National Water Quality Assessment Program

Annual Wastewater Nutrient Data Preparation and Load Estimation Using the Point-Source Load Estimation Tool (PSLoadEsT)

Open-File Report 2019–1025

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

Annual Wastewater Nutrient Data Preparation and Load Estimation Using the Point-Source Load Estimation Tool (PSLoadEsT)

By Lillian E. Gorman Sanisaca, Kenneth D. Skinner, and Molly A. Maupin

National Water Quality Assessment Program

Open-File Report 2019–1025

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

U.S. Department of the Interior
David Bernhardt, Acting Secretary

U.S. Geological Survey
James F. Reilly II, Director

U.S. Geological Survey, Reston, Virginia: 2019

Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Suggested citation:
Gorman Sanisaca, L.E., Skinner, K.D., and Maupin, M.A., 2019, Annual wastewater nutrient data preparation and load estimation using the Point Source Load Estimation Tool (PSLoadEsT): U.S. Geological Survey Open-File Report 2019-1025, 48 p., https://doi.org/10.3133/ofr20191025.

ISSN 2331-1258 (online)

Contents

Figures

Tables

This page left intentionally blank

Annual Wastewater Nutrient Data Preparation and Load Estimation Using the Point-Source Load Estimation Tool (PSLoadEsT)

By Lillian E. Gorman Sanisaca, Kenneth D. Skinner, and Molly A. Maupin

Abstract

The Point-Source Load Estimation Tool (PSLoadEsT) provides a user-friendly interface for generating reproducible load calculations for point source dischargers while managing common data challenges including duplicates, incompatible input tables, and incomplete or missing nutrient concentration or effluent flow data. Maintaining a consistent method across an entire study area is important when estimating loads to be used as calibration data for regional water-quality models. PSLoadEsT is written using the open-source programming language R and has an easy-to-use interface written in Visual Basic for Applications® within a Microsoft Access® database file that guides the user through the necessary steps to estimate point source loads. The purpose of this report is to provide a detailed user guide for PSLoadEsT.

1. Introduction

When developing regional-scale nutrient water-quality models such as SPARROW (SPAtially Reference Regression On Watershed attributes) (Schwarz and others, 2006) using annual wastewater nutrient concentration and effluent flow data for calibration, a number of challenges must be overcome (McMahon and others, 2007). Reporting requirements for nutrient concentration and effluent flow data are permit specific and therefore may not be consistent across an entire region, and missing and/or duplicated data are common, as well as inaccurate or erroneous data. Therefore, it is necessary to have a consistent method for dealing with these data challenges across the entire regional area. This user guide for the Point Source Load Estimation Tool (PSLoadEsT) documents how the tool provides a reproducible and easy-to-use method for highlighting and managing potential inaccuracies, duplications, and/or incomplete and missing data.

1.1. Point-Source Load Estimation Tool (PSLoadEsT)

The Point-Source Load Estimation Tool (PSLoadEsT) was developed as a user-friendly method for estimating annual wastewater nutrient loads while managing the challenges of missing, duplicated, or erroneous nutrient concentration and effluent flow data for the purposes of model calibration and application. PSLoadEsT is written in the R programing language (R Core Team, 2016) with a push-button Microsoft Access® interface scripted in Visual Basic for Applications®. The tool is primarily based on the methods outlined in “Methods for Estimating Annual Wastewater Nutrient Loads in the Southeastern United States” (McMahon and others, 2007), and program steps and object names follow these methods closely. The methods applied by McMahon and others (2007) were in the form of a series of SAS scripts designed to be run in a specific order to estimate point source nutrient loads specifically in the Southeastern United States, however, with some modifications the methods could be applied to other areas. PSLoadEsT was developed as a generalized application of the methods used by McMahon and others (2007) in any region with the same type of data used in the original methods. However, additional substitution methods as well as a user-friendly interface have been added.

2. Loading PSLoadEsT

2.1. Minimum System Requirements

1. Microsoft Windows® 7 or above
2. Microsoft Office® 2007–2016 with Microsoft Access® (32-bit)
3. Microsoft Access® Driver (*.mdb, *.accdb) (32-bit)
4. 64-bit Operating System

*The program has not been tested using Parallels Desktop® for Mac®

2.1.1. 32-Bit Microsoft Access and 64-Bit R

PSLoadEsT uses a Microsoft Access interface to guide the user through the program steps that are executed in R. The input data are also stored in a Microsoft Access database. A 64-bit version of R-3.3.0 (includes a 32-bit version) with all necessary libraries installed is included as part of PSLoadEsT and is the only supported version of R for the program at this time. The use of 64-bit R allows for processing of large datasets that are common in point-source nutrient concentration and effluent flow data, while the use of 32-bit R is required to connect to 32-bit Microsoft Office® files. Because the 32-bit version of Microsoft Office® is the most common installation of the program even on computers with 64-bit processors, PSLoadEsT includes a built-in method to manage the incompatibility between the part of the program that completes actions in 64-bit R and the 32-bit Microsoft Access interface and input data file. The 32-bit version of R-3.3.0 that is installed by default with the 64-bit version is used to import and export objects from Microsoft Access databases into the 64-bit R environment.

It is also necessary to verify that the required 32-bit Microsoft Access Driver that allows R to connect to the PSLoadEsT interface is present as occasionally drivers may be missing from the installation. There are two methods for verifying presence of the Microsoft Access Driver (*.mdb, *.accdb)—(1) navigating Windows Explorer and (2) R command.

