| 279 | Animal Specialities, NEC |
| 1475 | Phosphate Rock |
| 1541 | General Contractors-Industrial Buildings and Warehouses |
| 2011 | Meat Packing Plants |
| 2015 | Poultry Slaughtering and Processing |
| 2046 | Wet Corn Milling |
| 2085 | Distilled and Blended Liquors |
| 2611 | Pulp Mills |
| 2621 | Paper Mills |
| 2631 | Paperboard Mills |
| 2812 | Alkalies and Chlorine |
| 2819 | Industrial Inorganic Chemicals, NEC |
| 2821 | Plastics Material and Synthetic Resins, and Nonvulcanizable Elastomers |
| 2823 | Cellulosic Manmade Fibers |
| 2824 | Manmade Organic Fibers, Except Cellulosic |
| 2851 | Paints, Varnishes, Lacquers, Enamels, and Allied Products |
| 2865 | Cyclic Organic Crudes and Intermediates, and Organic Dyes and Pigments |
| 2869 | Industrial Organic Chemicals, NEC |
| 2873 | Nitrogenous Fertilizers |
| 2874 | Phosphatic Fertilizers |
| 2899 | Chemicals and Chemical Preparations, NEC |
| 2911 | Petroleum Refining |
| 3334 | Primary Production of Aluminum |
| 4941 | Water Supply |
| 4952 | Sewerage Systems |
| 4953 | Refuse Systems |
| 4959 | Sanitary Services, NEC |
| 5171 | Petroleum Bulk Stations and Terminals |
| 6513 | Operators of Apartment Buildings |
| 6515 | Operators of Residential Mobile Home Sites |
| 6552 | Land Subdividers and Developers, Except Cemeteries |
| 7011 | Hotels and Motels |
| 7032 | Sporting and Recreational Camps |
| 7033 | Recreational Vehicle Parks and Campsites |
| 7999 | Amusement and Recreation Services, NEC |
| 8062 | General Medical and Surgical Hospitals |
| 8211 | Elementary and Secondary Schools |
| 8221 | Colleges, Universities, and Professional Schools |
| 8222 | Junior Colleges and Technical Institutes |
| 8231 | Libraries |
| 8249 | Vocational Schools, NEC |
| 8351 | Child Day Care Services |
| 8361 | Residential Care |
| 8422 | Arboreta and Botanical or Zoological Gardens |
| 9223 | Correctional Institutions |
| 9621 | Regulation and Administration of Transportation Programs |
| 9711 | National Security |

## Retrieving Monitoring Data from the Permit Compliance System and State Databases

Permit Compliance System and state databases were used to develop a database of discharger flow and nutrient concentration data for the period 2000–2002. Each record in the databases represents a specific combination of a unique discharge identifier (designated by a permit number), a discharger outfall identifier (designating a discharge pipe), and date (typically data were reported as a summary of monthly mean conditions). A PCS retrieval resulted in four database files containing flow, concentration (file called DMR), facility, and base codes and unit information (file called LIMITS) for all dischargers in the study region (Attachment 1). All states in the region were contacted about the availability of state databases to use in conjunction with the PCS data to develop state-specific files of flow and concentration data. Only North and South Carolina were able to provide relatively complete state data files.

## Importing Data into a Common Database

The next step was to develop databases using a statistical program that could be used in subsequent steps to calculate annual nutrient loads for each discharger. A computer program (pcs_import_093005.sas; Attachment 2) was used to combine and analyze the data tables from PCS containing concentration data, facility information, flow data, and data on permit limits and data units (program lines 16–53; Attachment 2). Concentration and flow records with remark codes and coding errors were identified (lines 56–156; 160–243). Censored values were set to the censoring value for flow and concentration. Data files were created with a single record for each date, with variables defined for flow (lines 329–400) and concentrations of TN (lines 249–275) and TP (lines 277–314). All values represent mean monthly conditions. These files then were merged by discharger, outfall, and date. When subsequent load analysis detected errors in the data, the errors were corrected in the data-import program (e.g., lines 295–305; lines 333–391), a revised data file was created, and subsequent programs estimating nutrient load were rerun.

State data were processed similarly. The processing program for North Carolina nitrogen data is presented in Attachment 3. Nitrogen data, reported by river basin, were imported and combined with flow data (lines 20–164; Attachment 3). A data set with variables for individual nitrogen species was developed (lines 168–345). Several data characterization routines were run (lines 348–540), and a final data set was created (lines 543–581).

## Preparing Data for Load Estimation

The next step was to prepare state-specific data sets having a single record for each discharger, outfall, and date (typically, one record per month) containing flow and TN- and TP-concentration values. A data-preparation program (Attachment 4) was run for each state, using PCS data and state data, if available. In the example given in Attachment 4, data from the State of North Carolina were used to create a file that summarizes flow and concentration for each discharger and outfall by month (lines 31–102). Data from the state database were combined with PCS data (lines 109–187) to create a single record for each unique combination of permit/outfall/year/month. PCS values for flow (q1flow), TN (c2p600), and TP (c2p665) were based on the PCS variables Q1 (flow) and C2 (nutrient concentration). The PCS file containing information on flow and concentration units did not contain information for most discharger records, so an assumption regarding units was necessary. Flow and concentration values in the databases were assumed to be monthly mean values, and values for flow and concentration were assumed to represent million gallons per day and milligrams per liter, respectively. All flow and concentration data were screened for outlier values.

A single flow value was defined for each discharger in North Carolina for each month for which data were available during 2000–2002 (lines 189–236; Attachment 4) and five flow-class values were defined for each facility's discharge in any month (lines 229–233). Preference was given to using state data, if available, for reasons described above. The number of flow values contained in the dataset, by state, for the period 2000–2002 ranged from 25,806 in North Carolina to 7,768 in Alabama. The majority of the flow observations were from minor dischargers; that is, permit holders with discharges of less than 1 MGD (fig. 2).

Guidelines for the calculation of nutrient load in the next processing step were that (1) only records with flow data were used in load calculations, and (2) all records with flow data were used in load calculations if a suitable TN or TP concentration could be estimated. Two approaches were used to calculate the nutrient-concentration data for as many records, which included flow data, as possible and are discussed in order of preference.

Monthly TN and TP concentration values from PCS or state databases were used if they were available for a discharger in a given year and month. Single monthly TN and TP concentration values were defined for each discharger and outfall (lines 253–301; Attachment 4), with preference given to state data. If a few concentration values were missing for a given discharger during a year, median seasonal values for TN and TP were estimated using data for the entire 2000–2002 period for that

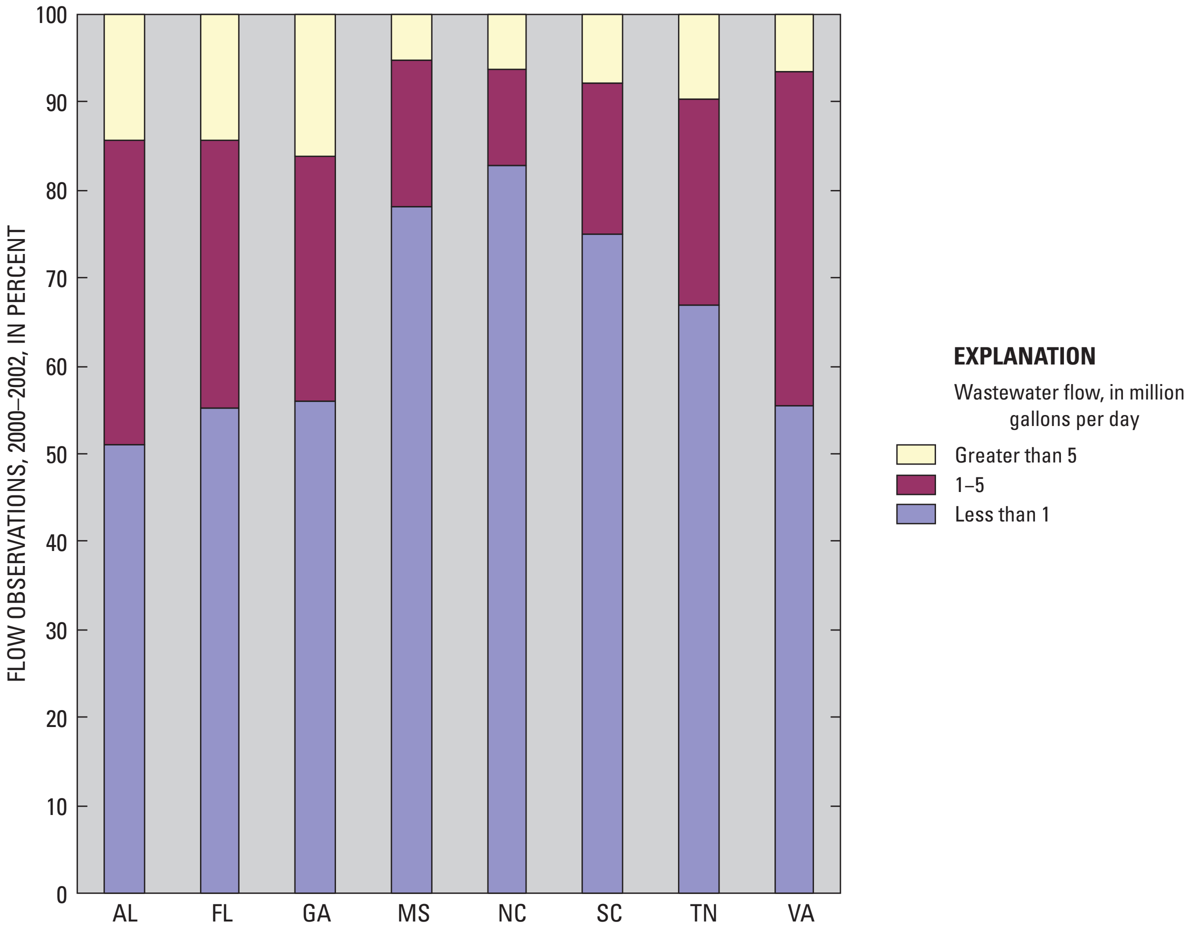


**Figure 2.** Summary of monthly effluent flow observations, by wastewater-flow classification and state, 2000–2002.

discharger; the resulting median concentration values were used for months in which flow data were available but concentration values were not (lines (330–370). Discharger-specific TN and TP concentration data were available most extensively in the States of Florida and North Carolina (fig. 3).

For the remaining records that contained flow data but no TN and TP concentration data, TPC values, developed in the program flowsicsea_060906.sas (Attachment 5), were substituted for missing concentration values. Three different types of TPC values were used in load calculations in declining order of preference. First, discharger data from all southeastern states were pooled and median seasonal concentration values were calculated by flow class, SIC code, and season (lines 37–121; Attachment 5). The rationale for preferring this method for estimating TPC is that flow class, SIC code, and season have an important effect on the variability of nutrient concentrations in effluent. Data on other factors that might affect variability, such as treatment level, were not uniformly available for all dischargers in the PCS and state databases. TPC values developed by using this method were merged with the main database by flow class, SIC code, and season (lines 402–425; Attachment 4).

To develop TN and TP concentration values for the remaining records with no concentration data (that is, monthly TN or TP data from either PCS or the state were not available and the TPC values based on flow class, SIC code, and season could not be matched to dischargers having flow data but no concentration data), the pooled data from all southeastern states were used to estimate median concentrations for TN and TP by flow class and SIC code (lines 124–161; Attachment 5). The resulting TPC values were merged with the main database and used to define missing TN and TP concentration values (lines 455–479; Attachment 4). Most concentration data for TN (fig. 3) and TP were derived either from facility-specific sources (that is, the monthly TN and TP concentration values for a discharger from PCS or the state data) or from TPC developed using discharger data specific to the MRB2 region.

Finally, if a TPC could not be drawn from the regional TPC database and matched with missing TN- or TP-concentration records, regional and national concentration data were used to develop TPC values by SIC code (lines 163–285; Attachment 5). National data were derived from a USEPA database (Steven Rubin, U.S. Environmental Protection Agency, written commun., February 2006). SIC-based TPC were used most

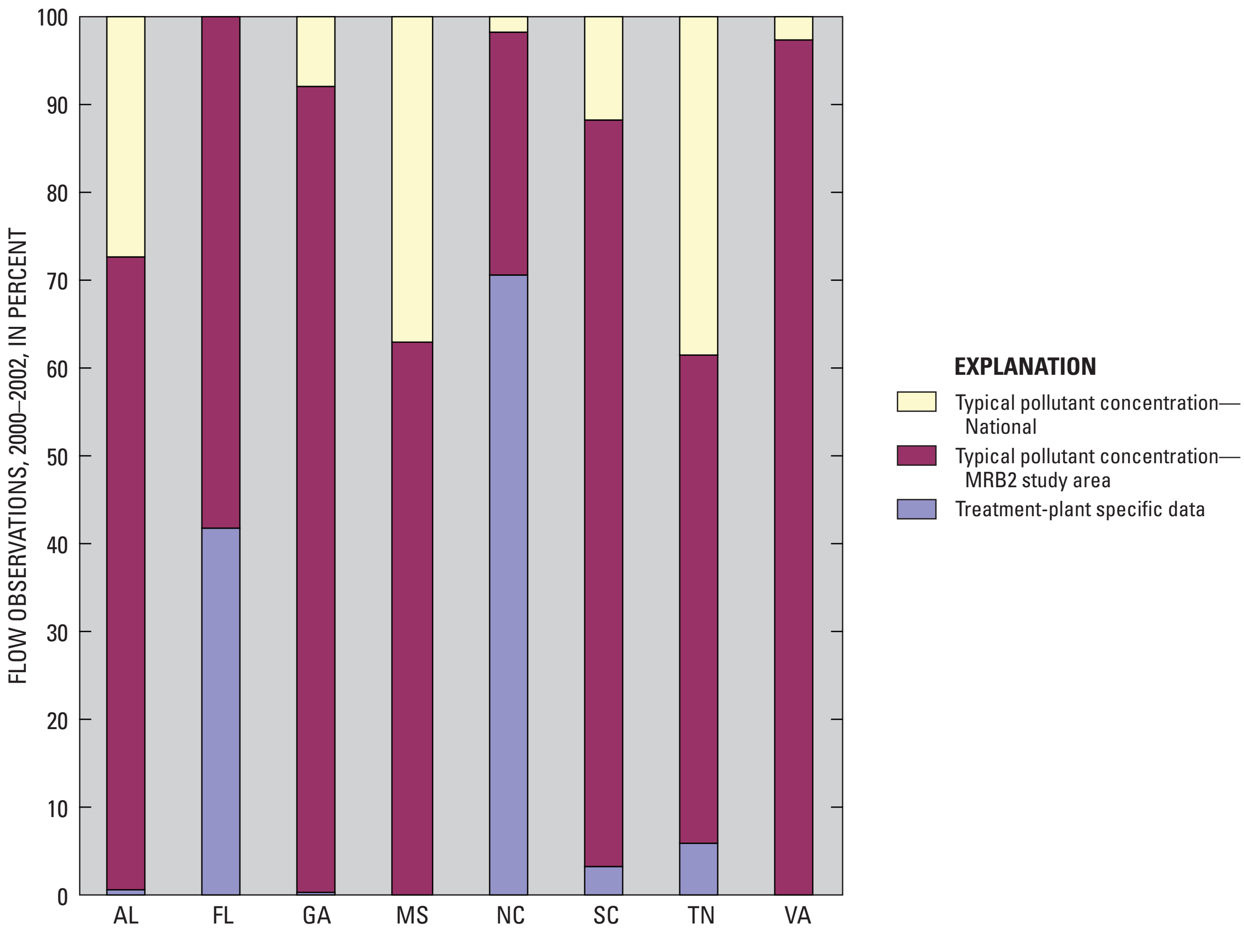


**Figure 3.** Summary, by state, of sources of total nitrogen concentration information used in load calculations.

extensively in the States of Alabama, Mississippi, and Tennessee (lines 508–534, Attachment 4; fig. 3).

TN and TP concentrations for the entire state data set were assessed to identify anomalous values for each discharger (Attachment 4, lines 607–669). Concentration greater than 10 times the median concentration value for a specific discharger during 2000–2002 and greater than the 95$^{th}$ percentile value of all concentration values over the entire data set were identified and replaced with the median concentration value for the specific discharger.

As a final step in the development of a data set for calculating nutrient loads, a value was calculated for each discharger, by outfall and year, indicating the number of months and the number of quarters with flow data (Attachment 4, lines 565–604). These values were used to determine the type of load calculation routine to be used for a discharger in a given year during the period 2000–2002, as described below. Summary data were used to prepare a GIS data set of facilities with annual nutrient load data; these GIS data were used to locate dischargers relative to streams within the MRB2 study area boundary.

## Estimating Annual Nutrient Load in Wastewater Discharge

The objective of this step was to calculate an annual nutrient load for each discharger in the database. In general, loads were calculated by multiplying flow and concentration values (typically representing mean conditions for a month) for a given time period and summing these load values for the period. Three load calculation scenarios were used. (Attachment 6 provides an example of a state-specific load calculation program.) If flow values were available for each month of the year for a discharger, annual load was calculated as: concentration multiplied by flow multiplied by the number of days in the month, summed over all months for the year (Attachment 6, lines 22–72). If monthly flow values existed for less than 12 months but were available in 3 or 4 quarters of the year, it was assumed that flow (and load) occurred in all 12 months of the year. In this case, annual load was calculated on a seasonal basis, equal to the discharger's seasonal median concentration (for 2000–2002) multiplied by its seasonal median flow multiplied by the number of days in the season, with seasonal load values summed to produce an annual load value (lines 77–131). If monthly flow values were available for less than 3 quarters of the year, load was calculated only for the months with flow data; monthly load values were summed to obtain an annual load estimate (lines 133–183). Data sets for individual years were created (Attachment 6, lines 206–250) and merged with site file information (lines 254–278).

The final objective of this step was to summarize information developed by the load preparation and load calculation programs (Attachment 7). Examples of data summaries include the number of dischargers by flow classification and state (table 2), the distribution of point-source load by treatment-plant size (table 3), the allocation of load calculation among the three load calculation methods (fig. 4), and the share of TN and TP produced by industries associated with the major SIC codes (table 4).

## Quality Assuring Estimates of Nutrient Load in Wastewater Discharges

The objective of this step was to identify and correct erroneous load values. In some cases, the sources of errors and appropriate corrective actions were obvious. For instance, if the median flow value for a discharger was 0.122 MGD and the reported flow value for a given month was 130 MGD, it was assumed that an error was made in data entry, and a corrected value of 0.130 MGD was entered in place of 130 MGD. Following this initial round of data corrections, states that supplied data were asked to examine estimated load values, identify potentially problematic values, and suggest possible corrections.

Two data-checking routines were used to identify potentially erroneous data. Initial inspection of load values for individual dischargers indicated that many of the load values that appeared to be erroneous were associated with very large flow values relative to the other flow values for a particular discharger rather than inordinately large TN or TP concentration values. A flow-magnitude checking program (Attachment 8) identified several flow categories of potential concern. The focus was to identify potential problems with the data from dischargers of flow greater than 1 MGD, referred to as major dischargers, as these dischargers contributed the large proportion of nutrient load in the MRB2 study area. Additional criteria for identifying data for further checking included mean flow values greater than 100 MGD, monthly flow values exceeding 10 times the median flow value for a particular discharger during the period 2000–2002, and flow values exceeding 100 times the median flow value.