To verify the driver is present through Windows Explorer, navigate to C:\Windows\SysWOW64\odbcad32.exe and open the executable. Click on the “Drivers” tab and search the list for the required “Microsoft Access Driver (*.mdb, *.accdb).” It is crucial that the driver be listed on the “Drivers” tab; the driver being present on the “User DSN” tab, which is the default tab on which the executable opens, is NOT sufficient for the execution of PSLoadEsT. If the driver is not present on the “Drivers” tab, it must be installed.

To verify the driver is present using R, open the Rgui.exe included in ~PSLoadEsT\R-3.3.0\bin\i386\. Type the following commands in the console window:

```
> install.packages("odbc")
> odbc::odbcListDrivers()
```

Look in the “name” column for the required “Microsoft Access Driver (*.mdb, *.accdb).” driver and if it is not present the user must install the missing driver. If prompted to select a CRAN mirror, select from the list of CRAN mirrors in the pop-up window. The user may also be asked to install package dependencies that may need compilation; if this message appears, select “y” to install the package dependencies.

2.2. PSLoadEsT Repository

PSLoadEsT is available on USGS Bitbucket as a publicly accessible code repository here https://my.usgs.gov/bitbucket/projects/PSLC/repos/psloadest/browse. Any code updates and/or future versions will be available for download or cloning of the repository with appropriate documentation of all changes. Cloning of the repository allows the user to view a complete progression of all code changes, while downloading the current version only gives access to the most recent version of the tool.

All files and folders should be downloaded or cloned into a directory that does not contain any spaces or special characters. While the user is free to place the repository in their preferred location, renaming and/or moving files or folders within the program will cause program failure. Directory structure and file details are given in appendix 1.

2.3. Enable Content

When starting PSLoadEsT for the first time in a given location on the user’s computer, the user must enable content in Microsoft Access files. This turns on the Visual Basic for Applications (VBA) code that allows the interface to guide the user through the program and execute R scripts from the PSLoadEsT.accdb interface. This is a one-time process assuming that the program is not moved/copied; however, the user will need to enable content in each new program location.

All files in PSLoadEsT with VBA code/macros must have the VBA content enabled so that they are listed as “trusted documents” on the user’s system allowing the code to run. The enable content message provides the user with protection so that unauthorized code that could make changes to the user’s system does not run automatically upon opening a document. Code in the PSLoadEsT interface is safe and will not make changes to the user’s system. The exampleDataset.accdb file does not have VBA code/macros, but to import data from this file the file must be listed as a “trusted document.”

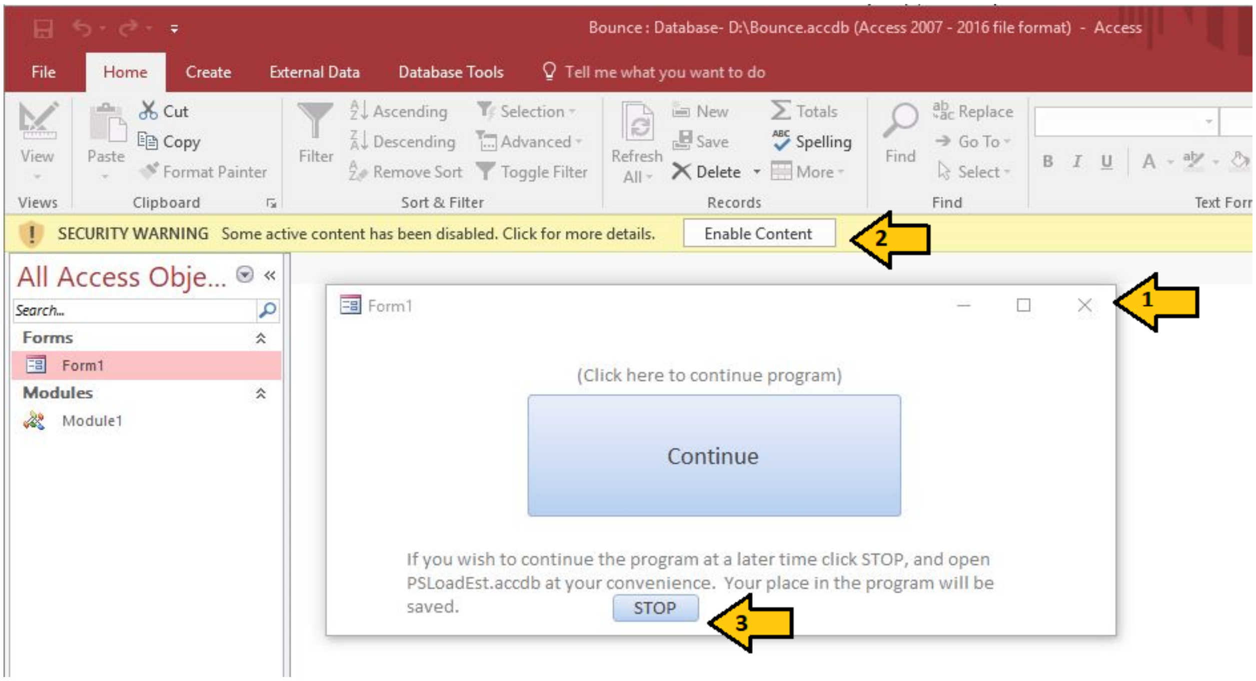
Failing to complete this step in the order detailed in section 2.3.1 will result in program failure. The following files contain VBA code:

1. PSLoadEsT\Rscripts\Bounce.accdb
2. PSLoadEsT\PSLoadEsT.accdb
3. PSLoadEsT\exampleDataset\exampleDataset.accdb (only if running PSLoadEsT with the included example dataset)

The PSLoadEsT\Rscripts\sizeTest.accdb and PSLoadEsT\Rscripts\Blank.accdb files do not have to be enabled for PSLoadEsT to run.

2.3.1. Steps to Enable VBA Content in PSLoadEsT

1. Enable content in the PSLoadEsT\Rscripts\Bounce.accdb file (fig. 1).
    a. Open the Bounce.accdb file. This file acts as a springboard to send the user back to the core file PSLoadEsT\PSLoadEsT.accdb and contains no information.
    b. Close the Pop-up window by clicking the “X” in the top right corner.
    c. Click “Enable Content” in the yellow ribbon on the top of the window.
    d. Click “Stop” on the bottom of the pop-up (closing the Bounce.accdb file).



Bounce : Database- D:\Bounce.accdb (Access 2007 - 2016 file format) - Access

SECURITY WARNING Some active content has been disabled. Click for more details.

(Click here to continue program)

If you wish to continue the program at a later time click STOP, and open PSLoadEst.accdb at your convenience. Your place in the program will be saved.

Figure 1. Screenshot illustrating steps to enable content in the PSLoadEsT\Rscripts\Bounce.accdb file.

1. Enable Content in the PSLoadEsT\PSLoadEsT.accdb file.
    a. Open the PSLoadEsT.accdb file.
    b. Close the Pop-up window by clicking the “X” in the top right corner (Pop-up only appears on first execution or when program is moved).
    c. Click “Enable Content” in the yellow ribbon on the top of the window.
2. Enable Content in the PSLoadEsT\exampleDataset\exampleDataset.accdb file (if executing PSLoadEsT with included example data).
    a. Open the exampleDataset.accdb file.
    b. Click “Enable Content” in the yellow ribbon on the top of the window.
    c. All content is now enabled, and the PSLoadEsT is ready to run.

2.4. Screen Resolution

PSLoadEsT was designed with the screen resolution set at 100 percent. If the user’s computer settings have the screen resolution set at a different level, the screenshots shown in this manual will look different and blank spaces in the screens may appear, however, all data output will still be visible. Occasionally, a user’s settings may cause the interface windows to be cut off, in that case simply resize the window.

3. Input Data Preparation and Formatting for PSLoadEsT

Monitoring data for import into PSLoadEsT should be pulled from the U.S. Environmental Protection Agency (EPA) Permit Compliance System (PCS) database (U.S. Environmental Protection Agency, 1990) using the Loading Tool (https://echo.epa.gov/trends/loading-tool). The only supported import data format in PSLoadEsT is a Microsoft Access .accdb file containing seven core tables and one optional table. The names of the input tables and fields may vary within the import file; however, the type of data detailed in table 1 must be present. Table and field names must NOT contain spaces or special characters, as this will cause program failure.

Table 1. Required types of input tables and fields.

[Original input table and field names may vary, but the type of table/field listed MUST be present in the import database. Additional fields may be present in the input tables, and although they will not be used by the program, they will be included in the output tables. Original data types (i.e. text, numeric, integer, etc.) are not used by PSLoadEsT; however, it is recommended that all fields be designated as “text” fields]

Table	Field name	Description
DMR	npdes	National Pollutant Discharge Elimination System identifier
DMR	outfall	identifier designating discharge pipe
DMR	parameter	parameter code indicating nutrient to be analyzed
DMR	mon_loc	monitoring location (PSLoadEsT only considers records with mon_loc = “1”, but others may be present in input data)
DMR	date	measurement date (mm/dd/yyyy)
DMR	c1	Minimum measured monthly concentration
DMR	c2	Average monthly concentration
DMR	c3	Maximum measured monthly concentration
DMR	q1	Average measured monthly discharge
DMR	q2	Maximum measured monthly discharge
FLOW	npdes	National Pollutant Discharge Elimination System identifier
FLOW	outfall	identifier designating discharge pipe
FLOW	parameter	parameter code indicating nutrient to be analyzed
FLOW	mon_loc	monitoring location (PSLoadEsT only considers records with mon_loc = “1”, but others may be present in input data)
FLOW	date	measurement date (mm/dd/yyyy)
FLOW	c1	Minimum measured monthly concentration
FLOW	c2	Average monthly concentration
FLOW	c3	Daily maximum measured concentration
FLOW	q1	Average measured monthly discharge
FLOW	q2	Maximum measured monthly discharge

Table	Field name	Description
LIMITS[1]	npdes	National Pollutant Discharge Elimination System identifier
LIMITS[1]	outfall	identifier designating discharge pipe
LIMITS[1]	parameter	parameter code indicating nutrient to be analyzed
LIMITS[1]	mon_loc	monitoring location (PSLoadEsT only considers records with mon_loc = “1”, but others may be present in input data)
LIMITS[1]	start_date	Limit begin date
LIMITS[1]	c1	limit values for the measured c1 values from the DMR table
LIMITS[1]	c1_stat	statistical base type codes relating to the c1 values from the LIMITS table
LIMITS[1]	c2	limit values for the measured c2 values from the DMR table
LIMITS[1]	c2_stat	statistical base type codes relating to the c2 values from the LIMITS table
LIMITS[1]	c3	limit values for the measured c3 values from the DMR table
LIMITS[1]	c3_stat	statistical base type codes relating to the c3 values from the LIMITS table
LIMITS[1]	q1	limit values for the measured q1 values from the FLOW table
LIMITS[1]	q1_stat	statistical base type codes relating to the q1 values from the LIMITS table
LIMITS[1]	q2	limit values for the measured q2 values from the FLOW table
LIMITS[1]	q2_stat	statistical base type codes relating to the q2 values from the LIMITS table
FACILITIES	npdes	National Pollutant Discharge Elimination System identifier
FACILITIES	sic_code	Standard Industrial Classification code
FACILITIES	treatment_level	Level of treatment at wastewater treatment facilities
FACILITIES	state	2 letter State designation
sic_codes	sic_code	Standard Industrial Classification code
State_Expansion	state	2 letter State designation
State_Expansion	Expansion_Group	2 letter State designations for all States within expansion group for the purposes of stepped substitutions of missing data
State_Expansion	Expansion_GroupName	name of expansion group for the purposes of stepped substitutions of missing data
National_Medians	sic_code	Standard Industrial Classification code
National_Medians	treatment_level	Level of treatment at wastewater treatment facilities
National_Medians	parameter	parameter code indicating nutrient to be analyzed
National_Medians	year	measurement year
National_Medians	concentration	nutrient concentration annual median
Rubin_TPC	sic_code	Standard Industrial Classification code
Rubin_TPC	parameter	parameter code indicating nutrient to be analyzed
Rubin_TPC	result	nutrient concentration