A second data-checking routine, located in the load-calculation program (Attachment 6, lines 287–322), calculates interannual differences in TN and TP load for each discharger, and identifies large positive (greater than the 95$^{th}$ percentile) and negative (less than the 5$^{th}$ percentile) differences. An assumption was made that data problems, such as extreme

**Table 2.** Point-source wastewater dischargers by state and flow classification, 2000.

[Flow from minor dischargers averages less than 1 million gallons per day (MGD); major dischargers greater than or equal to 1 MGD]

| State | Number of dischargers, 2000 | | |
|---|---|---|---|
| | Minor | Major | Total |
| Alabama | 54 | 93 | 147 |
| Florida | 136 | 103 | 239 |
| Georgia | 152 | 98 | 250 |
| Mississippi | 473 | 64 | 537 |
| North Carolina | 670 | 123 | 793 |
| South Carolina | 224 | 83 | 307 |
| Tennessee | 187 | 84 | 271 |
| Virginia | 21 | 16 | 37 |
| Total | 1,917 | 664 | 2,581 |

**Table 3.** Total nitrogen and total phosphorus loads from National Pollutant Discharge Elimination System permitted dischargers, by state and flow classification, 2000.

[Flow classification units are million gallons per day. Percentages refer to the percentage of metric tons from all flow classes]

| State | Flow classification | | | |
|---|---|---|---|---|
| | Less than 1 | 1–5 | Greater than 5 | Metric tons, all flow classes |
| | Total nitrogen | | | |
| Alabama | 5% | 19% | 76% | 7,462 |
| Florida | 3% | 8% | 89% | 6,768 |
| Georgia | 5% | 15% | 79% | 11,440 |
| Mississippi | 17% | 39% | 44% | 4,710 |
| North Carolina | 9% | 27% | 63% | 11,607 |
| South Carolina | 9% | 24% | 67% | 7,032 |
| Tennessee | 8% | 18% | 74% | 9,602 |
| Virginia | 11% | 36% | 53% | 982 |
| | Total phosphorus | | | |
| Alabama | 7% | 24% | 69% | 1,117 |
| Florida | 7% | 22% | 71% | 1,157 |
| Georgia | 6% | 27% | 67% | 1,312 |
| Mississippi | 17% | 51% | 32% | 737 |
| North Carolina | 9% | 31% | 60% | 2,384 |
| South Carolina | 11% | 31% | 58% | 942 |
| Tennessee | 10% | 23% | 67% | 1,239 |
| Virginia | 11% | 39% | 50% | 149 |

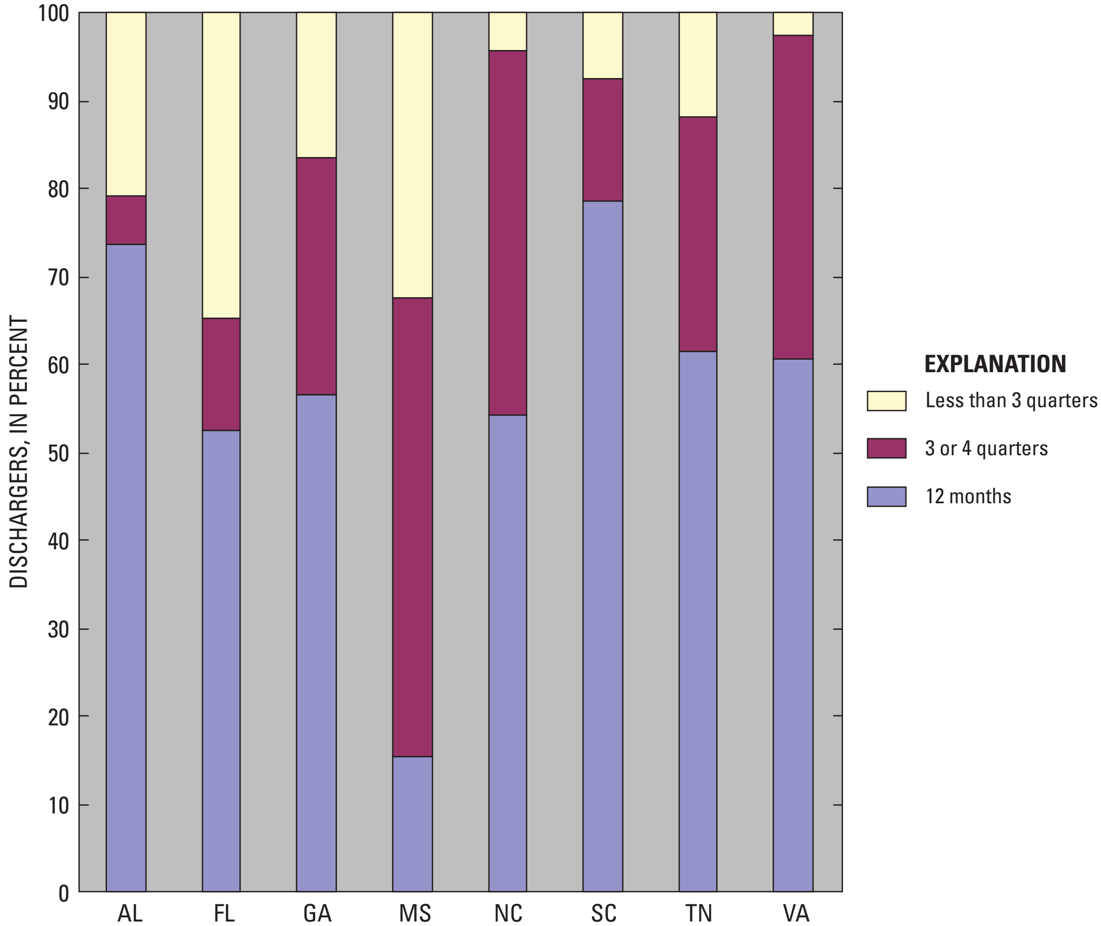


**Figure 4.** Percentage of dischargers using load calculation approaches based on data available for all 12 months, 3 or 4 quarters, or less than 3 quarters.

concentration or flow values attributed to laboratory errors or miscoding, would be revealed in larger interyear load estimates for a discharger. Annual load values flagged under these difference criteria, loads that were much larger than all other loads in the state (greater than the 95$^{th}$ percentile of all annual load estimates in the state), and data records with unusually high flow values, as discussed above, were compiled in a spreadsheet for further review by the states in the MRB2 study area.

Finally, data from 829 dischargers that had 12 months of flow data in 2002 and discharger-specific concentration data were used to compare loads generated using both original data and appropriate TPC SIC-based TN- and TP-concentration values (Attachment 7, lines 501–603). SIC-based concentration values were assumed most likely to differ from the discharger-specific concentration values. The Wilcoxen sign-rank test was used to test the hypothesis that the median difference between annual TN and TP loads calculated using the discharger-specific concentration data and TPC-based concentration data was zero (Helsel and Hirsch, 1992). No significant difference was noted in the TN loads calculated using the discharger and TPC concentration data. The TP loads calculated using the discharger-specific data were larger than loads calculated using the TPC data, a finding that indicates that TP loads calculated using SIC-based concentration values will be more conservative, or smaller, than TP loads calculated using discharger-specific concentration data.

**Table 4.** Annual total nitrogen and total phosphorus loads from point-source wastewater load dischargers in the southeastern United States, by Standard Industrial Classification (SIC) code, for 2000.

[Loads are expressed as the percentage share of total predicted point-source loads]

| SIC code | SIC description | Share of total load |
|---|---|---|
| | Nitrogen | |
| 4952 | Sewerage systems | 78% |
| 2611 | Pulp mills | 4% |
| 2621 | Paper mills | 4% |
| 2631 | Paperboard mills | 3% |
| 2823 | Cellulosic manmade fibers | 2% |
| 2011 | Meat packing plants | 2% |
| 2015 | Poultry slaughtering and processing | 2% |
| 2869 | Industrial organic chemicals, NEC | 1% |
| 2824 | Manmade organic fibers, except cellulosic | 1% |
| 9711 | National security | 1% |
| | Phosphorus | |
| 4952 | Sewerage systems | 73% |
| 2621 | Paper mills | 6% |
| 2611 | Pulp mills | 5% |
| 2874 | Phosphatic fertilizers | 4% |
| 2011 | Meat packing plants | 3% |
| 2631 | Paperboard mills | 3% |
| 2015 | Poultry slaughtering and processing | 2% |
| 1475 | Phosphate rock | 2% |

# Summary

Several lessons are evident from the development and application of the methods presented in this report for estimating annual nutrient loads from permitted point-source dischargers. First, access to flow, concentration, and location data requires close communication and collaboration with the agencies that collect and manage these data. Data needed to estimate loads at a regional scale cannot be retrieved from the Internet. Once load calculations have been completed, the accuracy and usefulness of the load estimates depend on the willingness of the states and the USEPA to provide guidance and review for at least a subset of the load values that appear to be problematic. States are in the best position to provide the most up-to-date location information for discharge facilities. Accurate locational information is critical for appropriate allocation of discharger nutrient-load estimates relative to the stream-reach data-management framework used in the SPARROW model.

Second, the number of discharger records with flow data alone far exceeds the number of records having both flow and nutrient-concentration data. In this situation, the accuracy of load estimates depends on the reasonableness of the TPC values. For this study, the major factors assumed in determining concentration variability were the magnitude of the discharge, the season when the discharge was measured, and the SIC code of the discharger. These factors were considered when assigning TPC to dischargers that did not have nutrient-concentration data available.

Finally, nutrient loads were calculated only for facilities that reported flow data. Although the majority of discharge observations across the study area were from minor dischargers (daily flow less than 1.0 MGD), a large proportion of nutrient load was generated by dischargers with flow greater than 1.0 MGD. From the standpoint of characterizing point-source nutrient loads for the region as a whole, it probably would be adequate to restrict load estimation to major dischargers. Because of the need in the SPARROW model to have a mass balance calculated using reach-scale data, nutrient load data from all dischargers are necessary in addition to facility location information, in order to locate discharges relative to stream reaches in the stream network.

## References Cited

Helsel, D.R., and Hirsch, R.M., 1992, Statistical methods in water resources: Amsterdam, Elsevier, 529 p.

McMahon, G., Alexander, R.B., and Qian, S., 2003, Support of TMDL programs using spatially referenced regression models: ASCE Journal of Water Resources Planning and Management, v. 129, p. 315–329.

Smith, R.A., Schwarz, G.E., and Alexander, R.B., 1997, Regional interpretation of water-quality monitoring data: Water Resources Research, v. 33, no. 12, p. 2781–2798.

U.S. Census Bureau, 2005, 1997 NAICS and 1987 SIC Correspondence Tables; accessed February 25, 2005, at *http://www.census.gov/epcd/naics/naicstb2.csv*

U.S. Environmental Protection Agency, 1998a, Documentation of Phase I and Phase II activities in support of point-source nutrient loading analysis in the Mississippi River system, Draft report, July 1998, USEPA Gulf of Mexico Program, Stennis Space Center, MS; accessed July 19, 2006, at *http://www.epa.gov/msbasin/taskforce/phases.htm*

U.S. Environmental Protection Agency, 1998b, Protocol for point source nutrient loading analysis in the Mississippi River system, April 1998, USEPA Gulf of Mexico Program, Stennis Space Center, MS; accessed July 19, 2006, at *http://www.epa.gov/msbasin/taskforce/protocol.htm*

U.S. Geological Survey, 2003, ERF1_2 — Enhanced River Reach File 2.0; accessed July 17, 2006, at *http://water.usgs.gov/GIS/metadata/usgswrd/XML/erf1_2.xml*

U.S. Geological Survey, 2006, National hydrography dataset; accessed July 17, 2006, at *http://nhd.usgs.gov/index.html*

United States Code, amended in 2002, Federal Water Pollution Control Act (a.k.a. Clean Water Act): Washington, D.C., U.S. Government Printing Office, 33 U.S.C. Secs. 101-607; accessed in June 2006, at *http://www.epa.gov/region5/water/pdf/ecwa.pdf*

# Attachments

# Attachment 1: PCS Data Retrieval Routine

**This retrieval extracts information describing the facility, including design flow:**

```
00 RMT=HOLD TIME=99M PRTY=0 MSGCLASS=A MSGLEVEL=1,1
01 04 FACILITIES INFORMATION
02 JERRY MCMAHON
10 STTE NE KY
20 QF CARD=NO DSN=FACILITY DISP=MOD
30 NPID
60 NPID FNMS MADI FLOW FLAT FLON FHBC CNTN
```

**This retrieval extracts effluent limits and monitoring requirements for flow, BOD, nitrogen, and phosphorus:**

```
00 RMT=HOLD TIME=99M PRTY=0 MSGCLASS=A MSGLEVEL=1,1
01 04 LIMITS INFORMATION
02 JERRY MCMAHON
10 STTE NE KY
20 QF CARD=NO DSN=LIMITS DISP=MOD GHOST=YES RESTRICT=YES
30 NPID PDSG PRAM MLOC SEAN MODN LIPQ LNTP
60 NPID PDSG LIPQ LNTP PRAM MLOC SEAN MODN ELSD ELED
60 LQUCD LQAV LQAS LQMX LQXS LCUCD LCMN LCMS LCAV LCAS LCMX LCXS
60 WITH PRAM AL 00310
60 WITH PRAM AL 00600
60 WITH PRAM AL 00610
60 WITH PRAM AL 00625
60 WITH PRAM AL 00630
60 WITH PRAM AL 00665
60 WITH PRAM AL 50050
60 WITH PRAM AL 80082
```

**This retrieval extracts actual flow measurements between January 1994 and December 2004:**

```
00 RMT=HOLD TIME=99M PRTY=0 MSGCLASS=A MSGLEVEL=1,1
01 04 DMR'S FOR FLOW
02 JERRY MCMAHON
10 STTE NE KY
20 QF CARD=NO DSN=FLOW DISP=MOD ARCH=YES RESTRICT=YES
30 NPID VDSG VPRM VMLO MVDT
60 NPID VDSG VIPQ VLIM VPRM VMLO VSEA VMOD MVDT MQAV MQMX MCMN MCAV MCMX
60 WITH VPRM EQ 50050
60 WITH MVDT GT 010194
60 WITH MVDT LT 010195
```

**This retrieval extracts BOD, nitrogen, and phosphorus measurements for 1994. One such retrieval was run for each year through 2004:**

```
00 RMT=HOLD TIME=99M PRTY=0 MSGCLASS=A MSGLEVEL=1,1
01 04 DMR'S FOR SELECTED PARAMETERS - 1994
02 JERRY MCMAHON
10 STTE NE KY
20 QF CARD=NO DSN=DMR94 DISP=MOD ARCH=YES RESTRICT=YES
30 NPID VDSG VPRM VMLO MVDT
60 NPID VDSG VIPQ VLIM VPRM VMLO VSEA VMOD MVDT MQAV MQMX MCMN MCAV MCMX
60 WITH VPRM AL 00310
```

```
60 WITH VPRM AL 00600
60 WITH VPRM AL 00610
60 WITH VPRM AL 00625
60 WITH VPRM AL 00630
60 WITH VPRM AL 00665
60 WITH VPRM AL 80082
60 WITH MVDT GT 010194
60 WITH MVDT LT 010195
```