[1] Optional table.

3.1. DMR Table

The Discharge Monitoring Report (DMR) table contains the facility-specific nutrient concentration data used to generate substitutions and in the final load calculation. Reporting requirements may vary by facility and therefore nutrient concentration data may be incomplete across the study region (McMahon and others, 2007). Duplicates resulting from the transfer of data from individual facilities to a State database and then a subsequent transfer to the regional EPA PCS databases may exist. Reporting errors (incorrect units, typographical errors, etc.) may also be present as in any raw dataset. PSLoadEsT has built-in methods for managing all these potential data problems and guides the user through the elimination and/or management of these problems prior to the final load calculation. Records in the DMR table are uniquely identified by National Pollutant Discharge Elimination System (NPDES) identifier, outfall (discharge pipe), parameter code identifying the nutrient measured (typically 5-digit), and measurement date. Multiple nutrient-concentration parameter codes can be run simultaneously through the program. The nutrient concentration field used for substitutions and the load calculation is the “c2” field for nutrient concentration (table 1.)

3.2. FLOW Table

The FLOW table contains effluent flow data uniquely identified by NPDES, outfall, 5-digit parameter code for effluent flow, and measurement date. Only one flow parameter code at a time can be run through the program for a load calculation. Many of the same types of data challenges that may exist in the DMR table may also be present in the FLOW table and are managed accordingly. Unlike the nutrient concentration data, however, missing effluent flow data are not replaced with any type of median substitution. The primary field used in the load calculation is the “q1” field for effluent flow given in million gallons per day (MGD) units, however, “q2” field values may be used if “q1” is missing. Facilities missing both the “q1” and “q2” fields (table 1) for effluent flow data will not have loads calculated.

3.3. LIMITS Table

The LIMITS table contains information related to the monitoring and quantitative limits for a parameter such as monitoring location, start and end dates, sample type and frequency, units, limit values, and statistical base codes (code representing the unit of measure applicable to the limit value). In previous studies (Maupin and Ivahnenko, 2011), the LIMITS table was used to perform routine quality checks on the data by identifying possible outliers (values outside the range of the permit limits for a given parameter) and by checking parameter units. PSLoadEsT only runs simple duplicate evaluation and formatting on this table. The LIMITS table is an optional input table. Records are uniquely identified by NPDES, outfall, 5-digit parameter code for nutrient concentration, and measurement start date.

3.4. FACILITIES Table

The FACILITIES table contains information about the facilities such as location, name, type of facility, etc. Records are uniquely identified by NPDES, Standard Industrial Classification code (sic_code), and treatment level. This table is evaluated for duplicates and joining errors, and it is then used to join facility information to output tables with nutrient concentration, effluent flow and/or load calculation data.

3.5. Sic_Codes Table

The sic_codes table contains the Standard Industrial Classification codes and descriptions. It is evaluated for duplicates and joining errors and is used for joining purposes throughout the program. This table can be used to remove facilities with select sic_codes from all output. If the sic_code does not appear in this table, facilities with those sic_codes will not have loads calculated.

3.6. State_Expansion Table

The State_Expansion table contains three fields—(1) State (2 letter abbreviation), (2) Expansion_GroupName, and (3) Expansion_Group. This table is used to designate a concentric expansion of area around a missing nutrient concentration data point, and the user may create as many expansion groups as they deem necessary. Figure 2 is an example of how an expansion group for Missouri could be defined (as in the State_Expansion table in the exampleDataset.accdb). More information on how the State_Expansion table is used in PSLoadEsT is provided in section 5.9.1.

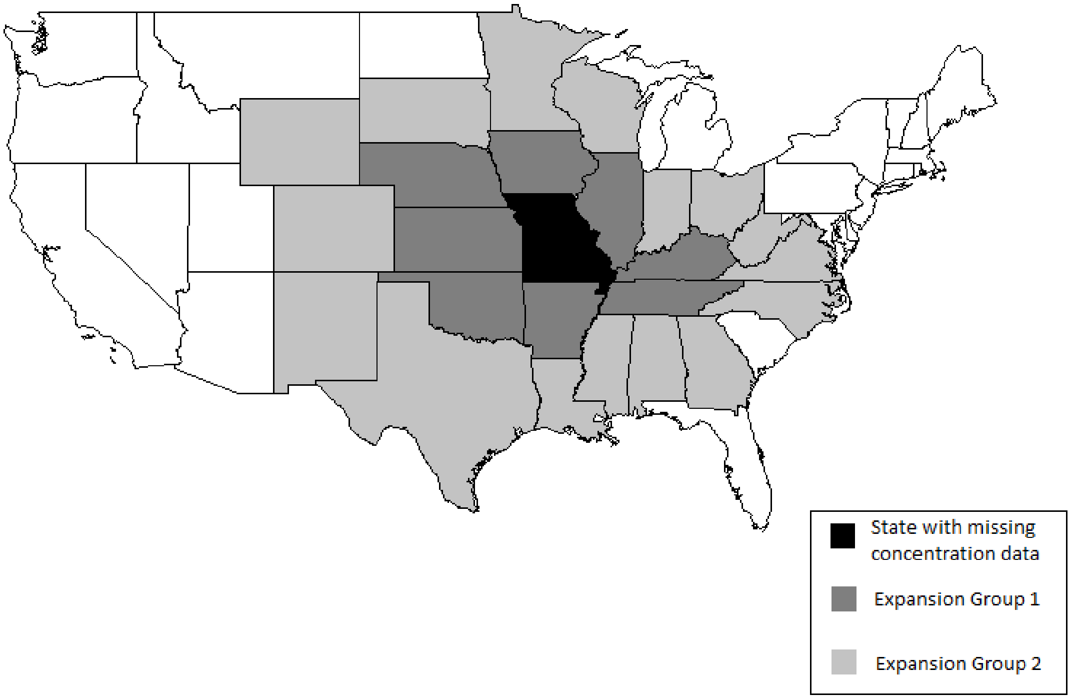


Figure 2. Example of expansion groups for Missouri. If insufficient concentration data exist in the State of Missouri, Expansion Group 1 is tested. If insufficient data exist in Expansion Group 1, Expansion Group 2 is tested. Expansion Group 2 includes Expansion Group 1. If insufficient data exist in Expansion Group 2, no expansion group based substitution is made.

3.7. National_Medians Table

The National_Medians table contains national medians for nutrient concentration by NPDES, sic_code, treatment_level, and parameter code. Assuming there are an insufficient number of available concentration values in all State_Expansion groups for a given missing data point, the National_Medians table will be used by the program. More information on how substitutions are applied is provided in section 5.9.1. It is important to note that even the National_Medians table may be incomplete and have missing nutrient concentration data.

3.8. Rubin_TPC Table

The Rubin_TPC table is a table of Typical Pollutant Concentration (TPC) values for nutrient concentration by sic_code and parameter code from Steven Rubin (U.S. Environmental Protection Agency, written commun., 2006). The table is the final stop for a missing concentration data point in PSLoadEsT. When all other methods have failed to provide sufficient data for a substitution, the concentrations in this table are applied. More information on how substitutions are applied is provided in section 5.9.1.