# Attachment 2: PCS Data Import Program

```
/* ******************************************************************************

program: pcs_import_032106.sas
date: 3/20-21/06 7/6/2006(GA) 7/7-14/06 (FL,AL, MS, and TN))
from: pcs_import_093005.sas (3-20-06)

****************************************************************************** */

libname test ‘z:\nutrients\sparrow\point_source\PCS_donahoo_0805’;

options ps=54 ls=80;

/* import data developed by Mike Donahoo, August 2005
 and create tabular PCS data sets, */

/* import DMR, facility, flow, and limits data */
/*

PROC IMPORT OUT= TEST.DMR
    DATATABLE= “DMR”
    DBMS=ACCESS2000 REPLACE;
DATABASE=“z:\nutrients\sparrow\point_source\PCS_donahoo_0805\mcmahon2.mdb”;
RUN;

proc contents; run;

PROC IMPORT OUT= TEST.facility
    DATATABLE= “Facility”
    DBMS=ACCESS2000 REPLACE;
DATABASE=“z:\nutrients\sparrow\point_source\PCS_donahoo_0805\mcmahon2.mdb”;
RUN;


proc contents; run;

PROC IMPORT OUT= TEST.flow
    DATATABLE= “Flow”
    DBMS=ACCESS2000 REPLACE;
DATABASE=“z:\nutrients\sparrow\point_source\PCS_donahoo_0805\mcmahon2.mdb”;
RUN;

proc contents; run;

PROC IMPORT OUT= TEST.limits
    DATATABLE= “Limits”
    DBMS=ACCESS2000 REPLACE;
DATABASE=“z:\nutrients\sparrow\point_source\PCS_donahoo_0805\mcmahon2.mdb”;
RUN;

proc contents; run;
```

```
/* preparation of dmr data

 identification of records where concentration and quantity have "<" ">" or "-" value
 or other value that cannot be translated into a numeric value. Identification done
by multiplying each of the concentration and quantity variables by one, looking at the
error messages, and making changes in code creating test.dmr2. Assumption is made that sets
conc and flow values that have a "<" or ">" remark code equal to the censored value.
*/

/*
data test.dmr1;
  format state $2. season $6.;;
  set test.dmr;
  month=month(datepart(date));
  if month = 12 or month = 1 or month = 2 then season = 'winter';
  if month = 5 or month = 3 or month = 4 then season = 'spring';
  if month = 6 or month = 7 or month = 8 then season = 'summer';
  if month = 11 or month = 9 or month = 10 then season = 'fall';
  state=substr(npdes,1,2);
  year=year(datepart(date));
  outfall=substr(outfall,1,3);
run;


data test.dmr2;
 set test.dmr1;
 remark_c1 = 0;
 remark_c2 = 0;
 remark_c3 = 0;
 remark_q1 = 0;
 remark_q2 = 0;
 if substr(c1,1,1)='<' then remark_c1 = 1;
 if substr(c1,1,1)='>' then remark_c1 = 2;
 if substr(c1,1,1)='-' then remark_c1 = 3;
 if substr(c1,1,1)='*' then remark_c1 = 3;
 if substr(c2,1,1)='<' then remark_c2 = 1;
 if substr(c2,1,1)='>' then remark_c2 = 2;
 if substr(c2,1,1)='-' then remark_c2 = 3;
 if substr(c2,6,1)='-' then remark_c2 = 3;
 if substr(c2,1,1)='*' then remark_c2 = 3;
 if c2 = '10 24' then remark_c2 = 3;
 if substr(c2,3,1)='-' then remark_c2 = 3;
 if c3 = '1 5' then remark_c3 = 3;
 if c3 = '1 9' then remark_c3 = 3;
 if c3 = 'T' then remark_c3 = 3;
 if c3 = '9 224' then remark_c3 = 3;
 if c3 = '1-7' then remark_c3 = 3;
 if c3 = '4.0-' then remark_c3 = 3;
 if c3 = '4.4-' then remark_c3 = 3;
 if c3 = '3.4-' then remark_c3 = 3;
 if substr(c3,1,1)='<' then remark_c3 = 1;
 if substr(c3,1,1)='>' then remark_c3 = 2;
 if substr(c3,1,1)='-' then remark_c3 = 3;
 if substr(q1,1,1)='<' then remark_q1 = 1;
```

```
 if substr(q1,1,1)=‘>’ then remark_q1 = 2;
 if substr(q1,1,1)=‘-’ then remark_q1 = 3;
 if q1 = ‘9 18’ then remark_q1 = 3;
 if q1 = ‘11 .42’ then remark_q1 = 3;
 if q1 = ‘300 626’ then remark_q1 = 3;
 if q1 = ‘5.54-’ then remark_q1 = 3;
 if q1 = ‘14-38’ then remark_q1 = 3;
 if q1 = ‘13.4-’ then remark_q1 = 3;
 if q1 = ‘28 .0’ then remark_q1 = 3;
 if q1 = ‘5 592.6’ then remark_q1 = 3;
 if q1 = ‘6 433’ then remark_q1 = 3;
 if q1 = ‘*.2’ then remark_q1 = 3;
 if substr(q2,1,1)=‘<’ then remark_q2 = 1;
 if substr(q2,1,1)=‘>’ then remark_q2 = 2;
 if substr(q2,1,1)=‘-’ then remark_q2 = 3;
 if substr(q2,1,1)=‘*’ then remark_q2 = 3;
 if q2 = ‘50.4-’ then remark_q2 = 3;
 if q2 = ‘30.04-’ then remark_q2 = 3;
 if q2 = ‘*.47’ then remark_q2 = 3;
 if q2 = ‘30.8 3’ then remark_q2 = 3;
run;

data test.temp;
 set test.dmr2;
 if remark_c3 eq 0;
 conc3=c3*1;
run;



data test.dmr3;
 set test.dmr2;
 if remark_c1 = 1 or remark_c1 = 2 then conc1=substr(c1,2,7)*1;
 if remark_c1 = 3 then conc1 = .;
 if remark_c1 = 0 then conc1 = c1*1;
 if remark_c2 = 1 or remark_c2 = 2 then conc2=substr(c2,2,7)*1;
 if remark_c2 = 3 then conc2 = .;
 if remark_c2 = 0 then conc2 = c2*1;
 if remark_c3 = 1 or remark_c3 = 2 then conc3=substr(c3,2,7)*1;
 if remark_c3 = 3 then conc3 = .;
 if remark_c3 = 0 then conc3 = c3*1;
 if remark_q1 = 1 or remark_q1 = 2 then quan1=substr(q1,2,7)*1;
 if remark_q1 = 3 then quan1 = .;
 if remark_q1 = 0 then quan1 = q1*1;
 if remark_q2 = 1 or remark_q2 = 2 then quan2=substr(q2,2,7)*1;
 if remark_q2 = 3 then quan2 = .;
 if remark_q2 = 0 then quan2 = q2*1;
run;



/* preparation of flow data

 including deletion of records with monitoring location ne 1 (effluent gross value)
 identification of records where concentration and quantity have “<” “>” or “-” value
 Identification done by multiplying each of the concentration and quantity variables by one,
```

```
 looking at the
    error messages, and making changes in code creating test.flow2. Assumption is made that sets
    conc and flow values that have a “<” or “>” remark code equal to the censored value.
*/

/*
data test.flow1;
 format state $2. season $6.;
 set test.flow;
 month=month(datepart(date));
 if month = 12 or month = 1 or month = 2 then season = ‘winter’;
 if month = 5 or month = 3 or month = 4 then season = ‘spring’;
 if month = 6 or month = 7 or month = 8 then season = ‘summer’;
 if month = 11 or month = 9 or month = 10 then season = ‘fall’;
 state=substr(npdes,1,2);
 year=year(datepart(date));
 outfall=substr(outfall,1,3);
run;

data test.flow2;
 set test.flow1;
 remark_c1 = 0;
 remark_c2 = 0;
 remark_c3 = 0;
 remark_q1 = 0;
 remark_q2 = 0;
 if substr(c1,1,1)=‘<’ then remark_c1 = 1;
 if substr(c1,1,1)=‘>’ then remark_c1 = 2;
 if substr(c1,1,1)=‘-’ then remark_c1 = 3;
 if substr(c1,1,1)=‘*’ then remark_c1 = 3;
 if substr(c2,1,1)=‘<’ then remark_c2 = 1;
 if substr(c2,1,1)=‘>’ then remark_c2 = 2;
 if substr(c2,1,1)=‘-’ then remark_c2 = 3;
 if substr(c2,6,1)=‘-’ then remark_c2 = 3;
 if substr(c2,1,1)=‘*’ then remark_c2 = 3;
 if substr(c3,1,1)=‘<’ then remark_c3 = 1;
 if substr(c3,1,1)=‘>’ then remark_c3 = 2;
 if substr(c3,1,1)=‘-’ then remark_c3 = 3;
 if substr(q1,1,1)=‘<’ then remark_q1 = 1;
 if substr(q1,1,1)=‘>’ then remark_q1 = 2;
 if substr(q1,1,1)=‘-’ then remark_q1 = 3;
 if q1 = ‘ . 798’ then remark_q1 = 3;
 if q1 = ‘1.58 1’ then remark_q1 = 3;
 if q1 = ‘.09 6’ then remark_q1 = 3;
 if q1 = ‘.03 0’ then remark_q1 = 3;
 if q1 = ‘0.1805 0’ then remark_q1 = 3;
 if substr(q2,1,1)=‘<’ then remark_q2 = 1;
 if substr(q2,1,1)=‘>’ then remark_q2 = 2;
 if substr(q2,1,1)=‘-’ then remark_q2 = 3;
 if substr(q2,1,1)=‘T’ then remark_q2 = 3;
 if q2 = ‘.0 0030’ then remark_q2 = 3;
 if q2 = ‘0.0 72’ then remark_q2 = 3;
 if q2 = ‘0. 007’ then remark_q2 = 3;

run;
```

```
data test.temp;
 set test.flow2;
 if remark_c3 eq 0;
 conc3=c3*1;
run;

data test.flow3;
 set test.flow2;
 if remark_c1 = 1 or remark_c1 = 2 then conc1=substr(c1,2,7)*1;
 if remark_c1 = 3 then conc1 = .;
 if remark_c1 = 0 then conc1 = c1*1;
 if remark_c2 = 1 or remark_c2 = 2 then conc2=substr(c2,2,7)*1;
 if remark_c2 = 3 then conc2 = .;
 if remark_c2 = 0 then conc2 = c2*1;
 if remark_c3 = 1 or remark_c3 = 2 then conc3=substr(c3,2,7)*1;
 if remark_c3 = 3 then conc3 = .;
 if remark_c3 = 0 then conc3 = c3*1;
 if remark_q1 = 1 or remark_q1 = 2 then quan1=substr(q1,2,7)*1;
 if remark_q1 = 3 then quan1 = .;
 if remark_q1 eq 0 then quan1=q1*1;
 if remark_q2 = 1 or remark_q2 = 2 then quan2=substr(q2,2,7)*1;
 if remark_q2 = 3 then quan2 = .;
 if remark_q2 eq 0 then quan2=q2*1;
run;

/* create and merge files for individual parameters — note that there
 are a number of edits related to checking limits file informtion
 and reviewing initial load estimates and associated data */
/*
data temp600;
 set test.dmr3;
 if parameter = ‘00600’;
 if mon_loc=‘1’ or mon_loc = ‘E’;
 if npdes eq ‘NC0006033’ and year = 2002 and (month = 11 or month = 12 ); conc2=conc2/10;

run;
proc sort; by year month; run;

proc print; var npdes outfall year month conc2 ;run;



data test.p600;
 set test.dmr3;
 if parameter = ‘00600’;
 if mon_loc=‘1’ or mon_loc = ‘E’;
 if npdes eq ‘NC0006033’ and year = 2002 and (month = 11 or month = 12 ) then conc2=conc2/10;
 c1p600=conc1;
 c2p600=conc2;
 c3p600=conc3;
 q1p600=quan1;
 q2p600=quan2;
 keep npdes outfall date year state c1p600 c2p600 c3p600 q1p600 q2p600;
run;

proc sort; by npdes outfall date; run;
```

```

data temp665;
 set test.dmr3;
 if parameter = ‘00665’;
 if mon_loc=‘1’ or mon_loc = ‘E’;
 if npdes eq ‘GA0000973’ and outfall = ‘020’ and year=2000 and month=7 ; conc2=conc2/100;

run;
proc sort; by year month; run;


proc print; var npdes outfall year month conc2 ;run;



data test.p665;
 set test.dmr3;
 if parameter = ‘00665’;
 if mon_loc=‘1’ or mon_loc = ‘E’;
 if npdes = ‘FL0043443’ and ( date ge “30APR2002:00:00:00”dt and
 date le “30JUN2004:00:00:00”dt )then conc2=conc1/1000;
 if npdes = ‘FL0168581’ and date gt “31JUL2002:00:00:00”dt then conc2=conc1/1000;
 if state = ‘FL’ and ( conc1 gt 0 or conc3 gt 0 ) and conc2 eq ‘ ‘ then conc2=(conc1+conc3)/2;
 if npdes eq ‘AL0061671’ and year=2002 and month=2 then conc2=conc2/100;
 if npdes eq ‘GA0000973’ and outfall = ‘010’ and year=2000 and month=7 then conc2=conc2/100;
 if npdes eq ‘GA0000973’ and outfall = ‘020’ and year=2000 and month=7 then conc2=conc2/100;
 if npdes eq ‘GA0030791’ and year=2001 and month=11 then conc2=conc2/100;
 if npdes eq ‘GA0046035’ and outfall = ‘0B0’ and year=2000 and month=7 then conc2=conc2/10;
 if npdes eq ‘GA0046035’ and outfall = ‘0B0’ and year=2001 and month=6 then conc2=conc2/10;
 if npdes eq ‘GA0047236’ and outfall = ‘0B0’ and year=2000 and month=5 then conc2=conc2/100;
 c1p665=conc1;
 c2p665=conc2;
 c3p665=conc3;
 q1p665=quan1;
 q2p665=quan2;
 keep npdes outfall date year state conc1 conc2 conc3 c1p665 c2p665 c3p665 q1p665 q2p665;
run;

proc sort; by npdes outfall date; run;
*/

data tempflow;
 set test.flow3;
 if parameter = ‘50050’;
 if mon_loc=‘1’ or mon_loc = ‘E’;
 flag=1;
 if npdes = ‘TN0065081’ and year =2002;
run;
proc sort; by year npdes outfall month; run;


proc print; var npdes outfall year month quan1 flag;run;


data test.p50050;
 set test.flow3;
```

```
if parameter = ‘50050’;
if mon_loc=‘1’ or mon_loc = ‘E’;
flag=1;
if state = ‘FL’ and quan1 eq ‘ ’ and quan2 gt 0 and quan2 le 8.1 then quan1=quan2;
if not (npdes eq ‘FL0000256’) ;
if not (npdes eq ‘FL0000230’) ;
if npdes eq ‘FL0000523’ and not( outfall=‘001’ or outfall=‘003’) then flag=0 ;
if npdes eq ‘FL0000655’ and not( outfall=‘102’ or outfall=‘104’) then flag=0 ;
if npdes eq ‘FL0001911’ and quan1 gt 1 then quan1=quan1/1000;
if npdes eq ‘FL0002488’ and not(outfall=‘002’) then flag=0;
if npdes eq ‘FL0002488’ and outfall=‘002’ and year=2001 and month=2 then quan1=quan1/10;
if npdes eq ‘FL0002607’ and year=2001 and month=4 then quan1=quan1/1000000;
if npdes eq ‘FL0002631’ and year=2001 and month=10 then quan1=quan1/100;
if npdes eq ‘FL0023922’ and year=2002 and month=8 then quan1=quan1/1000;
if not (npdes eq ‘FL0032441’) ;
if npdes eq ‘FL0037940’ and quan1 gt 10 then quan1=quan1/1000;
if npdes eq ‘FL0041785’ and year=2002 then flag=0;
if npdes eq ‘FL0042315’ and year=2001 and month=8 then quan1=quan1/1000;
if npdes eq ‘FL0043443’ and year=2000 and month = 7 then quan1=quan1/1000;
if npdes eq ‘FL0043770’ and not(year=2002) then flag=0;
if npdes eq ‘FL0044245’ and ((year=2000 and month=4) or (year=2001 and month=11)) then
   quan1=quan1/10;
if npdes eq ‘FL0140023’ and not(year=2002) then flag=0 ;
if npdes eq ‘FL0166511’ and year=2002 and month=4 then quan1=quan1/1000;
if not(npdes eq ‘FL0267538’) ;
if state = ‘AL’ and ( quan1 eq ‘ ’ or quan1 eq 0 ) and quan2 gt 0 and quan2 le .77 then quan1=quan2;
if not(npdes eq ‘AL0002780’);
if npdes =‘AL0020672’ and year=2000 and month=9 then quan1=quan2;
if npdes =‘AL0020869’ and year=2000 and month=1 then quan1=quan1/1000;
if npdes =‘AL0020869’ and year=2000 and month=3 then quan1=quan1/1000;
if npdes =‘AL0020869’ and year=2001 and month=2 then quan1=quan1/1000;
if npdes =‘AL0022713’ and year=2001 and month=11 then quan1=quan1/1000;
if npdes =‘AL0024724’ and year=2001 and month=3 then quan1=quan1/1000;
if npdes =‘AL0024724’ and year=2001 and month=7 then quan1=quan1/1000;
if npdes =‘AL0044105’ and year=2002 and month=12 then quan1=quan1/1000;
if npdes =‘AL0048763’ and year=2000 and month=9 then quan1=quan1/1000;
if npdes =‘AL0054631’ and year=2002 and month=5 then quan1=quan1/1000;
if npdes =‘AL0055841’ and year=2002 and month=12 and outfall=’001’ then quan1=quan1/1000;
if npdes =‘AL0055841’ and year=2001 and month=3 and outfall=’001’ then quan1=quan1/1000;
if npdes =‘AL0059218’ and year=2001 and month=8 then quan1=quan1/1000;
if npdes =‘AL0059218’ and year=2001 and month=5 then quan1=quan1/1000;
if npdes =‘AL0059218’ and year=2000 and month=8 then quan1=quan1/1000;
if npdes= ‘AL0020206’ and year=2000 and month=4 then quan1=7.5;
if not( npdes eq ‘AL0003093’);
if not( npdes eq ‘AL0003301’);
if not( npdes eq ‘AL0025968’);
if not( npdes eq ‘AL0002666’ and year=2002 and ( outfall = ‘003’ or outfall =’004’)) ;
if npdes eq ‘AL0020885’ and year=2000 and ( month=1 or month=2 or month=3 or month=4 or month=5
   or month=9 or month=10) then quan1=quan1/100;
if npdes eq ‘AL0022314’ and outfall=’002’ and year=2000 and ( month=1 or month=2 or month=3
   or month=4 or month=6 or month=8) then quan1=quan1/1000;
if npdes eq ‘AL0022314’ and outfall=’002’ and year=2000 and month=5 then quan1=quan1/1000000;
if npdes eq ‘AL0023922’ and year=2002 then quan1=.27;
if npdes eq ‘AL0052019’ and year=2002 then flag=0 ;
if npdes eq ‘AL0052850’ and year=2002 then quan1=.02;
if not(npdes eq ‘AL0053091’);
```

```
if npdes eq ‘AL0000116’ and ( outfall = ‘002’ or outfall =‘003’) then flag=0 ;
if npdes eq ‘AL0003026’ and not(outfall=‘001’) then flag=0;
if not(npdes eq ‘AL0003158’);
if npdes =‘SC0047821’ and year=2001 and month=10 then quan1=quan1/100;
if state = ‘MS’ and quan1 eq ‘ ‘ and quan2 gt 0 and quan2 le .05 then quan1=quan2;
if npdes =‘MS0033961’ and year=2001 and month=4 then quan1=quan1/1000;
if npdes =‘MS0033961’ and year=2002 and month=5 then quan1=quan1/1000;
if npdes=‘MS0000213’ and year=2000 and month=12 then quan1=quan1/100;
if npdes=‘MS0023345’ and year=2000 and month=1 then quan1=quan1/1000;
if npdes=‘MS0024627’ and year=2000 and month=1 then quan1=quan1/1000;
if npdes=‘MS0029513’ and year=2000 and month=9 then quan1=quan1/1000;
if npdes = ‘MS0000213’ then sic_code=‘2611’;
if not(npdes eq ‘MS0046655’);
if not(npdes eq ‘MS0052949’);
if not(npdes eq ‘MS0021601’);
if not(npdes eq ‘MS0038849’);
if not(npdes eq ‘MS0044296’);
if not(npdes eq ‘MS0044679’);
if not(npdes eq ‘MS0039934’);
if not(npdes eq ‘MS0046639’);
if not(npdes eq ‘MS0047147’);
if not(npdes eq ‘MS0049107’);
if not(npdes eq ‘MS0055387’);
if npdes eq ‘MS0001520’ and not(outfall=‘001’) then flag=0 ;
if npdes eq ‘MS0001686’ and outfall=‘001’ and year=2000 and month=9 then flag=0;
if npdes eq ‘MS0020206’ and year=2001 and month=9 then quan1=quan1/1000;
if npdes eq ‘MS0020362’ and outfall=‘003’ and year=2001 and month=3 then quan1=quan1/1000;
if npdes eq ‘MS0025526’ and year=2002 and month=7 then quan1=quan1/100;
if npdes eq ‘MS0028029’ and year=2002 and month=6 then quan1=quan1/1000;
if npdes eq ‘MS0028185’ and year=2000 and month=6 then quan1=quan1/10;
if not(npdes eq ‘MS0034878’);
if npdes eq ‘MS0036331’ and year=2000 and month=12 then quan1=quan1/100 ;
if npdes eq ‘MS0039144’ and year=2002 and month=12 then flag=0;
if npdes eq ‘MS0045551’ and year=2002 and month =9 and outfall=’001’ then quan1=quan1/10 ;
if npdes eq ‘MS0045772’ and year=2001 and month=9 and outfall=’001’ then quan1=quan1/1000;
if npdes eq ‘MS0045772’ and year=2002 and month=12 and outfall=’002’ then quan1=quan1/1000;
if npdes eq ‘MS0053589’ and year=2001 and month=3 then quan1=quan1/1000;
if npdes eq ‘MS0055379’ and year=2002 and (month=4 or month=5 or month=6 or month=7 or month=8
or month=9 or month=11) then quan1=quan1/1000;
if npdes eq ‘GA0000973’ and outfall=‘020’ and year=2000 and month=10 then quan1=quan1/100 ;
if npdes eq ‘GA0020478’ and outfall=‘0B0’ and year=2000 and month=5 then quan1=quan1/1000 ;
if npdes eq ‘GA0031046’ and outfall=‘0B1’ and year=2002 and month=8 then quan1=quan1/1000 ;
if npdes eq ‘GA0032514’ and year=2002 and month=7 then quan1=quan1/1000 ;
if npdes=‘GA0000311’ and year=2001 and month = 5 then quan1=.067;
if not(npdes eq ‘GA0003671’);
if not(npdes eq ‘GA0038318’);
if not(npdes eq ‘GA0003590’);
if npdes eq ‘GA0001279’ and year = 2000 and month = 2 then quan1=.201;
if npdes eq ‘GA0002798’ and year = 2000 and month=10 then quan1=quan1/10;
if npdes eq ‘GA0002798’ and year = 2002 and month=1 and outfall = ‘001’ then quan1=quan1/10;
if npdes eq ‘GA0003620’ and year = 2000 and month = 2 then quan1=quan1/10;
if npdes eq ‘GA0021032’ and quan1 ge 100 then quan1=quan1/1000;
if npdes eq ‘GA0021032’ and ( quan1 ge 1.65 and quan1 lt 10) then quan1=quan1/10;
if npdes eq ‘GA0021512’ and quan1 gt 1 then quan1=quan1/1000;
if npdes eq ‘GA0022900’ and year = 2002 and month=10 then quan1=quan1/100;
if npdes eq ‘GA0025674’ and year = 2002 and month=2 then quan1=quan1/10;
```

```
if npdes eq ‘GA0034819’ and year=2000 and month=4 then quan1=quan1/1000;
if npdes eq ‘GA0049166’ and year=2001 and month=1 then quan1=.058;
if npdes eq ‘TN0002135’ and year=2000 and month=5 then quan1=quan1/1000;
if npdes eq ‘TN0002356’ and year=2000 and month=7 and outfall=‘001’ then quan1=quan1/100;
if npdes eq ‘TN0003433’ and year=2000 and month=9 then quan1=quan1/1000 ;
if npdes eq ‘TN0020494’ and year=2002 and (month=2 or month=5 or month=6) then quan1=quan1/1000 ;
if npdes eq ‘TN0020613’ and year=2001 and month=8 then quan1=quan1/1000 ;
if npdes eq ‘TN0020672’ and year=2000 and month=7 then quan1=quan1/1000 ;
if npdes eq ‘TN0020702’ and year=2000 and month=7 then quan1=quan1/1000 ;
if npdes eq ‘TN0021164’ and year=2001 and month=6 then quan1=.028 ;
if npdes eq ‘TN0023001’ and year=2000 and month=12 then quan1=quan1/1000 ;
if npdes eq ‘TN0023353’ and year=2000 and month=10 then quan1=quan1/10000 ;
if npdes eq ‘TN0023469’ and year=2002 and month=12 then quan1=quan1/1000 ;
if npdes eq ‘TN0023477’ and year=2000 and month=7 then quan1=quan1/10;
if npdes eq ‘TN0024996’ and year=2001 and month=10 then quan1=quan1/1000 ;
if npdes eq ‘TN0025038’ and year=2002 and month=5 then quan1=2.5 ;
if npdes eq ‘TN0026247’ and year=2000 and month=10 then quan1=quan1/1000 ;
if npdes eq ‘TN0026506’ and year=2000 and month ne 3 then quan1=quan1/1000 ;
if npdes eq ‘TN0026506’ and year=2001 and (month=1 or month=2 or month=3 or month=11) then
quan1=quan1/1000 ;
if npdes eq ‘TN0026506’ and year=2002 and (month=3 or month=8 ) then quan1=quan1/1000 ;
if npdes eq ‘TN0026573’ and year=2001 and (month=5 or month=10 ) then quan1=quan1/1000 ;
if npdes eq ‘TN0026638’ and year=2001 and month=10 then quan1=quan1/1000 ;
if npdes eq ‘TN0028622’ and year=2001 and month=2 then quan1=quan1/1000 ;
if npdes eq ‘TN0058181’ and year=2001 and month=10 then quan1=quan1/1000 ;
if npdes eq ‘TN0060186’ and year=2002 and (month=12 or month=1 or month=2) then quan1=.103 ;
if npdes eq ‘TN0060186’ and year=2002 and (month=3 or month=4 or month=5) then quan1=.0855 ;
if npdes eq ‘TN0060186’ and year=2002 and (month=6 or month=7 or month=8) then quan1=.67 ;
if npdes eq ‘TN0060186’ and year=2002 and (month=9 or month=10 or month=11) then quan1=.095 ;
if npdes eq ‘TN0062057’ and year=2002 and month=1 then quan1=quan1/1000;
if npdes eq ‘TN0062294’ and year=2002 and (month=10 or month=11) then quan1=quan1/1000;
if npdes eq ‘TN0062499’ and year=2001 and month=9 then quan1=quan1/1000;
if npdes eq ‘TN0064912’ and year=2001 and month=4 then quan1=quan1/1000;
if npdes eq ‘TN0065358’ and year=2001 and month=2 then quan1=quan1/1000;
if npdes eq ‘TN0065501’ and year=2001 and month=2 then quan1=quan1/100000;
if npdes eq ‘TN0067423’ and year=2000 and month=10 then quan1=quan1/1000;
if not(npdes eq ‘TN0002356’);
if not(npdes eq ‘TN0002411’ );
if not(npdes eq ‘TN0000205’);
if not(npdes eq ‘TN0002640’);
if not(npdes eq ‘TN0002950’);
if not(npdes eq ‘TN0003671’);
if not(npdes eq ‘TN0022519’);
if npdes eq ‘TN0026506’ and year=2001 and month=2 then flag=0;
if not(npdes eq ‘TN0041939’);
if not(npdes eq ‘TN0057487’);
if npdes eq ‘TN0061387’ and year=2002 and month=7 and quan1=50000 then flag=0;
if not(npdes eq ‘TN0062120’);
if not(npdes eq ‘TN0074730’);
if npdes eq ‘TN0000060’ and year=2002 and month=10 and outfall = ‘009’ then quan1=quan1/1000;
if npdes eq ‘TN0000060’ and not(outfall=’009’) then flag=0;
if not(npdes eq ‘TN0003751’ );
if npdes eq ‘TN0020753’ and quan1 gt 1 then quan1=quan1/1000 ;
if npdes eq ‘TN0021644’ and year=2002 and month=9 then quan1=quan1/1000 ;
if npdes eq ‘TN0021717’ and year=2000 and month=10 then quan1=quan1/100000;
if npdes eq ‘TN0022560’ and quan1 ge 1 then quan1=quan1/10;
```

```
 if npdes eq 'TN0022586' and year=2000 and month=5 then quan1=quan1/10;
 if npdes eq 'TN0024171' and year=2002 and month=5 then quan1=quan1/10000;
 if npdes eq 'TN0024210' and not(outfall='001') then flag=0;
 if npdes eq 'TN0024953' and year=2001 and (month=7 or month=8) then quan1=quan1/100 ;
 if npdes eq 'TN0024970' and year=2000 and (month=10 or month=11 or month=12)then flag=0;
 if npdes = 'TN0025488' and year=2000 and month=11 then quan1=quan1/10;
 if npdes = 'TN0026166' and year=2001 and month=12 then quan1=quan1/10;
 if npdes = 'TN0026573' and quan1 gt 10 then quan1=quan1/1000;
 if npdes = 'TN0058181' and quan1 gt 1000 then quan1=quan1/1000;
 if npdes = 'TN0059226' and year=2002 and month=10 and outfall='004' then quan1=quan1/10;
 if npdes = 'TN0059226' and year=2002 and month=7 and outfall='004' then quan1=quan1/1000;
 if npdes = 'TN0060186' and year=2001 and month=12 then quan1=quan1/1000;
 if npdes = 'TN0064785' and quan1 gt 10 then quan1=quan1/100;
 if flag=1;
 c1flow=conc1;
 c2flow=conc2;
 c3flow=conc3;
 q1flow=quan1;
 q2flow=quan2;
 keep npdes outfall date year state c1flow c2flow c3flow q1flow q2flow;
run;

proc sort; by npdes outfall date; run;

data test.dmr_flow_093005;
 merge test.p600 test.p665 test.p50050;
 by npdes outfall date;
run;


proc sort data=test.dmr_flow_093005; by state year; run;

proc univariate noprint;
 var c1p600 c2p600 c3p600 q1p600 q2p600
 c1p665 c2p665 c3p665 q1p665 q2p665
 c1flow c2flow c3flow q1flow q2flow;
 by state year;
 output out=stat median=medc1p600 medc2p600 medc3p600 medq1p600 medq2p600
    medc1p665 medc2p665 medc3p665 medq1p665 medq2p665
                                  medc1flow medc2flow medc3flow medq1flow
medq2flow
               n=n_c1p600 n_c2p600 n_c3p600 n_q1p600 n_q2p600
              n_c1p665 n_c2p665 n_c3p665 n_q1p665 n_q2p665
                        n_c1flow n_c2flow n_c3flow n_q1flow n_q2flow;
run;

proc print; run;

PROC EXPORT DATA= work.stat
 OUTFILE= "Z:\Nutrients\SPARROW\point_source\PCS_donahoo_0805
\dmr_flow_stat_010406t.xls"
 DBMS=EXCEL REPLACE;
SHEET="stats";
RUN;



quit;
```