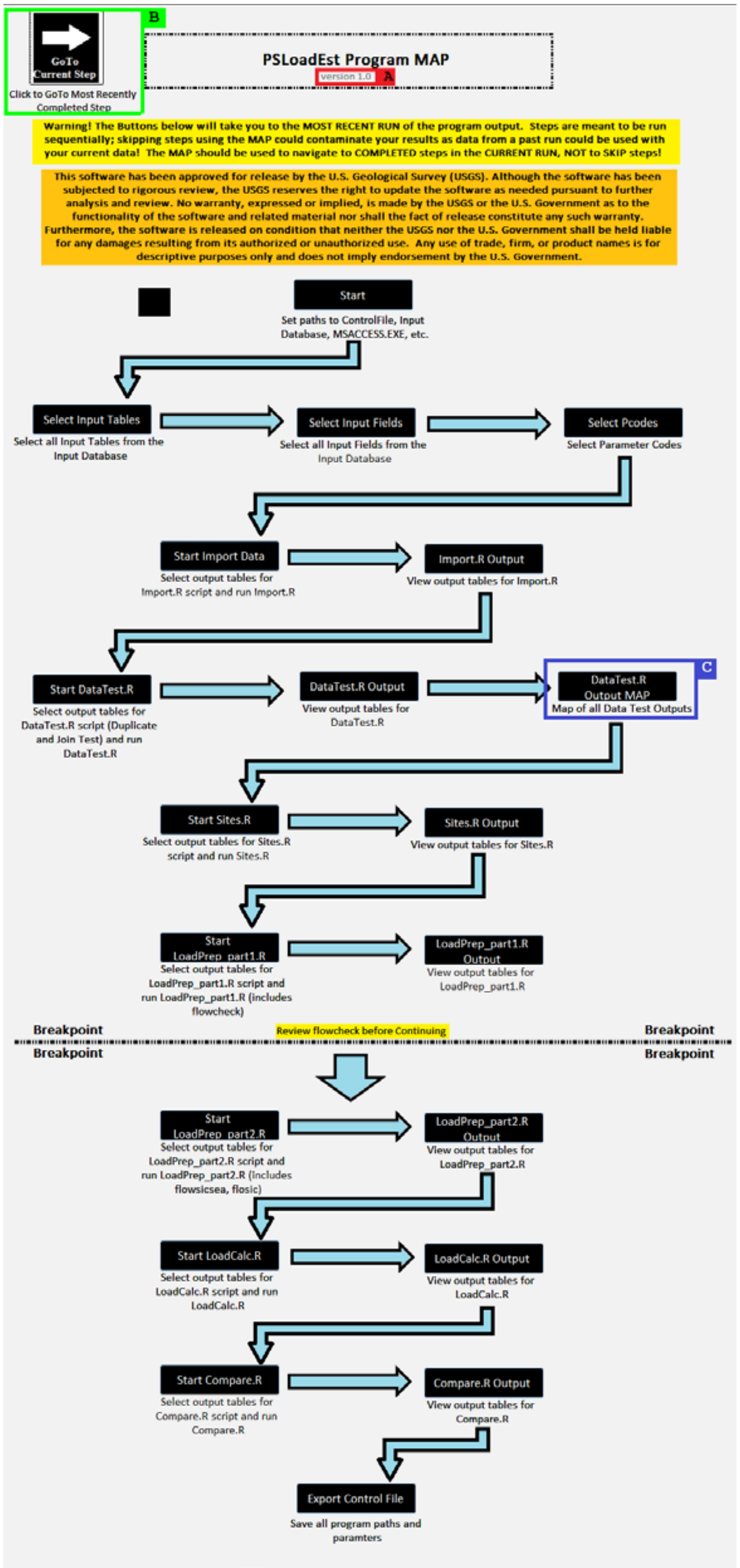
3.9. Example Import Dataset

An example import dataset (PSLoadEsT\exampleDataset\exampleDataset.accdb) is included with PSLoadEsT so that the user can become familiar with the program prior to working with a real dataset. The example data are NOT real and should never be used for anything other than learning the program. The example data were designed to contain duplicates, inaccuracies, missing values, and high flows, so that the user will be exposed to all types of data problems that can be managed within PSLoadEsT. The exceptions are the sic_codes, State_Expansion and Rubin_TPC tables. These tables are provided in Skinner and Maupin (2018) and assuming the user is satisfied with the expansion groups in the State_Expansion table and is analyzing the same nutrients designated by the parameter codes in the Rubin_TPC table, the user may use these tables as real data.

4. Navigating PSLoadEsT

4.1. Program MAP

The initial launch page and core navigation interface for PSLoadEsT is the Program MAP (fig. 3). The version of PSLoadEsT being executed is given at the top of the Program MAP.



B
GoTo Current Step
Click to GoTo Most Recently Completed Step
PSLoadEst Program MAP
version 1.0
A
Start
Set paths to ControlFile, Input Database, MSACCESS.EXE, etc.
Select Input Tables
Select all Input Tables from the Input Database
Select Input Fields
Select all Input Fields from the Input Database
Select Pcodes
Select Parameter Codes
Start Import Data
Select output tables for Import.R script and run Import.R
Import.R Output
View output tables for Import.R
Start DataTest.R
Select output tables for DataTest.R script (Duplicate and Join Test) and run DataTest.R
DataTest.R Output
View output tables for DataTest.R
DataTest.R Output MAP
Map of all Data Test Outputs
C
Start Sites.R
Select output tables for Sites.R script and run Sites.R
Sites.R Output
View output tables for Sites.R
Start LoadPrep_part1.R
Select output tables for LoadPrep_part1.R script and run LoadPrep_part1.R (includes flowcheck)
LoadPrep_part1.R Output
View output tables for LoadPrep_part1.R
Breakpoint
Review flowcheck before Continuing
Breakpoint
Breakpoint
Breakpoint
Start LoadPrep_part2.R
Select output tables for LoadPrep_part2.R script and run LoadPrep_part2.R (includes flowsicsea, flosic)
LoadPrep_part2.R Output
View output tables for LoadPrep_part2.R
Start LoadCalc.R
Select output tables for LoadCalc.R script and run LoadCalc.R
LoadCalc.R Output
View output tables for LoadCalc.R
Start Compare.R
Select output tables for Compare.R script and run Compare.R
Compare.R Output
View output tables for Compare.R
Export Control File
Save all program paths and paramters

Warning! The Buttons below will take you to the MOST RECENT RUN of the program output. Steps are meant to be run sequentially; skipping steps using the MAP could contaminate your results as data from a past run could be used with your current data! The MAP should be used to navigate to COMPLETED steps in the CURRENT RUN, NOT to SKIP steps!

This software has been approved for release by the U.S. Geological Survey (USGS). Although the software has been subjected to rigorous review, the USGS reserves the right to update the software as needed pursuant to further analysis and review. No warranty, expressed or implied, is made by the USGS or the U.S. Government as to the functionality of the software and related material nor shall the fact of release constitute any such warranty. Furthermore, the software is released on condition that neither the USGS nor the U.S. Government shall be held liable for any damages resulting from its authorized or unauthorized use. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Figure 3. PSLoadEsT Program MAP, where PSLoadEsT version is shown at top (red box, A), GoTo Current Step button top left (green box, B), and button to go to sub-map, Data Test MAP, halfway down on the right (blue box, C). Steps are to be executed sequentially; the Program MAP should never be used to skip steps. While the user can navigate through the steps from the Program MAP, it is not necessary to return to the Program MAP to execute the next step. PSLoadEsT will automatically guide the user through the correct program execution path when the user clicks the “Start” button at the top.

The Program MAP allows the user to navigate through previously completed steps but should NEVER be used to skip steps. Attempting to skip steps using the Program MAP will at best terminate the program and at worst give contaminated results.

Although the Program MAP allows the user to navigate steps in the program, it is not necessary to run through the program clicking on each step in the map. Once the user clicks "start" on the Program MAP, each step will complete in order without triggering the Program MAP. The "GoTo Current Step" button in the upper left-hand corner of the Program MAP will take the user to the most recently completed step. If the program is closed and re-opened, PSLoadEsT will automatically open at the most recently completed step.

If the Program MAP is used to navigate to a previously complete step, the user must continue forward from that step to complete the program. DO NOT change user input from a previous step and then use the Program MAP to skip to a step several steps ahead, as this will cause either program failure or contaminated results. To restart the program with a different dataset, the user should begin again at the "Start" screen.

The Program MAP can be accessed from any screen in the program by clicking on the Program MAP icon (fig. 4) in the upper right-hand corner of any screen in the program.

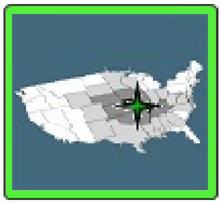
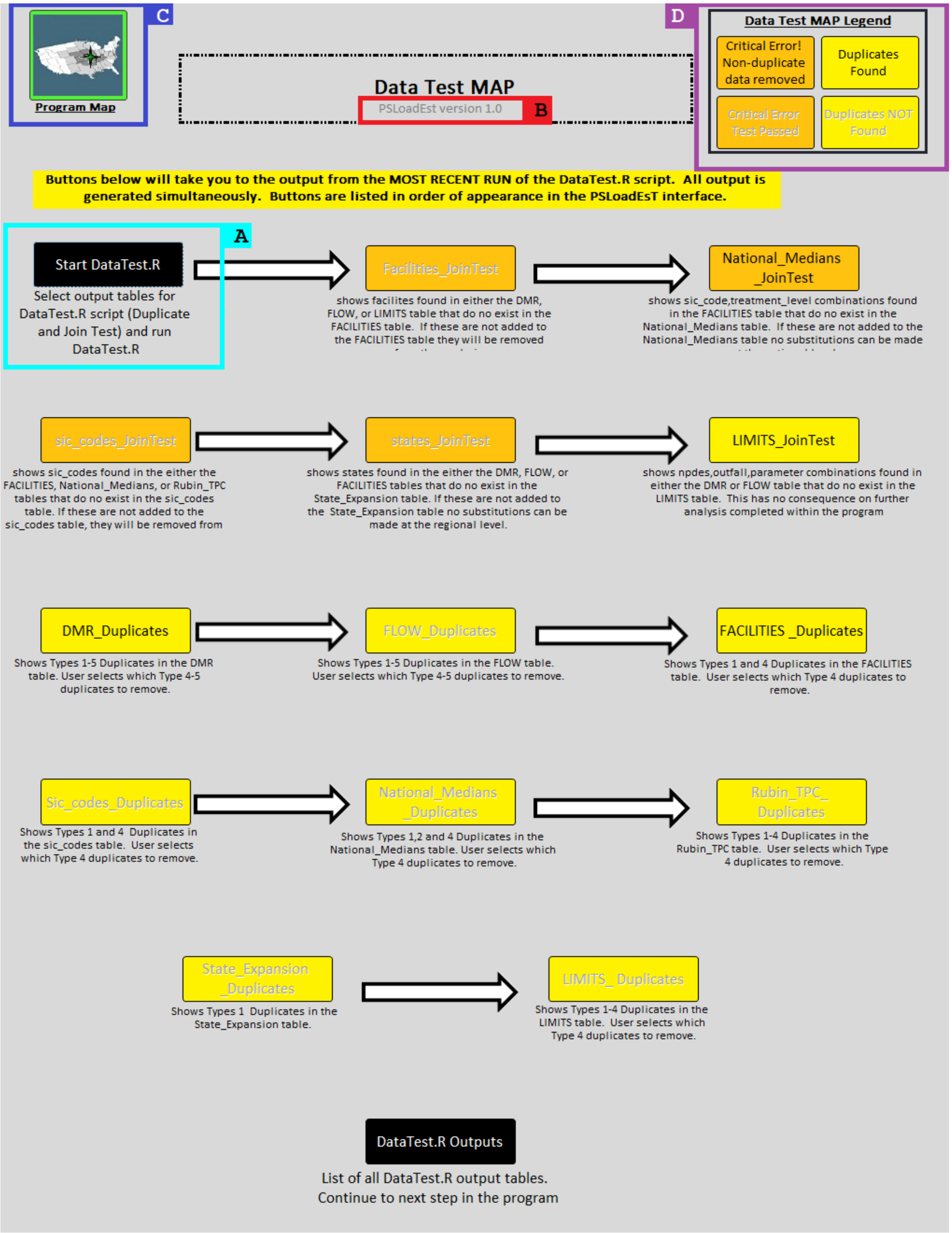


Figure 4. Program MAP icon.

4.1.1. DataTest.R Output MAP

The submap of the DataTest.R script output can be accessed from the Program MAP (fig. 3) by clicking on the "DataTest.R Output MAP," which sends the user to the data test submap (fig. 5). The DataTest.R Output MAP displays all output from the DataTest.R script including joining errors (mismatches) and duplicates in all the input tables for PSLoadEsT. Only output tables with records can be selected (or have enabled buttons with black text) from this submap, allowing the user to quickly view which data testing procedures passed and which may have failed. If, for example, the FLOW table does not contain duplicates, the text on the button for FLOW_Duplicates will be grayed out and disabled.



Buttons below will take you to the output from the MOST RECENT RUN of the DataTest.R script. All output is generated simultaneously. Buttons are listed in order of appearance in the PSLoadEsT interface.

Select output tables for DataTest.R script (Duplicate and Join Test) and run DataTest.R

shows facilites found in either the DMR, FLOW, or LIMITS table that do no exist in the FACILITIES table. If these are not added to the FACILITIES table they will be removed

shows sic_code,treatment_level combinations found in the FACILITIES table that do no exist in the National_Medians table. If these are not added to the National_Medians table no substitutions can be made

shows sic_codes found in the either the FACILITIES, National_Medians, or Rubin_TPC tables that do no exist in the sic_codes table. If these are not added to the sic_codes table, they will be removed from

shows states found in the either the DMR, FLOW, or FACILITIES tables that do no exist in the State_Expansion table. If these are not added to the State_Expansion table no substitutions can be made at the regional level.

shows npdes,outfall,parameter combinations found in either the DMR or FLOW table that do no exist in the LIMITS table. This has no consequence on further analysis completed within the program

Shows Types 1-5 Duplicates in the DMR table. User selects which Type 4-5 duplicates to remove.