# Attachment 3: North Carolina Data Import Program

```
/* **************************************************************

program: nc_n_031305.sas
date: 3/9-14/05 10-11-05
based on: point92_data.sas (3-7/8-01)

************************************************************** */

libname test ‘z:\Nutrients\sparrow\point_source\nc\nitrogen’;
libname test1 ‘z:\Nutrients\sparrow\point_source\nc\flow’;
libname test2 ‘Z:\Nutrients\SPARROW\point_source\site_1005’;
options ps=54 ls=80;




/* read in nitrogen data files and combine */

/*
PROC IMPORT OUT= TEST.SUB0301n
 DATAFILE= “z:\Nutrients\SPARROW\point_source\nc\nitrogen\SUB0301n.xls”
 DBMS=EXCEL2000 REPLACE;
GETNAMES=YES;
RUN;

proc contents; run;

PROC IMPORT OUT= TEST.Sub0302n
 DATAFILE= “z:\Nutrients\SPARROW\point_source\nc\nitrogen\Sub0302n.xls”
 DBMS=EXCEL2000 REPLACE;
GETNAMES=YES;
RUN;

proc contents; run;


PROC IMPORT OUT= TEST.SUB0303n
 DATAFILE= “z:\Nutrients\SPARROW\point_source\nc\nitrogen\SUB0303n.xls”
 DBMS=EXCEL2000 REPLACE;
GETNAMES=YES;
RUN;

proc contents; run;

PROC IMPORT OUT= TEST.Sub0304n
 DATAFILE= “z:\Nutrients\SPARROW\point_source\nc\nitrogen\Sub0304n.xls”
 DBMS=EXCEL2000 REPLACE;
GETNAMES=YES;
RUN;

proc contents; run;
```

```

PROC IMPORT OUT= TEST.SUB0305n
 DATAFILE= “z:\Nutrients\SPARROW\point_source\nc\nitrogen\SUB0305n.xls”
 DBMS=EXCEL2000 REPLACE;
GETNAMES=YES;
RUN;

proc contents; run;


PROC IMPORT OUT= TEST.Sub0306n
 DATAFILE= “z:\Nutrients\SPARROW\point_source\nc\nitrogen\Sub0306n.xls”
 DBMS=EXCEL2000 REPLACE;
GETNAMES=YES;
RUN;
proc contents; run;


PROC IMPORT OUT= TEST.SUB0307n
 DATAFILE= “z:\Nutrients\SPARROW\point_source\nc\nitrogen\SUB0307n.xls”
 DBMS=EXCEL2000 REPLACE;
GETNAMES=YES;
RUN;
proc contents; run;


PROC IMPORT OUT= TEST.Sub0308n
 DATAFILE= “z:\Nutrients\SPARROW\point_source\nc\nitrogen\Sub0308n.xls”
 DBMS=EXCEL2000 REPLACE;
GETNAMES=YES;
RUN;
proc contents; run;


PROC IMPORT OUT= TEST.Sub0313n
 DATAFILE= “z:\Nutrients\SPARROW\point_source\nc\nitrogen\Sub0313n.xls”
 DBMS=EXCEL2000 REPLACE;
GETNAMES=YES;
RUN;
proc contents; run;


PROC IMPORT OUT= TEST.SUB04n
 DATAFILE= “z:\Nutrients\SPARROW\point_source\nc\nitrogen\SUB04n.xls”
 DBMS=EXCEL2000 REPLACE;
GETNAMES=YES;
RUN;
proc contents; run;


PROC IMPORT OUT= TEST.Sub05n
 DATAFILE= “z:\Nutrients\SPARROW\point_source\nc\nitrogen\Sub05n.xls”
 DBMS=EXCEL2000 REPLACE;
GETNAMES=YES;
RUN;
proc contents; run;
```

```
data test.nc_nitrogen_030805;
 set test.sub0301n test.sub0302n test.sub0303n test.sub0304n
 test.sub0305n test.sub0306n test.sub0307n test.sub0308n
 test.sub0313n test.sub04n test.sub05n;
run;

*/




/* combine flow and nitrogen data */
/*
proc contents data=test.nc_nitrogen_030805; run;
proc contents data=test1.nc_flow_030805; run;

data test.flow1;
 set test1.nc_flow_030805;
 flow_mgd=value;
 keep permit outfall date flow_mgd;
run;

data test.nitrogen;
 set test.nc_nitrogen_030805;
 units=UoM;
 keep facility permit outfall region subbasin parm value date month year units ;
run;

proc sort data=test.nitrogen;
 by permit outfall date;
run;

proc sort data=test.flow1;
 by permit outfall date;
 run;

 data test.tmp1;
 merge test.flow1 test.nitrogen;
by permit outfall date;
run;

proc print data=test.tmp1(obs=500); run;

data test.flow_n_030905;
 set test.tmp1;
 if value ne '.';
run;

proc contents; run;

proc print data=test.flow_n_030905(obs=500); run;
```

```
/* create tabular data set, 1 record per date */

/*
proc sort data=test.flow_n_030905; by parm units; run;

proc univariate data=test.flow_n_030905 noprint;
 var value;
 by parm units;
 output out=temp1 n=nobs_parm mean=mean_val;
run;

proc print; run;

data test.p600a;
 set test.flow_n_030905;
 if parm eq ‘P00600’;
 if units eq ‘mg/l’;
 p600_mgl=value;
 drop units value parm;
run;

proc sort data=test.p600a; by permit outfall date; run;

data test.p600b;
 set test.flow_n_030905;
 if parm eq ‘P00600’;
 if units eq ‘lbs/day’;
 p600lbdy=value;
 drop units value parm;
run;
proc sort data=test.p600b; by permit outfall date; run;

data test.p600c;
 set test.flow_n_030905;
 if parm eq ‘P00600’;
 if units eq ‘lbs/yr’;
 p600lbyr=value;
 drop units value parm;
run;
proc sort data=test.p600c; by permit outfall date; run;

data test.p610a;
 set test.flow_n_030905;
 if parm eq ‘P00610’;
 if units eq ‘mg/l’;
 p610_mgl=value;
 drop units value parm;
run;

proc sort data=test.p610a; by permit outfall date; run;

data test.p610b;
 set test.flow_n_030905;
 if parm eq ‘P00610’;
 if units eq ‘lbs/day’;
 p610lbdy=value;
```

```
 drop units value parm;
run;
proc sort data=test.p610b; by permit outfall date; run;

data test.p610c;
 set test.flow_n_030905;
 if parm eq ‘P00610’;
 if units eq ‘lbs/yr’;
 p610lbyr=value;
 drop units value parm;
run;
proc sort data=test.p610c; by permit outfall date; run;

data test.p625a;
 set test.flow_n_030905;
 if parm eq ‘P00625’;
 if units eq ‘mg/l’;
 p625_mgl=value;
 drop units value parm;
run;
proc sort data=test.p625a; by permit outfall date; run;

data test.p630a;
 set test.flow_n_030905;
 if parm eq ‘P00630’;
 if units eq ‘mg/l’;
 p630_mgl=value;
 drop units value parm;
run;
proc sort data=test.p630a; by permit outfall date; run;

data test.nc_nitrogen;
 merge test.p600a test.p600b test.p600c test.p610a test.p610b test.p610c test.p625a test.p630a;
 by permit outfall date;
run;


data test.nc_nitrogen;
 set test.nc_nitrogen;
 year=year(datepart(date));
 quarter=qtr(datepart(date));
 month=month(datepart(date));
 if p600_mgl gt 0 then screen = ‘p600_mgl’;
 if ( screen ne ‘p600_mgl’ and p600lbdy gt 0 ) then screen = ‘p600lbdy’;
 if ( screen = ‘ ’ and p600lbyr gt 0 ) then screen = ‘p600lbyr’;
 if ( screen = ‘ ’ and p610_mgl gt 0 ) then screen = ‘p610_mgl’;
 if ( screen = ‘ ’ and p610lbdy gt 0 ) then screen = ‘p610lbdy’;
 if ( screen = ‘ ’ and p610lbyr gt 0 ) then screen = ‘p610lbyr’;
 if ( screen = ‘ ’ and p625_mgl gt 0 ) then screen = ‘p625_mgl’;
 if ( screen = ‘ ’ and p630_mgl gt 0 ) then screen = ‘p630_mgl’;
 if ( screen = ‘p625_mgl’ and p630_mgl gt 0 ) then screen = ‘p625,30’;
run;

data test.nc_nitrogen;
 set test.nc_nitrogen;
 if screen ne ‘ ’;
```

```
    run;

/* convert p625 + p630 to p600 mg/l */
/*
data temp625_30;
 set test.nc_nitrogen;
 if screen = 'p625,30';
 p600_mgl = p625_mgl + p630;
 keep permit outfall date p600_mgl ;
run;

proc sort; by permit outfall date; run;

/* convert p600 lb/day to p600 mg/l */
/*
data temp600a;
 set test.nc_nitrogen;
 if screen = 'p600lbdy';
 p600_mgl =(p600lbdy/flow_mgd)*.264*.000001*453*1000;
 keep permit outfall date p600_mgl p600lbdy;
run;

proc sort; by permit outfall date; run;

/* convert p600 lb/year to p600 mg/l */
/*
data temp600b;
 set test.nc_nitrogen;
 if screen = 'p600lbyr';
 p600_mgl =((p600lbyr/365)/flow_mgd)*.264*.000001*453*1000;
 keep permit outfall date p600_mgl p600lbyr;
run;

proc sort; by permit outfall date; run;

/* convert p610lbdy to p610 mg/l */
/*
 data temp610a;
 set test.nc_nitrogen;
 if screen = 'p610lbdy';
 p610_mgl =(p610lbdy/flow_mgd)*.264*.000001*453*1000;
 keep permit outfall date p600_mgl p600lbdy;
run;

proc sort; by permit outfall date; run;

/* convert p610lbyr to p610 mg/l */
/*
 data temp610b;
 set test.nc_nitrogen;
 if screen = 'p610lbyr';
 p610_mgl =((p610lbyr/365)/flow_mgd)*.264*.000001*453*1000;
 keep permit outfall date p600_mgl p600lbyr;
run;

proc sort; by permit outfall date; run;
```

```
data test.nc_nitrogen;
 merge test.nc_nitrogen temp625_30 temp600a temp600b temp610a temp610b;
 by permit outfall date;
run;



proc contents data=test.nc_nitrogen; run;
proc print data=test.nc_nitrogen(obs=25); run;


/* median value and count for each parameter, by permit, outfall, year */
/*
data ckyr94;
 set test.nc_nitrogen;
 if year = 1994;
run;
proc sort data=ckyr94; by permit outfall year; run;

proc univariate data=ckyr94 noprint;
 var p600_mgl p610_mgl p625_mgl p630_mgl;
 by permit outfall year;
 output out=temp94 median=med600_mgl med610_mgl med625_mgl med630_mgl
n=n600_mgl n610_mgl n625_mgl n630_mgl;;
run;

data ckyr95;
 set test.nc_nitrogen;
 if year = 1995;
run;
proc sort data=ckyr95; by permit outfall year; run;

proc univariate data=ckyr95 noprint;
 var p600_mgl p610_mgl p625_mgl p630_mgl;
 by permit outfall year;
 output out=temp95 median=med600_mgl med610_mgl med625_mgl med630_mgl
n=n600_mgl n610_mgl n625_mgl n630_mgl;
run;

data ckyr96;
 set test.nc_nitrogen;
 if year = 1996;
run;
proc sort data=ckyr96; by permit outfall year; run;

proc univariate data=ckyr96 noprint;
 var p600_mgl p610_mgl p625_mgl p630_mgl;
 by permit outfall year;
 output out=temp96 median=med600_mgl med610_mgl med625_mgl med630_mgl
n=n600_mgl n610_mgl n625_mgl n630_mgl;
run;


data ckyr97;
 set test.nc_nitrogen;
```

```
 if year = 1997;
run;
proc sort data=ckyr97; by permit outfall year; run;

proc univariate data=ckyr97 noprint;
 var p600_mgl p610_mgl p625_mgl p630_mgl;
 by permit outfall year;
 output out=temp97 median=med600_mgl med610_mgl med625_mgl med630_mgl
n=n600_mgl n610_mgl n625_mgl n630_mgl;
run;


data ckyr98;
 set test.nc_nitrogen;
 if year = 1998;
run;
proc sort data=ckyr98; by permit outfall year; run;

proc univariate data=ckyr98 noprint;
 var p600_mgl p610_mgl p625_mgl p630_mgl;
 by permit outfall year;
 output out=temp98 median=med600_mgl med610_mgl med625_mgl med630_mgl
n=n600_mgl n610_mgl n625_mgl n630_mgl;
run;


data ckyr99;
 set test.nc_nitrogen;
 if year = 1999;
run;
proc sort data=ckyr99; by permit outfall year; run;

proc univariate data=ckyr99 noprint;
 var p600_mgl p610_mgl p625_mgl p630_mgl;
 by permit outfall year;
 output out=temp99 median=med600_mgl med610_mgl med625_mgl med630_mgl
n=n600_mgl n610_mgl n625_mgl n630_mgl;
run;


data ckyr00;
 set test.nc_nitrogen;
 if year = 2000;
run;
proc sort data=ckyr00; by permit outfall year; run;

proc univariate data=ckyr00 noprint;
 var p600_mgl p610_mgl p625_mgl p630_mgl;
 by permit outfall year;
 output out=temp00 median=med600_mgl med610_mgl med625_mgl med630_mgl
n=n600_mgl n610_mgl n625_mgl n630_mgl;
run;


data ckyr01;
 set test.nc_nitrogen;
```

```
 if year = 2001;
run;
proc sort data=ckyr01; by permit outfall year; run;

proc univariate data=ckyr01 noprint;
 var p600_mgl p610_mgl p625_mgl p630_mgl;
 by permit outfall year;
 output out=temp01 median=med600_mgl med610_mgl med625_mgl med630_mgl
n=n600_mgl n610_mgl n625_mgl n630_mgl;
run;


data ckyr02;
 set test.nc_nitrogen;
 if year = 2002;
run;
proc sort data=ckyr02; by permit outfall year; run;

proc univariate data=ckyr02 noprint;
 var p600_mgl p610_mgl p625_mgl p630_mgl;
 by permit outfall year;
 output out=temp02 median=med600_mgl med610_mgl med625_mgl med630_mgl
n=n600_mgl n610_mgl n625_mgl n630_mgl;
run;


data ckyr03;
 set test.nc_nitrogen;
 if year = 2003;
run;
proc sort data=ckyr03; by permit outfall year; run;

proc univariate data=ckyr03 noprint;
 var p600_mgl p610_mgl p625_mgl p630_mgl;
 by permit outfall year;
 output out=temp03 median=med600_mgl med610_mgl med625_mgl med630_mgl
n=n600_mgl n610_mgl n625_mgl n630_mgl;
run;


data test.n_check_yr;
 set temp94 temp95 temp96 temp97 temp98 temp99 temp00 temp01
 temp02 temp03 ;
run;

proc sort; by permit outfall year; run;

PROC EXPORT DATA= TEST.N_CHECK_yr
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\nc\Nitrogen\n_ch
eck_yrtmp.xls”
 DBMS=EXCEL2000 REPLACE;
RUN;



/* count number of quarters with data for each permit, by year and parameter */
```

```
/*
proc sort data=test.nc_nitrogen; by permit outfall year quarter; run;

proc univariate noprint;
 var p600_mgl p610_mgl p625_mgl p630_mgl;
 by permit outfall year quarter;
 output out=test.q1 n=q600_mgl q610_mgl q625_mgl q630_mgl;
run;

proc print data=test.q1(obs=100); var permit outfall year quarter q600_mgl
 q610_mgl q625_mgl q630_mgl; run;

data test.q2;
 set test.q1;
 if q600_mgl > 0 then q600_mgl = 1;
 if q610_mgl > 0 then q610_mgl = 1;
 if q625_mgl > 0 then q625_mgl = 1;
 if q630_mgl > 0 then q630_mgl = 1;
 keep permit outfall year quarter q600_mgl q610_mgl q625_mgl q630_mgl;
 run;

 proc sort; by permit outfall year; run;

 proc univariate noprint;
  var q600_mgl q610_mgl q625_mgl q630_mgl;
    by permit outfall year;
    output out=test.q3 sum=qs600_mgl qs610_mgl qs625_mgl qs630_mgl;
run;

data test.q4;
 set test.q3;
 keep permit outfall year qs600_mgl qs610_mgl qs625_mgl qs630_mgl;
run;

proc sort; by permit outfall year; run;

proc print data=test.q4(obs=100); run;


/* create final NC nitrogen point source data set; include site information
 updated in Oct 2005, that includes identification of SIC codes used
 in EPA Mississippi R. PS loads study */

proc sort data=test.nc_nitrogen; by permit outfall year; run;
proc sort data=test.q4; by permit outfall year; run;
proc sort data=test.n_check_yr; by permit outfall year; run;

data temppcs;
format permit $9.;
set test2.pcs_site_100505;
 permit=npdes;
 lat_pcs1=latitude;
 lat_pcs2=lat_0805;
 long_pcs1=longitude;
 long_pcs2=long_0805;
 if state = 'NC';
```

```
 keep permit npdes epa_noaa flow lat_pcs1 lat_pcs2 long_pcs1 long_pcs2 sic_code sic_desc ;
run;
proc sort; by permit; run;
proc contents; run;

data test.n_final_031405;
 merge test.nc_nitrogen test.q4 test.n_check_yr;
 by permit outfall year;
run;

proc sort; by permit; run;

data test.n_final_031405;
 merge temppcs test.n_final_031405 ;
 by permit;
run;

proc print data=test.n_final_031405(obs=50); var permit outfall lat_pcs sic_pcs permit_flo; run;

proc contents; run;

quit;
```

# Attachment 4: Data Preparation Program

```
/* **************************************************************************

program: nc_loadprep_021606.sas
date: 10/13-20/05 11/9/05 1/5/06 1/12/06 1/25/06 2/6/06 2/16/06 2/23/06
 5/5/06 6/12/06 6/19/06 6/23/06



************************************************************************** */

libname test ‘Z:\Nutrients\SPARROW\point_source\load_1005\NC’;
libname ncn ‘z:\Nutrients\sparrow\point_source\nc\nitrogen’;
libname ncp ‘z:\Nutrients\sparrow\point_source\nc\phosphorus’;
libname sc ‘z:\Nutrients\sparrow\point_source\sc\Flow’;
libname pcs ‘z:\nutrients\sparrow\point_source\PCS_donahoo_0805’;
libname site ‘Z:\Nutrients\SPARROW\point_source\site_1005’;
libname ncsite ‘Z:\Nutrients\SPARROW\point_source\nc\site’;
libname summary ‘Z:\Nutrients\SPARROW\point_source\load_1005\summary’;
libname noaa ‘Z:\GIS\PointSources\NOAA_1991’;
libname tpc ‘Z:\Nutrients\SPARROW\point_source\load_1005\Rubin_0206’;

options ps=54 ls=80;


/*
proc contents data=ncn.nc_nitrogen; run;
proc contents data=ncp.nc_phosphorus;run;
proc contents data=site.pcs_site_101205;run;*/

/* NC data — input flow, 600 and 665 data from NC DEHNR, summarize values by month to deal
with cases where there is more than one observation per month */
/*
data ncn;
format npdes $9.;
 set ncn.nc_nitrogen;
 npdes=permit;
 month=month(datepart(date));
 year=year(datepart(date));
 quarter=qtr(datepart(date));
 keep facility permit npdes outfall year quarter month p600_mgl flow_mgd;
run;
proc sort; by facility npdes permit outfall year quarter month ; run;


proc univariate noprint;
 var p600_mgl flow_mgd;
 by facility npdes permit outfall year quarter month ;
 output out=ncn1 median=med600 medflow;
run;

data ncn2;
 set ncn1;
p600_mgl=med600;
```

```
 flow_mgd=medflow;
 drop med600 medflow;
run;

proc sort; by npdes outfall year month; run;

data ncp;
format npdes $9.;
 set ncp.nc_phosphorus;
 npdes=permit;
 month=month(datepart(date));
 year=year(datepart(date));
 quarter=qtr(datepart(date));
 keep facility permit npdes outfall year quarter month p665_mgl flow_mgd;
run;

proc sort; by facility npdes permit outfall year quarter month ; run;

proc univariate noprint;
 var p665_mgl flow_mgd;
 by facility npdes permit outfall year quarter month ;
 output out=ncp1 median=med665 medflow;
run;

data ncp2;
 set ncp1;
 p665_mgl=med665;
 flow_mgd=medflow;
 drop med665 medflow;
run;

proc sort; by npdes outfall year month; run;

data test.ncdmr_101305;
 merge ncn2 ncp2;
 by npdes outfall year month;
run;

data test.ncdmr_101305;
 format outfall $4. date_nc datetime20.;
 set test.ncdmr_101305;
 if month = 12 or month = 1 or month = 2 then season = ‘winter’;
 if month = 5 or month = 3 or month = 4 then season = ‘spring’;
 if month = 6 or month = 7 or month = 8 then season = ‘summer’;
 if month = 11 or month = 9 or month = 10 then season = ‘fall’;
 nc_data=1;
run;

proc sort data=test.ncdmr_101305; by npdes outfall year month;; run;

proc print data=test.ncdmr_101305(obs=500); run;


/* combine NC and PCS data and sitefile data and create
 season class variable */
/*
```

```
data test.nc_pcs1;
 format date datetime20.;
 set pcs.dmr_flow_093005;
 month=month(datepart(date));
 year=year(datepart(date));
 quarter=qtr(datepart(date));
 if state = ‘NC’;
 keep npdes outfall year quarter month q1flow c2p600 c2p665;
run;

proc sort ; by npdes outfall year quarter month; run;

proc univariate noprint;
 var c2p600 c2p665 q1flow;
 by npdes outfall year quarter month ;
 output out=ncp1 median=med600 med665 medflow;
run;

data test.temp1;
 set ncp1;
 if month = 12 or month = 1 or month = 2 then season = ‘winter’;
 if month = 5 or month = 3 or month = 4 then season = ‘spring’;
 if month = 6 or month = 7 or month = 8 then season = ‘summer’;
 if month = 11 or month = 9 or month = 10 then season = ‘fall’;
 c2p600=med600;
 c2p665=med665;
 q1flow=medflow;
 pcs_data=1;
 drop med600 med665 medflow;
run;

proc sort; by npdes outfall year month; run;
proc sort data=test.ncdmr_101305; by npdes outfall year month;; run;


data test.nc_pcs2;
 merge test.ncdmr_101305 test.temp1;
 by npdes outfall year month;
run;


proc sort data=test.nc_pcs2; by npdes; run;

data site;
 set site.pcs_site_101405;
 if state=‘NC’;
run;

data site2;
 set site;
 keep facility nc_facility npdes sic_code sic_desc epa_noaa permit_flo;
run;

proc sort; by npdes; run;
```

```
data test.temp78;
 format npdes $9.;
 set ncsite.nc_site_101105;
 npdes=permit;
run;

proc sort data=test.temp78; by npdes; run;

data test.nc_pcs3;
 merge test.nc_pcs2 site2 test.temp78;
 by npdes;
run;

data test.nc_pcs4;
 set test.nc_pcs3;
 drop med600mgl med665mgl medflow_n medflow_p date_nc nobs600 nobs665 nobsnflow nobspflow;
run;

proc print data=test.nc_pcs4(obs=500); var npdes outfall month year season nc_data p600_mgl
flow_mgd pcs_data c2p600 q1flow sic_code; run;

/* define single flow variable and subset data for 2000-2002 and epa_noaa=1 and flowtype ne ‘4’
this assumes that flow_mgd is the appropriate flow variable from the state data
and q1flow is the appropriate flow data from the PCS data. This needs to be determined on a
state-by-state basis. Also assumption is made to use the state data in preference to the PCS data,
where both available. Data edits done after 2/23/06 checking; further editing (5/5/06) done
pursuant to error noted by Mike templeton for NC6033. Additional editing done based on results
of flowcheck_050506.sas program, which identified several discharge records that needed
to be deleted or edited. */
/*
data temp99;
 set test.nc_pcs4;
 if npdes eq ‘NC0059251’ and year=2002 and month = 4 ; flow_mgd = .017;
 run;
proc sort; by outfall year month; run;

proc print; var npdes outfall year month q1flow flow_mgd name sic_code; run;


data test.nc_pcs5;
 format flowtype $2. flowclass $15.;
 set test.nc_pcs4;
 if nc_facility eq ‘’ then nc_facilty = facility;
 if npdes =‘NC0064149’ and year=2002 and month = 8 then flow_mgd=flow_mgd/1000;
 if npdes eq ‘NC0006033’ and year = 2002 and (month = 11 or month = 12 ) then
    p600_mgl=p600_mgl/10;
 if not ( npdes eq ‘NC0003255’ and year = 2002 and month = 11 and outfall = ‘005’ );
 if npdes eq ‘NC0029131’ and year = 2000 and (month=6 or month = 7) then flow_mgd = .015;
 if not ( npdes eq ‘NC0030180’ and year = 2002);
 if npdes eq ‘NC0052469’ and year=2000 and (month=1 or month=2 or month=3 or month=10 or
    month=11 or month=12) then q1flow=1.254 ;
 if npdes eq ‘NC0059251’ and year=2002 and month = 4 then flow_mgd = .017;
 if flow_mgd gt 0 and (q1flow eq ‘.’ or q1flow eq ‘’) then flowtype = ‘1’;
 if ( flow_mgd eq ‘.’ or flow_mgd eq ‘’) and q1flow gt 0 then flowtype = ‘2’;
 if flow_mgd gt 0 and q1flow gt 0 then flowtype = ‘3’;
 if ( flow_mgd eq 0 or flow_mgd eq ‘.’ or flow_mgd eq ‘’ )
```

```
    and (q1flow eq ‘.’ or q1flow eq 0 or q1flow eq ‘’) then flowtype = ‘4’;
 if flowtype = ‘1’ then flow=flow_mgd;
 if flowtype = ‘2’ then flow = q1flow;
 if flowtype = ‘3’ then flow=flow_mgd;
 if flowtype = ‘4’ then flow=‘.’;
 if flow gt 0 and flow le 0.05 then flowclass = ‘1’;
 if flow gt .05 and flow le .2 then flowclass = ‘2’;
 if flow gt .2 and flow le 1 then flowclass = ‘3’;
 if flow gt 1 and flow le 5 then flowclass = ‘4’;
 if flow gt 5 then flowclass = ‘5’;
 if year ge 2000 and year le 2002;
 if epa_noaa = 1;
 run;

 proc sort; by flowtype; run;

 proc univariate noprint;
 var flow flow_mgd q1flow;
 by flowtype;
 output out=flowstat median=medflo medflomgd medq1flo n=nflo nflomgd nq1flo;
run;

proc print data=flowstat; run;


data test.nc_pcs5; set test.nc_pcs5; if flowtype ne ‘4’; run;



/* analyze p600 and p665 data from state of NC and from PCS and create single, unified values
 for 600 and 665. This assumes that p600_mgl and p665_mgl are the appropriate concentration
 variables from the NC state data and c2p600 and c2p665 are the appropriate concentration data
 from the PCS data. This needs to be determined on a state-by-state basis. Also assumption
 is made to use the NC data in preference to the PCS data, where both available. Data edits
 made after 2/23/06 data checking */

/*
data test.nc_pcs6;
 format p600type $2. p665type $2.;
 set test.nc_pcs5;
 if npdes =‘NC0000272’ and year=2002 and month = 12 then p600_mgl=p600_mgl/10;
 if p600_mgl gt 0 and (c2p600 eq ‘.’ or c2p600 eq 0) then p600type = ‘1’;
 if p665_mgl gt 0 and (c2p665 eq ‘.’ or c2p665 eq 0) then p665type = ‘1’;
 if ( p600_mgl eq ‘.’ or p600_mgl eq ‘’) and c2p600 gt 0 then p600type = ‘2’;
 if ( p665_mgl eq ‘.’ or p665_mgl eq ‘’) and c2p665 gt 0 then p665type = ‘2’;
 if p600_mgl gt 0 and c2p600 gt 0 then p600type = ‘3’;
 if p665_mgl gt 0 and c2p665 gt 0 then p665type = ‘3’;
 if (p600_mgl eq 0 or p600_mgl eq ‘.’) and (c2p600 eq ‘.’ or c2p600 eq 0) then p600type = ‘4’;
 if (p665_mgl eq 0 or p665_mgl eq ‘.’) and (c2p665 eq ‘.’ or c2p665 eq 0) then p665type = ‘4’;
run;


data test.nc_pcs7;
 format type600 $10. type665 $10.;
 set test.nc_pcs6;
 if p600type = ‘1’ or p600type = ‘3’ then do;
```

```
 p600=p600_mgl; orig600=1; type600='orig600'; end;
 if p665type = '1' or p665type = '3' then do;
 p665 = p665_mgl; orig665=1; type665='orig665'; end;
 if p665type = '2' then do;
 p665=c2p665; orig665=1; type665='orig665'; end;
run;

proc univariate noprint;
 var p600 p665;
 output out=test.nc600665 p99=p600_99 p665_99 p90=p600_90 p665_90 q3=p600_75 p665_75
 median=p600_50 p665_50 q1=p600_25 p665_25 p10=p600_10 p665_10 p1=p600_01 p665_01
 mean=p600_mean p665_mean;
run;

data test.nc600665;
 set test.nc600665;
 state='NC';
run;

proc print data=test.nc600665;
 title 'Univariate stats for p600 and p665 and flow based on NC or PCS original data';
run;



/* summary stats for p600/665 */
/*
proc sort data=test.nc_pcs7; by flowclass sic_code; run;

proc univariate noprint;
 var p600 p665;
 by flowclass sic_code;
 output out=p600665 median=med600 med665 n=n_600 n_665;
run;

data p600665;
 format type600 $10. type665 $10.;
 set p600665;
 if med600 ne '.' or med665 ne '.';
 if med600 gt 0 then type600 = 'p600';
 if med665 gt 0 then type665 = 'p665';
run;

proc sort; by sic_code flowclass type600 type665; run;

proc print data=p600665;
 title 'Summary stats for p600/665 observations using p600/665 ';
run;


/* Create seasonal median p600 p665 values for each permit/outfall (across 2000-2002 data)
and merge back into data set. Median seasonal values will be used in a couple of instances,
including where there is no value for p600/p665 from original NC and PCS data and where
individual monthly values are VERY large (e.g. gt 10 times the median seasonal value).
The goal is ultimatley to have a p600/p665 value for each record where there is a flow value
(all of the records have SIC that are associated with TN and TP discharge, per EPA study)
```

```
Seasonal median values for 600/665/flow also will be used in load calculations when there is not
12 months of flow data for a given year

Also note that there are observations where flow = 0 or flow = blank and where p600/665 (either
original PCS data for medsea values) is gt 0. In these cases it is assumed that flow exists for
that NPDES/outfall. A median seasonal flow value across the 2000-2002 time period is developed
and substituted where flow = 0 or blank and there is a positive value for p600/665 for that discharger. */

/*
proc sort data=test.nc_pcs7; by npdes outfall season; run;

proc univariate noprint;
 var p600 p665 flow;
 by npdes outfall season;
 output out=median1 median=medsea600 medsea665 medseaflo;
run;

proc sort data=median1; by npdes outfall season; run;
proc sort data=test.nc_pcs7; by npdes outfall season; run;

data test.nc_pcs8;
 merge test.nc_pcs7 median1;
 by npdes outfall season;
run;

data test.nc_pcs8;
 format type600 $10. type665 $10.;
 set test.nc_pcs8;
 if p600 = ‘.’ and medsea600 gt 0 then do;
 type600 = ‘medsea600’; sea600=1; p600=medsea600; end;
 if p665 = ‘.’ and medsea665 gt 0 then do;
 type665=‘medsea665’; sea665=1; p665=medsea665; end;
 if flowclass eq ‘ ’ and ( p600 gt 0 or p665 gt 0 ) then flowsub = 1;
 if flowsub = 1 then flow = medseaflo;
run;

/* analyze which records have no values for p600/665 */

data temp55;
 set test.nc_pcs8;
 if flow gt 0 and p600 eq ‘.’ ;
run;

proc univariate noprint;
 var p600 ;
 output out=tempn nmiss=n_no600;
run;

data temp56;
 set test.nc_pcs8;
 if flow gt 0 and p665 eq ‘.’ ;
run;

proc univariate noprint;
 var p665 ;
 output out=tempp nmiss=n_no665;
```

```
run;

data temp57;
 set tempn tempp;
run;

proc print; title ‘nobs of records with no p600/p665 after using medsea ‘; run;



/* analyze p600 p665 and flow data by flowclass, SIC and season (2000-2002 data) based
 on combined MRB2 state and PCS data for records with p600/665 conentration data,
 to create median values of 600 and 665 to use when there is no data from original data
 (p600/665) or median data (by permit outfall season; medsea600/665. These TPC values
 are created with Z:\Nutrients\SPARROW\point_source\load_1005\summary\flowsicsea_060906.sas.
 Median values are based on at least 5 observations */

/*
proc sort data=test.nc_pcs8; by flowclass sic_code season; run;
proc sort data=summary.flosicsea; by flowclass sic_code season; run;

data test.nc_pcs9;
 merge test.nc_pcs8 summary.flosicsea;
 by flowclass sic_code season;
run;

data test.nc_pcs9;
 format type600 $10. type665 $10.;
 set test.nc_pcs9;
 if p600=‘.’ and flosicsea600 gt 0 then do;
 type600=‘fss600’; fss_600=1; p600=flosicsea600; end;
 if p665=‘.’ and flosicsea665 gt 0 then do;
 type665=‘fss665’; fss_665=1; p665=flosicsea665; end;
run;

/* analyze which records have no values for p600/665 after flosicsea */

data temp55;
 set test.nc_pcs9;
 if flow gt 0 and p600 eq ‘.’ ;
run;

proc univariate noprint;
 var p600 ;
 output out=tempn nmiss=n_no600;
run;

data temp56;
 set test.nc_pcs9;
 if flow gt 0 and p665 eq ‘.’ ;
run;

proc univariate noprint;
 var p665 ;
 output out=tempp nmiss=n_no665;
run;
```

```

data temp57;
 set tempn tempp;
run;

proc print; title ‘nobs of records with no p600/p665 after flosicsea ‘; run;

/* For situations where there is not an original value for p600/665 and there is not a value from
 flosicsea600/665, ./load_1005/summary/flowsicsea_060906.sas analyzes original p600/665 and flow
 data by flowclass and SIC (2000-2002 data) to create median seasonal values of 600 and 665 by
 flow class/SIC-code to use when there is no p600/665 data . A minimum of 5 observations must be
 present to use a median value for a flowclass/sic_code. */

/*
proc sort data=test.nc_pcs9; by flowclass sic_code ; run;
proc sort data=summary.flosic; by flowclass sic_code ; run;

data test.nc_pcs9a;
 merge test.nc_pcs9 summary.flosic;
 by flowclass sic_code ;
run;

data test.nc_pcs9b;
 format type600 $10. type665 $10.;
 set test.nc_pcs9a;
 if p600=‘.’ and flosic600 gt 0 then do;
 type600=‘fs600’; fs_600=1; p600=flosic600; end;
 if p665=‘.’ and flosic665 gt 0 then do;
 type665=‘fs665’; fs_665=1; p665=flosic665; end;
run;

/* analyze which records have no values for p600/665 */

data temp55;
 set test.nc_pcs9b;
 if flow gt 0 and p600 eq ‘.’ ;
run;

proc univariate noprint;
 var p600 ;
 output out=tempn nmiss=n_no600;
run;

data temp56;
 set test.nc_pcs9b;
 if flow gt 0 and p665 eq ‘.’ ;
run;

proc univariate noprint;
 var p665 ;
 output out=tempp nmiss=n_no665;
run;

data temp57;
 set tempn tempp;
run;
```

```

proc print; title 'nobs of records with no p600/p665 after flosic '; run;


/* Use ./load_1005/summary/flowsicsea_060906.sas to create "typical
facility concentration" (TPC) for TN and TP, using
several data sources, to calculate loads when flow data exists and no
other source of concentration data is available. Primary sources of TPC
data is from a Tetra Tech analysis done for EPA, with data provided by
EPA's Steve Rubin. Also SIC-based TPC are developed from composite MRB2
data for this study. The lower of the nutrient TPC from the EPA/Tetra Tech
and the NC/SC studies are used as the TPC. */


proc sort data=test.nc_pcs9b; by sic_code ; run;

proc sort data=summary.mrb2_tpc_022306; by sic_code ; run;

data test.nc_pcs9c;
 merge test.nc_pcs9b summary.mrb2_tpc_022306;
 by sic_code ;
run;

data test.nc_pcs9d;
 format type600 $10. type665 $10.;
 set test.nc_pcs9c;
 if p600='.' and tpc600 gt 0 then do;
 type600='tpc600'; tpc_600=1; p600=tpc600; end;
 if p665='.' and tpc665 gt 0 then do;
 type665='tpc665'; tpc_665=1; p665=tpc665; end;
run;

/* analyze which records have no values for p600/665 after tpc */

data temp55;
 set test.nc_pcs9d;
 if flow gt 0 and p600 eq '.' ;
run;

proc univariate noprint;
 var p600 ;
 output out=tempn nmiss=n_no600;
run;

data temp56;
 set test.nc_pcs9d;
 if flow gt 0 and p665 eq '.' ;
run;

proc univariate noprint;
 var p665 ;
 output out=tempp nmiss=n_no665;
run;

data temp57;
 set tempn tempp;
```

```
run;

proc print; title ‘nobs of records with no p600/p665 after TPC ‘; run;


/* count number of months and quarters with data, by permit/outfall, for 2000-2002 */

proc sort data=test.nc_pcs9d; by npdes outfall year ; run;

proc univariate noprint;
 var flow;
 by npdes outfall year ;
 output out=mon_flow n=mon_flow;
run;

proc sort data=test.nc_pcs9d; by npdes outfall year quarter ; run;

proc univariate noprint;
 var flow;
 by npdes outfall year quarter ;
 output out=qtr1 n=qtr_flow;
run;

data qtr2;
 set qtr1;
 if qtr_flow>0 then qtr_flow=1;
run;

proc sort; by npdes outfall year; run;

proc univariate noprint;
 var qtr_flow;
 by npdes outfall year;
 output out=qtrstat sum=qtrflow;
run;

proc sort data=qtrstat; by npdes outfall year; run;
proc sort data=mon_flow; by npdes outfall year; run;
proc sort data=test.nc_pcs9; by npdes outfall year ; run;

data test.nc_pcs10;
 merge qtrstat mon_flow test.nc_pcs9d;
 by npdes outfall year ;
run;
proc contents data=test.nc_pcs10; run;


/* create final data set, replacing very large values of p600/665 with median values
(by npdes/outfall) First, look at distribution of p600/665. To be replaced, values of p600/665
will have to be greater than 10 times the size of the median concentration value for all values
for that discharger/outfall during 2000-2002 and greater than the 95th percentile value of all
p600/665 values in the entire data set. */

proc sort data=test.nc_pcs10; by npdes outfall; run;

proc univariate data=test.nc_pcs10;
```

```
 var p600 p665;
run;

/* TN stats */

proc sort data=test.nc_pcs10; by npdes outfall; run;

data tn;
 set test.nc_pcs10;
 if p600 gt 0;
run;

proc univariate noprint;
 var p600;
 by npdes outfall;
 output out=temptn median=median_p600;
run;

proc sort data=temptn; by npdes outfall; run;

/* TP stats */

proc sort data=test.nc_pcs10; by npdes outfall; run;

data tp;
 set test.nc_pcs10;
 if p665 gt 0;
run;

proc univariate noprint;
 var p665;
 by npdes outfall;
 output out=temptp median=median_p665;
run;

proc sort data=temptp; by npdes outfall; run;

data test.temp88;
 merge test.nc_pcs10 temptn temptp;
 by npdes outfall;
run;


data test.nc_pcs11;
 format type600 $10. type665 $10. high_tn $10. high_tp $10.;
 set test.temp88;
 state =‘NC’;
 if p600 gt (10*median_p600) then high_tn=‘10*med_tn’;
 if p665 gt (10*median_p665) then high_tp=‘10*med_tp’;
 if high_tn=‘10*med_tn’ and p600 gt 37.1 then do;
 p600=median_p600; hi600=1; type600=‘hi600’; end;
 if high_tp=‘10*med_tp’ and p665 gt 6.6 then do;
 p665=median_p665; hi665=1; type665=‘hi665’; end;
run;

quit;
```

# Attachment 5: Typical Pollutant-Concentration Program

```
/* ************************************************************************

program : flowsicsea_060906.sas
date: 6/9/2006
from: flowsicsea_012706.sas (4/21/06)

This program is used to develop estimates of p600/665 based on composite data sets.
flosicsea600/665 variables are median p600/665 values by flow class, SIC code,
and season, calculated using NC, VA, GA, MS, AL, TN, and SC data.
flowsic600/665 are median p600/665 values by flow class and SIC code. The program was revised
(6/9/06)
to incorporate code from rubin_tpc_022206.sas, allowing calculation of all SIC TPC values
in this program.

*************************************************************************************** */

libname test 'Z:\Nutrients\SPARROW\point_source\load_1005\summary';
libname ncn 'z:\Nutrients\sparrow\point_source\nc\nitrogen';
libname ncp 'z:\Nutrients\sparrow\point_source\nc\phosphorus';
libname sc 'z:\Nutrients\sparrow\point_source\sc\Flow';
libname pcs 'z:\nutrients\sparrow\point_source\PCS_donahoo_0805';
libname site 'Z:\Nutrients\SPARROW\point_source\site_1005';
libname ncsite 'Z:\Nutrients\SPARROW\point_source\nc\site';
libname scsite 'Z:\Nutrients\SPARROW\point_source\sc\site_1005';
libname ncdata 'Z:\Nutrients\SPARROW\point_source\load_1005\NC';
libname fldata 'Z:\Nutrients\SPARROW\point_source\load_1005\FL';
libname scdata 'Z:\Nutrients\SPARROW\point_source\load_1005\SC';
libname msdata 'Z:\Nutrients\SPARROW\point_source\load_1005\MS';
libname aldata 'Z:\Nutrients\SPARROW\point_source\load_1005\al';
libname gadata 'Z:\Nutrients\SPARROW\point_source\load_1005\ga';
libname tndata 'Z:\Nutrients\SPARROW\point_source\load_1005\TN';
libname vadata 'Z:\Nutrients\SPARROW\point_source\load_1005\va_ky';
libname rubin 'Z:\Nutrients\SPARROW\point_source\load_1005\Rubin_0206';

/* analyze p600 p665 and flow data by flow class, SIC and season (2000-2002 data)
from SC, NC, VA, GA, AL, MS, and TN to create median seasonal values of 600 and 665 by flow class
to use when there is no year/month-specific original data (p600/665) or median data
(by permit outfall season; medsea600/665. A minimum of 5 observations must be
present to estimate a seasonal value for a flow class/sic_code. Use year-month-specific
data from states not data summarized by season (i.e. nc_pcs7 data sets).
     */

data sctemp;
 set scdata.nc_pcs7a;
 keep npdes outfall year season sic_code p600 p665 flow flow class;
run;

data nctemp;
 set ncdata.nc_pcs7;
 keep npdes outfall year season sic_code p600 p665 flow flow class;
run;

data vatemp;
```

```
     set vadata.nc_pcs7;
 keep npdes outfall year season sic_code p600 p665 flow flow class;
run;

data tntemp;
 set tndata.nc_pcs7;
 keep npdes outfall year season sic_code p600 p665 flow flow class;
run;

data mstemp;
 set msdata.nc_pcs7;
 keep npdes outfall year season sic_code p600 p665 flow flow class;
run;

data gatemp;
 set gadata.nc_pcs7;
 keep npdes outfall year season sic_code p600 p665 flow flow class;
run;

data altemp;
 set aldata.nc_pcs7;
 keep npdes outfall year season sic_code p600 p665 flow flow class;
run;

data test.ncscfl1;
 set sctemp nctemp vatemp gatemp mstemp altemp tntemp;
run;

/* create flowsicsea variables. Examine values of flosicsea600/665 gt 75th pctl */

proc sort data=test.ncscfl1 ; by flow class sic_code season; run;

proc univariate noprint;
 var p600 p665;
 by flow class sic_code season;
 output out=flosicsea1 median=flosicsea600 flosicsea665
 n=n600flosicsea n665flosicsea ;
run;

data flosicsea;
 set flosicsea1;
 if flosicsea600 gt 0 or flosicsea665 gt 0;
run;

data flosicsea1;
 set flosicsea;
 if n600flosicsea ge 5 ;
 keep flow class sic_code season flosicsea600 n600flosicsea;
run;

proc sort; by flow class sic_code season; run;

data flosicsea2;
 set flosicsea;
 if n665flosicsea ge 5 ;
 keep flow class sic_code season flosicsea665 n665flosicsea;
```

```
run;

proc sort; by flow class sic_code season; run ;

data test.flosicsea;
 merge flosicsea1 flosicsea2;
 by flow class sic_code season;
run;

proc print; ‘MRB2 regional p600/665 values — FLO/SIC/SEA’; run;


/* create flowsic variables */

proc sort data=test.ncscfl1 ; by flow class sic_code ; run;

proc univariate noprint;
 var p600 p665;
 by flow class sic_code ;
 output out=flosic1 median=flosic600 flosic665
 n=n600flosic n665flosic ;
run;

data flosic;
 set flosic1;
 if flosic600 gt 0 or flosic665 gt 0;
run;

data flosic1;
 set flosic;
 if n600flosic ge 5 ;
 keep flow class sic_code flosic600 n600flosic;
run;

proc sort; by flow class sic_code ; run;

data flosic2;
 set flosic;
 if n665flosic ge 5 ;
 keep flow class sic_code flosic665 n665flosic;
run;

proc sort; by flow class sic_code ; run ;

data test.flosic;
 merge flosic1 flosic2;
 by flow class sic_code ;
run;

proc print; title ‘MRB2 regional p600/665 values — FLO/SIC’; run;

/* create sic values */

proc sort data=test.ncscfl1 ; by sic_code ; run;

proc univariate noprint;
```

```
 var p600 p665;
 by sic_code ;
 output out=sic1 median=sic600 sic665
 n=n600sic n665sic ;
run;

data sic;
 set sic1;
 if sic600 gt 0 or sic665 gt 0;
run;

data sic1;
 set sic;
 if n600sic ge 5 ;
 keep sic_code sic600 n600sic;
run;

proc sort; by sic_code ; run;

data sic2;
 set sic;
 if n665sic ge 5 ;
 keep sic_code sic665 n665sic;
run;

proc sort; by sic_code ; run ;

data test.sic042106;
 merge sic1 sic2;
 by sic_code ;
run;

proc print; title 'MRB2 regional p600/665 values — SIC'; run;


/* This is the code incorporated from rubin_tpc_022206.sas.

import EPA/Tetra Tech TPC data from Steve Rubin 2/22/06 */


PROC IMPORT OUT= WORK.epa_tpc
 DATAFILE= "Z:\Nutrients\SPARROW\point_source\load_1005\Rubin
_0206\NEWTPC4.XLS"
 DBMS=EXCEL REPLACE;
SHEET="MRB2";
GETNAMES=YES;
MIXED=NO;
SCANTEXT=YES;
USEDATE=YES;
SCANTIME=YES;
RUN;

data epa_tpc;
 format sic_code $4.;
 set epa_tpc;
 sic_code=sic;
```

```
 drop sic2 sic sicdesc_epa tn_epa tp_epa;
run;

proc sort nodupkey ; by sic_code; run;
proc contents; run;
proc print; run;

/* import sic code/epa_noaa information and merge with pcs site information and merge */


PROC IMPORT OUT= work.siccode
 DATAFILE= "Z:\GIS\PointSources\PCS_site_0205\sic_codes.xls"
 DBMS=EXCEL2000 REPLACE;
GETNAMES=YES;
RUN;

data sic_code;
 format sic_code $4.;
 set siccode;
 sic_code=sic3;
 drop sic_i sic3 sic;
run;

proc sort; by sic_code; run;

data test.epa_tpc;
 merge epa_tpc sic_code;
 by sic_code;
run;

data test.epa_tpc;
 set test.epa_tpc;
 if epa_noaa=1;
run;

Proc sort; by sic_code; run;

proc contents; run;
proc print; title 'Rubin-based SIC TPC'; run;

/* import MRB2 states-based sic TPC and merge */

proc sort data=test.sic042106; by sic_code; run;

data temp;
 merge test.epa_tpc test.sic042106;
 by sic_code;
run;

data test.mrb2_tpc_022306;
 set temp;
 tpc600=epa_tn;
 tpc665=epa_tp;
 if epa_tn ='.' then tpc600=sic600;
 if epa_tp='.' then tpc665=sic665;
 if sic600 gt 0 and epa_tn gt sic600 then tpc600=sic600;
```

```
 if sic665 gt 0 and epa_tp gt sic665 then tpc665=sic665;
 drop epa_noaa count n600sic n665sic;
run;

proc print; title ‘TN and TP concentrations, combined EPA national database and MRB2 PCS database’;
run;



/* EXPORT FLOWSICSEA AND FLOSIC TABLES TO EXCEL */

PROC EXPORT DATA= TEST.FLOSIC
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\flosicsea_060906t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“flosic”;
RUN;

PROC EXPORT DATA= TEST.flosicsea
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\flosicsea_060906t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“flosicsea”;
RUN;

PROC EXPORT DATA= test.mrb2_tpc_022306
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\flosicsea_060906t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“sic_tpc”;
RUN;
```

# Attachment 6: Load Calculation Program

```
/* ******************************************************************************

program: nc_loadcalc_021606.sas
date: 10/13-19/05 11/3/05 11/8-9/05 11-17-05 1-13-06 1-20-06 2/6-23/06
    3/21/06 6/9/06 6/19/06

****************************************************************************** */

libname test 'Z:\Nutrients\SPARROW\point_source\load_1005\NC';
libname ncn 'z:\Nutrients\sparrow\point_source\nc\nitrogen';
libname ncp 'z:\Nutrients\sparrow\point_source\nc\phosphorus';
libname summary 'Z:\Nutrients\SPARROW\point_source\load_1005\summary';
libname pcs 'z:\nutrients\sparrow\point_source\PCS_donahoo_0805';
libname site 'Z:\Nutrients\SPARROW\point_source\site_1005';
libname ncsite 'Z:\Nutrients\SPARROW\point_source\nc\site';

options ps=54 ls=80;

/* Calculate loads for permits/outfall with 12 months of flow */

data test.temp1;
 set test.nc_pcs11;
 if mon_flow=12;
 if month = 1 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 2 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*28;
 if month = 3 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 4 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*30;
 if month = 5 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 6 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*30;
 if month = 7 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 8 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 9 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*30;
 if month = 10 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 11 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*30;
 if month = 12 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 1 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 2 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*28;
 if month = 3 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 4 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*30;
 if month = 5 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 6 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*30;
 if month = 7 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 8 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 9 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*30;
 if month = 10 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 11 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*30;
 if month = 12 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 keep type600      type665  npdes    outfall  nc_facility    sic_code sic_desc
 year      Month   qtrflow  mon_flow          flowclass      quarter  season   p600_mgl
 flow_mgd  p665_mgl         c2p600   c2p665   q1flow   flow  p600     orig600  p665
 orig665   medsea600        medsea665         medseaflo      flowsub  flosicsea600
flosicsea665
```

```
 flosic600    flosic665 tpc600 tpc665 high_tn high_tp       kgnmgl12             kgpmgl12;
run;

proc sort; by year npdes sic_code; run;

proc univariate noprint;
 var kgnmgl12 kgpmgl12;
 by year npdes sic_code;
 output out=temp1 sum=kgn_12 kgp_12;
run;

data temp1;
 set temp1;
 calc12=1;
run;

proc sort data=temp1; by npdes year; run;




/* calculate loads for permits with 3 or 4 quarters of flow data (mon_flow lt 12)
 Use seasonal values for p600/665. Load calculated for
 a permit/outfall by season and summed for each discharger in each year.
 proc univariate gets a single value for each year/permit/outfall/season for flow
 (which varies across this combination of variables) and the various p600/665
 variables (which are constant across this combination of variables and can be used
 as by variables and carried over into next data set. */

data test.temp2;
 set test.nc_pcs11;
 if mon_flow lt 12 and qtrflow ge 3;
run;

proc sort ; by npdes outfall sic_code year season ; run;

proc univariate noprint;
 var p600 p665 flow;
 by npdes outfall sic_code year season ;
 output out=temp3 median=med600 med665 medseaflo;
run;

data temp4;
 set temp3;
 p600=med600;
 p665=med665;
 drop med600 med665;
run;


data test.temp5;
 set temp4;
 if season=‘winter’ then kgn_34qtr=((p600)*(medseaflo*3785000)*(.000001))*90;
 if season=‘spring’ then kgn_34qtr=((p600)*(medseaflo*3785000)*(.000001))*92;
 if season=‘summer’ then kgn_34qtr=((p600)*(medseaflo*3785000)*(.000001))*92;
 if season=‘fall’ then kgn_34qtr=((p600)*(medseaflo*3785000)*(.000001))*91;
```

```
 if season='winter' then kgp_34qtr=((p665)*(medseaflo*3785000)*(.000001))*90;
 if season='spring' then kgp_34qtr=((p665)*(medseaflo*3785000)*(.000001))*92;
 if season='summer' then kgp_34qtr=((p665)*(medseaflo*3785000)*(.000001))*92;
 if season='fall' then kgp_34qtr=((p665)*(medseaflo*3785000)*(.000001))*91;
run;

proc sort; by year npdes sic_code; run;

proc univariate noprint;
 var kgn_34qtr kgp_34qtr;
 by year npdes sic_code;
 output out=temp6 sum=kgn_34qtr kgp_34qtr;
run;

data temp6;
 set temp6;
 calc34qtr=1;
run;

proc sort data=temp6; by npdes year; run;

/* calculate load for permit/outfall with less than 3 quarters of data */

data test.temp60;
 set test.nc_pcs11;
 if qtrflow lt 3;
 if month = 1 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 2 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*28;
 if month = 3 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 4 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*30;
 if month = 5 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 6 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*30;
 if month = 7 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 8 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 9 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*30;
 if month = 10 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 11 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*30;
 if month = 12 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 1 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 2 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*28;
 if month = 3 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 4 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*30;
 if month = 5 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 6 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*30;
 if month = 7 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 8 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 9 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*30;
 if month = 10 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 11 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*30;
 if month = 12 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 keep type600        type665  npdes     outfall  nc_facility      sic_code sic_desc
 year       Month    qtrflow  mon_flow           flowclass        quarter  season   p600_mgl
 flow_mgd   p665_mgl          c2p600    c2p665   q1flow    flow   p600     orig600  p665
 orig665    medsea600         medsea665          medseaflo        flowsub  flosicsea600
    flosicsea665
 flosic600  flosic665 tpc600 tpc665 high_tn high_tp    kgnmgl12            kgpmgl12;
```

```
run;



proc sort; by year npdes sic_code; run;

proc univariate noprint;
 var kgnmgl12 kgpmgl12;
 by year npdes sic_code;
 output out=temp61 sum=kgnqtrlt3 kgpqtrlt3;
run;

data temp61;
 set temp61;
 calcqtrlt3=1;
run;

proc sort data=temp61; by npdes year; run;

data temp7;
 merge temp1 temp6 temp61;
 by npdes year;
run;

data temp8;
 set temp7;
 if kgnqtrlt3 eq ‘.’ then kgnqtrlt3=0;
 if kgpqtrlt3 eq ‘.’ then kgpqtrlt3=0;
 if kgn_12 eq ‘.’ then kgn_12=0;
 if kgn_34qtr eq ‘.’ then kgn_34qtr=0;
 if kgp_34qtr eq ‘.’ then kgp_34qtr=0;
 if kgp_12 eq ‘.’ then kgp_12=0;
 kg_n=kgn_12+kgn_34qtr+kgnqtrlt3 ;
 kg_p=kgp_34qtr + kgp_12+kgpqtrlt3;
run;

proc print; run;

/* create data set with indiv TN and TP load estimate variables by year */

data load00;
 set temp8;
 if year=2000;
 kgn_00=kg_n;
 kgp_00=kg_p;
 calc1200=calc12;
 calc34qtr00=calc34qtr;
 calcqtrlt300=calcqtrlt3;
 keep npdes sic_code kgn_00 kgp_00 calc1200 calc34qtr00 calcqtrlt300;
run;

proc sort; by npdes sic_code; run;

data load01;
 set temp8;
 if year=2001;
```

```
 kgn_01=kg_n;
 kgp_01=kg_p;
 calc1201=calc12;
 calc34qtr01=calc34qtr;
 calcqtrlt301=calcqtrlt3;
 keep npdes sic_code kgn_01 kgp_01 calc1201 calc34qtr01 calcqtrlt301;
run;

proc sort; by npdes sic_code; run;

data load02;
 set temp8;
 if year=2002;
 kgn_02=kg_n;
 kgp_02=kg_p;
 calc1202=calc12;
 calc34qtr02=calc34qtr;
 calcqtrlt302=calcqtrlt3;
 keep npdes sic_code kgn_02 kgp_02 calc1202 calc34qtr02 calcqtrlt302;
run;

proc sort; by npdes sic_code; run;

data nc_load;
 merge load00 load01 load02;
 by npdes sic_code;
run;

proc sort; by npdes; run;

/* import NC site information and create NC load + site information */

data temp78;
 format npdes $9.;
 set ncsite.nc_site_101105;
 npdes=permit;
run;
proc sort out=tempnc nodupkey ; by npdes ; run;

proc sort data=tempnc; by npdes; run;

data site;
 set site.pcs_site_101405;
 if state=‘NC’;
 keep npdes facility_0805 name_0805 flow_0805 lat_0805 long_0805
 lat_0405 long_0405 sic_desc sic_code;
run;

proc sort; by npdes; run;


data temp99;
 merge tempnc nc_load site;
 by npdes;
run;

```

```
proc contents; run;
data tempload;
 set temp99;
run;
/* flag load values where the percent difference from year to year exceeds
 the 95th pctl or less than the 5 th pctl of all differences for those pairings of years.
 Also flag load values greater than the 95th percentile of all load values */
data test.nc_load_site_110905;
 format lat0405 $10. long0405 $10.;
 set tempload;
 flag=0;
 pctn0001=(kgn_01-kgn_00)/kgn_00;
 if pctn0001 ge 1 or pctn0001 lt-.44 then flag = 1;
 if pctn0001 = ‘ ’ or pctn0001 = ‘.’ then flag =0;
 pctn0102=(kgn_02-kgn_01)/kgn_01;
 if pctn0102 ge 1 or pctn0102 lt -.53 then flag = 1;
 if pctn0102 = ‘ ’ or pctn0102 = ‘.’ then flag =0;
 pctp0001=(kgp_01-kgp_00)/kgp_00;
 if pctp0001 ge 1 or pctp0001 lt -.47 then flag = 1;
 if pctp0001 = ‘ ’ or pctp0001 = ‘.’ then flag =0;
 pctp0102=(kgp_02-kgp_01)/kgp_01;
 if pctp0102 ge 1 or pctp0102 lt -.47 then flag = 1;
 if pctp0102 = ‘ ’ or pctp0102 = ‘.’ then flag =0;
 if kgn_00 ne ‘.’ or kgp_00 ne ‘.’ or kgn_01 ne ‘.’ or kgp_01 ne ‘.’ or
kgn_02 ne ‘.’ or kgp_02 ne ‘.’;
 if ( kgn_00 gt 75400 or kgn_01 gt 73900 or kgn_02 gt 76500 )or
( kgp_00 gt 14300 or kgp_01 gt 13000 or kgp_02 gt 14700 ) then flag=1;
 state=‘NC’;
 lat0405=lat_0405; long0405=long_0405;
 drop lat_0405 long_0405;
run;
proc contents; run;
proc univariate;
 title ‘Summary stats for North Carolina TN and TP annual loads and for inter-annual changes’;
 var kgn_00 kgn_01 kgn_02 kgp_00 kgp_01 kgp_02 pctn0001 pctn0102 pctp0001 pctp0102;
run;
proc contents data=TEST.temp1; run;
proc contents data=TEST.temp60; run;
/* export data to spreadsheet */
PROC EXPORT DATA= TEST.nc_load_site_110905
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\site_1005\nc\nc_loa
d_site_061206t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“load summary by discharger”;
```

```
RUN;


PROC EXPORT DATA= TEST.temp1
 OUTFILE= "Z:\Nutrients\SPARROW\point_source\site_1005\nc\nc_loa
d_site_061206t.xls"
 DBMS=EXCEL REPLACE;
SHEET="12 months";
RUN;

PROC EXPORT DATA= TEST.temp5
 OUTFILE= "Z:\Nutrients\SPARROW\point_source\site_1005\nc\nc_loa
d_site_061206t.xls"
 DBMS=EXCEL REPLACE;
SHEET="gt 3-4 qtrs";
RUN;

PROC EXPORT DATA= TEST.temp60
 OUTFILE= "Z:\Nutrients\SPARROW\point_source\site_1005\nc\nc_loa
d_site_061206t.xls"
 DBMS=EXCEL REPLACE;
SHEET="all the rest";
RUN;

/* export high flow values created in program flowcheck_050506.sas */

PROC EXPORT DATA= summary.flowcheck061306
 OUTFILE= "Z:\Nutrients\SPARROW\point_source\site_1005\nc\nc_loa
d_site_061206t.xls"
 DBMS=EXCEL REPLACE;
SHEET="flowcheck";
RUN;


quit;
```

# Attachment 7: Load Summary Program

```
/* ****************************************************************************

program: summary_072106.sas
date: 1/25/06 2/7/06 2/15/06 2/21/06 3/21/06 4/10/06 6-9-2006
 7/17-25/06
notes: This programs summarizes data developed in the loadprep and loadcalc
programs for individual states. It summarizes the original p600/665 data available
from state and PCS databases. It summarizes various attributes of the calculated loads.

**************************************************************************** */

libname test ‘Z:\Nutrients\SPARROW\point_source\load_1005\summary’;
libname ncn ‘z:\Nutrients\sparrow\point_source\nc\nitrogen’;
libname ncp ‘z:\Nutrients\sparrow\point_source\nc\phosphorus’;
libname sc ‘z:\Nutrients\sparrow\point_source\sc\Flow’;
libname pcs ‘z:\nutrients\sparrow\point_source\PCS_donahoo_0805’;
libname site ‘Z:\Nutrients\SPARROW\point_source\site_1005’;
libname ncsite ‘Z:\Nutrients\SPARROW\point_source\nc\site’;
libname scsite ‘Z:\Nutrients\SPARROW\point_source\sc\site_1005’;
libname ncload ‘Z:\Nutrients\SPARROW\point_source\load_1005\NC’;
libname scload ‘Z:\Nutrients\SPARROW\point_source\load_1005\SC’;
libname flload ‘Z:\Nutrients\SPARROW\point_source\load_1005\FL’;
libname msload ‘Z:\Nutrients\SPARROW\point_source\load_1005\MS’;
libname alload ‘Z:\Nutrients\SPARROW\point_source\load_1005\al’;
libname gaload ‘Z:\Nutrients\SPARROW\point_source\load_1005\ga’;
libname tnload ‘Z:\Nutrients\SPARROW\point_source\load_1005\TN’;
libname vaload ‘Z:\Nutrients\SPARROW\point_source\load_1005\VA_KY’;
libname sparrow ‘D:\sagtsparrow_rf1\data’;

options ps=54 ls=80;

/* **************************************************************************** */

/* summarize p600/665 data for states that have data — original files prepared in
loadprep programs for various states, in section creating single p600/665 variable */
/*
proc sort data=flload.fl600665; by state; run;
proc sort data=ncload.nc600665; by state; run;
proc sort data=scload.sc600665; by state; run;
proc sort data=msload.ms600665; by state; run;
proc sort data=alload.al600665; by state; run;
proc sort data=gaload.ga600665; by state; run;
proc sort data=tnload.tn600665; by state; run;

data test.p600665sum_012506;
 merge flload.fl600665 ncload.nc600665 scload.sc600665 msload.ms600665
 alload.al600665 gaload.ga600665 tnload.tn600665;
 by state;
run;

proc print; run;

PROC EXPORT DATA= test.p600665sum_012506
```

```
    OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\
npdes_sum_071706t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“p600665_stats”;
RUN;

/* **************************************************************************** */

/* summarize types of p600/665 data used and number of flow obs */
/*
data test.summary;
 set alload.nc_pcs11 ncload.nc_pcs11 flload.nc_pcs11 scload.nc_pcs11 msload.nc_pcs11
 gaload.nc_pcs11 tnload.nc_pcs11 vaload.nc_pcs11;;
run;

proc contents; run;

data test.tempflo;
 set test.summary;
 if flow gt 0;
run;
proc sort ; by state ; run;

proc univariate noprint;
 var flow;
 by state ;
 output out=sumflo n=n_flow;
run;


PROC EXPORT DATA= work.sumflo
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“observations with flow”;
RUN;

proc sort data=test.summary ; by state ; run;

proc univariate noprint;
 var orig600 orig665 sea600 sea665 fss_600 fss_665 fs_600 fs_665
 tpc_600 tpc_665 hi600 hi665 flowsub;
 by state ;
 output out=sum sum=sumorig600 sumorig665 sumsea600 sumsea665 sumfss600 sumfss665
 sumfs600 sumfs665 sumtpc600 sumtpc665 sumhi600 sumhi665 sumflosub;
run;


PROC EXPORT DATA= work.sum
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“observations data source”;
RUN;

proc print data=sum; title ‘data sources, by state’;run;
```

```
data test.wwtp;
 set test.summary;
 if sic_code=4952;
run;

proc sort; by flowclass type600 ; run;

proc univariate noprint;
 var p600 ;
 by flowclass type600;
 output out=flowstat median=med600 n=nobs;
run;

proc print data=flowstat; title ‘WWTP(4952) median p600 and number of obs, by flowclass’; run;

PROC EXPORT DATA= WORK.FLOWSTAT
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“median_wwtp_type600”;
RUN;

proc sort data=test.wwtp ; by flowclass type665 ; run;

proc univariate noprint;
 var p665;
 by flowclass type665;
 output out=flowstat median= med665;
run;

proc print data=flowstat; title ‘WWTP(4952) median p665 and number of obs, by flowclass’; run;

PROC EXPORT DATA= WORK.FLOWSTAT
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“median_wwtp_type665”;
RUN;





proc sort data=test.summary ; by state type600 ; run;

proc univariate noprint;
 var p600;
 by state type600;
 output out=type600 median=med600 n=n_600 ;
run;

PROC EXPORT DATA= work.type600
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“type600”;
RUN;

proc print data=type600; title ‘median p600 and number of obs, by state’; run;
```

```
proc sort data=test.summary ; by state type665 ; run;

proc univariate noprint;
 var p665;
 by state type665;
 output out=type665 median=med665 n=n_665 ;
run;

PROC EXPORT DATA= work.type665
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“type665”;
RUN;

proc print data=type665; title ‘median p665 and number of obs, by state’; run;





proc sort data=test.summary ; by state flowclass ; run;

proc univariate noprint;
 var flow;
 by state flowclass;
 output out=flow median=medflow n=n_flow ;
run;

PROC EXPORT DATA= work.flow
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“flow_state_flowclass”;
RUN;

proc print data=flow; title ‘median flow and number of obs, by state and flowclass’; run;




/* merge load data sets from individual states...final data has about 3000 obs */


proc sort data=alload.al_load_011206;; by state; run;
proc sort data=ncload.nc_load_site_110905; by state; run;
proc sort data=scload.sc_load_121905; by state; run;
proc sort data=flload.fl_load_site_112205; by state; run;
proc sort data=msload.ms_load_010906; by state; run;
proc sort data=gaload.ga_load_011206; by state; run;
proc sort data=tnload.tn_load_032006; by state; run;
proc sort data=vaload.va_load_072006; by state; run;
```

```
data test.merge011306;
 merge alload.al_load_011206 ncload.nc_load_site_110905 scload.sc_load_121905
 flload.fl_load_site_112205 msload.ms_load_010906 gaload.ga_load_011206
 tnload.tn_load_032006 vaload.va_load_072006;
 by state;
run;


data test.temp1;
 set test.summary;
 if flow gt 0 ;
run;

proc sort; by descending flow; run;

proc univariate;
 var flow;
run;

proc sort; by year npdes outfall; run;

proc univariate noprint;
 var flow;
 by year npdes outfall;
 output out=temp99 median=med_flow_yr;
run;

data y2000a;
 set temp99;
 if year=2000;
run;

proc sort; by npdes descending med_flow_yr ; run;

proc sort out=temp2000 nodupkey; by npdes; run;

data y2000b;
 set temp2000;
 medflo2000=med_flow_yr;
 major2000=0;
 if medflo2000 gt 0 and medflo2000 le 0.05 then flowclass2000 = ‘1’;
 if medflo2000 gt .05 and medflo2000 le .2 then flowclass2000 = ‘2’;
 if medflo2000 gt .2 and medflo2000 le 1 then flowclass2000 = ‘3’;
 if medflo2000 gt 1 and medflo2000 le 5 then flowclass2000 = ‘4’;
 if medflo2000 gt 5 then flowclass2000 = ‘5’;
 if medflo2000 ge 1 then major2000=1;
 keep npdes medflo2000 flowclass2000 major2000;
run;
proc sort; by npdes; run;

data y2001a;
 set temp99;
 if year=2001;
run;
```

```
proc sort; by npdes descending med_flow_yr ; run;

proc sort out=temp2001 nodupkey; by npdes; run;

data y2001b;
 set temp2001;
 medflo2001=med_flow_yr;
 major2001=0;
 if medflo2001 gt 0 and medflo2001 le 0.05 then flowclass2001 = ‘1’;
 if medflo2001 gt .05 and medflo2001 le .2 then flowclass2001 = ‘2’;
 if medflo2001 gt .2 and medflo2001 le 1 then flowclass2001 = ‘3’;
 if medflo2001 gt 1 and medflo2001 le 5 then flowclass2001 = ‘4’;
 if medflo2001 gt 5 then flowclass2001 = ‘5’;
 if medflo2001 ge 1 then major2001=1;
 keep npdes medflo2001 flowclass2001 major2001;
run;

proc sort; by npdes; run;

data y2002a;
 set temp99;
 if year=2002;
run;

proc sort; by npdes descending med_flow_yr ; run;

proc sort out=temp2002 nodupkey; by npdes; run;

data y2002b;
 set temp2002;
 medflo2002=med_flow_yr;
 major2002=0;
 if medflo2002 gt 0 and medflo2002 le 0.05 then flowclass2002 = ‘1’;
 if medflo2002 gt .05 and medflo2002 le .2 then flowclass2002 = ‘2’;
 if medflo2002 gt .2 and medflo2002 le 1 then flowclass2002 = ‘3’;
 if medflo2002 gt 1 and medflo2002 le 5 then flowclass2002 = ‘4’;
 if medflo2002 gt 5 then flowclass2002 = ‘5’;
 if medflo2002 ge 1 then major2002=1;
 keep npdes medflo2002 flowclass2002 major2002;
run;

proc sort ; by npdes ; run;

proc sort data=test.merge011306; by npdes ; run;


data test.merge041006;
 merge test.merge011306 y2000b y2001b y2002b;
 by npdes ;
run;

proc contents; run;
proc contents data=vaload.va_load_072006; run;
```

```
data loadcov;
 set test.merge041006;
 keep npdes lat_0405 long_0405 lat0405 long0405 lat_0805 long_0805 lat_nc long_nc lat_ms long_ms
    lat_sc long_sc
 lat_tn long_tn dd_lat dd_long ;
run;

PROC EXPORT DATA= work.loadcov
 OUTFILE=
    "Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_latlong_071706t.xls"
 DBMS=EXCEL REPLACE;
SHEET="npdes_lat_long_temp";
RUN;



proc sort data=test.merge041006; by state ; run;

proc univariate data=test.merge041006 noprint;
 var calc1200 calc1201 calc1202 calc34qtr00 calc34qtr01 calc34qtr02
 calcqtrlt300 calcqtrlt301 calcqtrlt302;
 by state ;
 output out=amtdata sum= scalc1200 scalc1201 scalc1202 scalc34qtr00 scalc34qtr01 scalc34qtr02
 scalcqtrlt300 scalcqtrlt301 scalcqtrlt302;
run;

PROC EXPORT DATA= work.amtdata
 OUTFILE= "Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls"
 DBMS=EXCEL REPLACE;
SHEET="load_calc_type_st_yr";
RUN;

proc print data=amtdata; title 'summary of ways that loads were calculated'; run;


data temp4952;
 set test.merge041006;
 if sic_code = '4952';
run;

proc sort ; by sic_code sic_desc flowclass2002 state; run;

proc univariate noprint;
 var kgn_02 kgp_02 ;
 by sic_code sic_desc flowclass2002 state;
 output out=kgsum_sicstfl sum=sicstfl_tn02 sicstfl_tp02;
run;

proc print data=kgsum_sicstfl; title 'TN and TP loads for SIC=4952, 2002, summed by state and
    flowclass'; run;

PROC EXPORT DATA= work.kgsum_sicstfl
    OUTFILE= "Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls"
 DBMS=EXCEL REPLACE;
SHEET="kgsum2002_sicstfl";
RUN;
```

```
proc sort data=test.merge041006; by sic_code sic_desc; run;

proc univariate noprint;
 var kgn_00 kgn_01 kgn_02 kgp_00 kgp_01 kgp_02;
 by sic_code sic_desc;
 output out=kg_sum_sic sum=sic_tn00 sic_tn01 sic_tn02 sic_tp00 sic_tp01 sic_tp02
 n=nobs_n00 nobs_n01 nobs_n02 nobs_p00 nobs_p01 nobs_p02;
run;

proc print data=kg_sum_sic; title 'TN and TP loads, summed by SIC-code'; run;

PROC EXPORT DATA= work.kg_sum_sic
 OUTFILE= "Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls"
 DBMS=EXCEL REPLACE;
SHEET="kg_sum_sic";
RUN;



proc sort data=test.merge041006; by state; run;

proc univariate noprint;
 var kgn_02 kgp_02 ;
 by state;
 output out=n_discharger n=n_discharger_tn02 n_discharger_tp02;
run;

proc print data=n_discharger; title 'TN and TP dischargers,2002, summed by state'; run;

PROC EXPORT DATA= work.n_discharger
 OUTFILE= "Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls"
 DBMS=EXCEL REPLACE;
SHEET="2002_dischargers";
RUN;




proc sort data=test.merge041006; by state flowclass2000; run;

proc univariate noprint;
 var kgn_00 kgp_00;
 by state flowclass2000;
 output out=kgnp00_flow sum=kgn00 kgp00 n=nob_n nobs_p;
run;

PROC EXPORT DATA= work.kgnp00_flow
 OUTFILE= "Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls"
 DBMS=EXCEL REPLACE;
SHEET="kgnp00_state_flow";
RUN;
```

```
proc sort data=test.merge041006; by state flowclass2001; run;

proc univariate noprint;
 var kgn_01 kgp_01;
 by state flowclass2001;
 output out=kgnp01_flow sum=kgn01 kgp01 n=nob_n nobs_p;
run;

PROC EXPORT DATA= work.kgnp01_flow
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“kgnp01_state_flow”;
RUN;



proc sort data=test.merge041006; by state flowclass2002; run;

proc univariate noprint;
 var kgn_02 kgp_02;
 by state flowclass2002;
 output out=kgnp02_flow sum=kgn02 kgp02 n=nob_n nobs_p;
run;

PROC EXPORT DATA= work.kgnp02_flow
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“kgnp02_st_flow”;
RUN;


/* import sic code information and merge into summary table */


PROC IMPORT OUT= sic_code
 DATAFILE= “Z:\GIS\PointSources\PCS_site_0205\sic_codes.xls”
 DBMS=EXCEL2000 REPLACE;
GETNAMES=YES;
RUN;

data sic_code;
 format sic_code $4. sic_desc $70.;
 set sic_code;
 sic_code=sic;
 keep sic_code sic_desc epa_noaa;
run;

PROC EXPORT DATA= work.sic_code
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“sic_code”;
RUN;

/* compare loads calculated for dischargers with 12 months of 2002 data with original p600 and
p665 data with loads for same dischargers using TPC (sic only) concentrations */
```

```
data test.origdata;
 set test.summary;
 if year=2002 and mon_flow=12 and (orig600=1 or orig665=1);
 if not(state='FL');
run;

data orig;
 set test.origdata;
 if month = 1 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 2 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*28;
 if month = 3 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 4 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*30;
 if month = 5 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 6 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*30;
 if month = 7 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 8 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 9 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*30;
 if month = 10 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 11 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*30;
 if month = 12 then kgnmgl12=((p600)*(flow*3785000)*(.000001))*31;
 if month = 1 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 2 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*28;
 if month = 3 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 4 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*30;
 if month = 5 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 6 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*30;
 if month = 7 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 8 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 9 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*30;
 if month = 10 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
 if month = 11 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*30;
 if month = 12 then kgpmgl12=((p665)*(flow*3785000)*(.000001))*31;
run;

proc sort; by year npdes sic_code flowclass; run;

proc univariate noprint;
 var kgnmgl12 kgpmgl12;
 by year npdes sic_code flowclass;
 output out=orig2 sum=kgn_orig kgp_orig;
run;

proc sort data=orig2; by year npdes sic_code flowclass; run;

data tpc;
 set test.origdata;
 if month = 1 then kgnmgl12=((tpc600)*(flow*3785000)*(.000001))*31;
 if month = 2 then kgnmgl12=((tpc600)*(flow*3785000)*(.000001))*28;
 if month = 3 then kgnmgl12=((tpc600)*(flow*3785000)*(.000001))*31;
 if month = 4 then kgnmgl12=((tpc600)*(flow*3785000)*(.000001))*30;
 if month = 5 then kgnmgl12=((tpc600)*(flow*3785000)*(.000001))*31;
 if month = 6 then kgnmgl12=((tpc600)*(flow*3785000)*(.000001))*30;
 if month = 7 then kgnmgl12=((tpc600)*(flow*3785000)*(.000001))*31;
 if month = 8 then kgnmgl12=((tpc600)*(flow*3785000)*(.000001))*31;
 if month = 9 then kgnmgl12=((tpc600)*(flow*3785000)*(.000001))*30;
 if month = 10 then kgnmgl12=((tpc600)*(flow*3785000)*(.000001))*31;
```

```
 if month = 11 then kgnmgl12=((tpc600)*(flow*3785000)*(.000001))*30;
 if month = 12 then kgnmgl12=((tpc600)*(flow*3785000)*(.000001))*31;
 if month = 1 then kgpmgl12=((tpc665)*(flow*3785000)*(.000001))*31;
 if month = 2 then kgpmgl12=((tpc665)*(flow*3785000)*(.000001))*28;
 if month = 3 then kgpmgl12=((tpc665)*(flow*3785000)*(.000001))*31;
 if month = 4 then kgpmgl12=((tpc665)*(flow*3785000)*(.000001))*30;
 if month = 5 then kgpmgl12=((tpc665)*(flow*3785000)*(.000001))*31;
 if month = 6 then kgpmgl12=((tpc665)*(flow*3785000)*(.000001))*30;
 if month = 7 then kgpmgl12=((tpc665)*(flow*3785000)*(.000001))*31;
 if month = 8 then kgpmgl12=((tpc665)*(flow*3785000)*(.000001))*31;
 if month = 9 then kgpmgl12=((tpc665)*(flow*3785000)*(.000001))*30;
 if month = 10 then kgpmgl12=((tpc665)*(flow*3785000)*(.000001))*31;
 if month = 11 then kgpmgl12=((tpc665)*(flow*3785000)*(.000001))*30;
 if month = 12 then kgpmgl12=((tpc665)*(flow*3785000)*(.000001))*31;
run;

proc sort; by year npdes sic_code flowclass; run;

proc univariate noprint;
 var kgnmgl12 kgpmgl12;
 by year npdes sic_code flowclass;
 output out=tpc2 sum=kgn_tpc kgp_tpc;
run;

proc sort data=tpc2; by year npdes sic_code flowclass; run;

data test.compare;
 merge orig2 tpc2;
 by year npdes sic_code flowclass;
run;

data test.compare;
 set test.compare;
 kgn_dif=((kgn_orig-kgn_tpc)/kgn_orig)*100;
 kgp_dif=((kgp_orig-kgp_tpc)/kgp_orig)*100;
 kgn_dif=kgn_orig-kgn_tpc;
 kgp_dif=kgp_orig-kgp_tpc;
run;

proc univariate;
 title1 'Wilcoxen signed rank test to see if there is a difference in loads';
 title2 'calculated with orig vs TPC';
 var kgn_dif kgp_dif;
run;



/* ************************************************************************************ */
/* ************************************************************************************** */

/* input data about reach of dischargers and the stream meanq and watershed area
and summarize total 2002 TN and TP by by the type of receiving stream, in terms
of the mean annual flow of the reach receiving the discharge and the drainage area,
and also by major and minor dischargers. */

/* ************************************************************************************ */
```

```
/* ************************************************************************** */

PROC IMPORT OUT= WORK.pts_catch
 DATAFILE= "Z:\GIS\PointSources\sparrow\pts_catch.xls"
 DBMS=EXCEL REPLACE;
SHEET="pts_catch$";
GETNAMES=YES;
MIXED=NO;
SCANTEXT=YES;
USEDATE=YES;
SCANTIME=YES;
RUN;

data pts_catch;
 format npdes $9. wshed $5.;
 set pts_catch;
 state=substr(npdes,1,2);
 npdes=npdes_arc;
 wshed=e2rf1__;
 drop npdes_arc e2rf1__;
run;

proc sort; by wshed; run;

data temp99;
 set sparrow.sparrow_data1;
 keep wshed meanq demtarea pskgn_02 pskgp_02;
run;

proc sort; by wshed; run;

data temp88;
 merge temp99 pts_catch;;
 by wshed;
run;

data temp88;
 format areaclass $7.;
 set temp88;
 if npdes ne ' ';
 if demtarea le 100 then areaclass = '1';
 if demtarea gt 100 and areaclass le 1000 then areaclass = '2';
 if demtarea gt 1000 and areaclass le 10000 then areaclass = '3';
 if demtarea gt 10000 then areaclass = '4';
 if wshed gt 80000 then areaclass = 'coastal';
run;

proc sort; by npdes; run;

proc sort data=test.merge041006; by npdes; run;

data test.merge072406;
 merge temp88 test.merge041006;
 by npdes;
run;
```

```
proc sort; by state areaclass; run;

proc univariate noprint;
 var kgn_02 kgp_02;
 by state areaclass;
 output out=temp999 sum=sumkgn sumkgp;
run;

PROC EXPORT DATA= work.temp999
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“rch_area_tn_tp_load”;
RUN;

data major1;
 set test.merge072406;
 if major2002=1;
run;

proc sort ; by wshed ; run;

proc univariate noprint;
 var kgn_02 kgp_02;
 by wshed;
 output out=major2 sum=psn02_major psp02_major;
run;

proc sort data=major2; by wshed; run;

data minor1;
 set test.merge072406;
 if major2002=0;
run;

proc sort ; by wshed ; run;

proc univariate noprint;
 var kgn_02 kgp_02;
 by wshed;
 output out=minor2 sum=psn02_minor psp02_minor;
run;

proc sort data=minor2; by wshed; run;

data test.point02;
 merge major2 minor2 temp99;
 by wshed;
run;

data test.point02;
 set test.point02;
 if psn02_major=‘.’ then psn02_major=0;
 if psp02_major=‘.’ then psp02_major=0;
 if psn02_minor=‘.’ then psn02_minor=0;
 if psp02_minor=‘.’ then psp02_minor=0;
```

```
 psn02_total=psn02_major+psn02_minor;
 psp02_total=psp02_major+psp02_minor;
run;

data test.point02;
 set test.point02;
 if not(psn02_total=0 and psp02_total=0);
run;

proc print ;
 title ‘comparison of ps_02, before and after recnet round of program edits’;
 var wshed psn02_total pskgn_02 psp02_total pskgp_02;
run;


PROC EXPORT DATA= test.point02
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“point_source_02”;
RUN;




PROC EXPORT DATA= TEST.merge072406
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summary\npdes_sum_071706t.xls”
 DBMS=EXCEL REPLACE;
SHEET=“ncscflmsal”;
RUN;


quit;
```

# Attachment 8: Flow Magnitude Checking Program

```
/* ******************************************************************************

program: flowcheck_050506.sas
date: 05-05-06 6-12-06 6-22-06

****************************************************************************** */

libname test 'Z:\Nutrients\SPARROW\point_source\load_1005\summary';

libname ncdata 'Z:\Nutrients\SPARROW\point_source\load_1005\NC';
libname scdata 'Z:\Nutrients\SPARROW\point_source\load_1005\SC';
libname fldata 'Z:\Nutrients\SPARROW\point_source\load_1005\FL';
libname msdata 'Z:\Nutrients\SPARROW\point_source\load_1005\MS';
libname aldata 'Z:\Nutrients\SPARROW\point_source\load_1005\al';
libname gadata 'Z:\Nutrients\SPARROW\point_source\load_1005\ga';
libname tndata 'Z:\Nutrients\SPARROW\point_source\load_1005\TN';
libname vadata 'Z:\Nutrients\SPARROW\point_source\load_1005\VA_KY';
libname sparrow 'D:\sagtsparrow_rf1\data';

options ps=54 ls132;

/* ****************************************************************************** */

data sctemp;
 set scdata.nc_pcs7a;
 keep npdes outfall year month sic_code p600 p665 flow name;
run;

data nctemp;
 format name $30.;
 set ncdata.nc_pcs7;
 name=facility;
 keep npdes outfall year month sic_code p600 p665 flow name;
run;

data vatemp;
 set vadata.nc_pcs7;
 keep npdes outfall year month sic_code p600 p665 flow name;
run;

data tntemp;
 set tndata.nc_pcs7;
 keep npdes outfall year month sic_code p600 p665 flow name;
run;

data mstemp;
 set msdata.nc_pcs7;
 keep npdes outfall year month sic_code p600 p665 flow name;
run;

data gatemp;
 set gadata.nc_pcs7;
```

```
 keep npdes outfall year month sic_code p600 p665 flow name;
run;

data altemp;
 set aldata.nc_pcs7;
 keep npdes outfall year month sic_code p600 p665 flow name;
run;

data fltemp;
 set fldata.nc_pcs7;
 keep npdes outfall year month sic_code p600 p665 flow name;
run;

data test.temp1;
 set sctemp nctemp vatemp gatemp mstemp altemp tntemp fltemp;
run;

proc sort data=test.temp1; by npdes outfall; run;

data test.temp1;
 set test.temp1;
 if flow gt 0;
run;

proc univariate noprint;
 var flow;
 by npdes outfall;
 output out=tempflow median=median_flow;
run;

proc sort data=tempflow; by npdes outfall; run;

data test.mrb2_flow_050506;
 merge test.temp1 tempflow;
 by npdes outfall;
run;

proc univariate; var flow; run;

data test.mrb2_flow_050506;
 format highflow $10.;
 set test.mrb2_flow_050506;
 if flow gt 100 then highflow=‘flo>100MGD’;
 if flow gt (10*median_flow) then highflow=‘10*medflo’;
 if flow gt (100*median_flow) then highflow=‘100*medflo’;
run;

/* id flow values ge 1 (major dischargers, the source of most load) where the flow value
is either very high (ge 100 MGD; there was a significant break point in the distribution of flow values
beyond 100 MGD) or is much higher than the median flow value of that discharger */


data test.flowcheck061306;
 set test.mrb2_flow_050506;;
 if flow ge 1 and (highflow=‘10*medflo’ or highflow=‘100*medflo’ or highflow=‘flo>100MGD’);
run;
```

```
proc sort; by npdes; run;

proc print;
 title ‘records with flow ge 1 and flow gt 100 MGD or with flow gt 10/100* median flow’;
 var name npdes outfall year month sic_code flow median_flow highflow;
run;

/* export high flow values created in program flowcheck_050506.sas */

PROC EXPORT DATA= TEST.FLOWCHECK061306
 OUTFILE= “Z:\Nutrients\SPARROW\point_source\load_1005\summar
y\flowcheck_062206.xls”
 DBMS=EXCEL REPLACE;
SHEET=“flowcheck_all”;
RUN;



Quit;
```

McMahon, Tervelt, and Donehoo—**Methods for Estimating Annual Wastewater Nutrient Loads in the Southeastern United States**—Open-File Report 2007–1040