Shows Types 1-5 Duplicates in the FLOW table. User selects which Type 4-5 duplicates to remove.

Shows Types 1 and 4 Duplicates in the FACILITIES table. User selects which Type 4 duplicates to remove.

Shows Types 1 and 4 Duplicates in the sic_codes table. User selects which Type 4 duplicates to remove.

Shows Types 1,2 and 4 Duplicates in the National_Medians table. User selects which Type 4 duplicates to remove.

Shows Types 1-4 Duplicates in the Rubin_TPC table. User selects which Type 4 duplicates to remove.

Shows Types 1 Duplicates in the State_Expansion table.

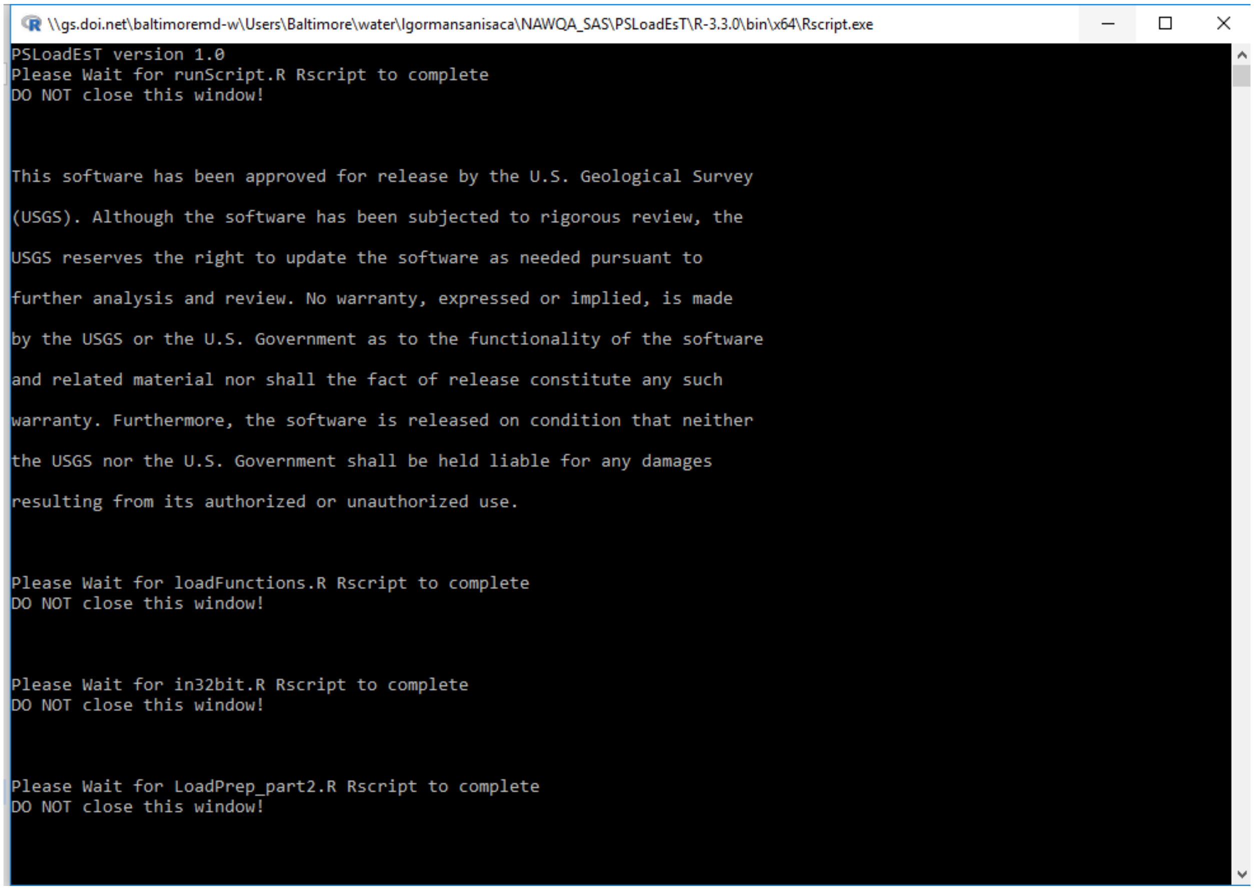
Shows Types 1-4 Duplicates in the LIMITS table. User selects which Type 4 duplicates to remove.

List of all DataTest.R output tables.
Continue to next step in the program

Figure 5. PSLoadEsT Data Test MAP displays the available output screens from the DataTest.R script. All output is generated in a single step (DataTest.R, light-blue box, A); the map shown above is as it will appear after running the exampleDataset.accdb file. PSLoadEsT version is shown at top (red box, B), and the return to main ProgramMAP button top left (blue box, C). Buttons with results enabled with black text, buttons without results disabled with gray text; Orange buttons represent critical errors where non-duplicate data will be removed from consideration, and yellow buttons represent duplicate data errors that may require user input to avoid load calculation errors. The button-color/text-color guide is shown in the upper-right corner in the Data Test MAP Legend (purple box, D). Buttons are displayed in the order in which the screens will appear during normal program execution (without the use of the map); screens associated with disabled buttons will skipped.

4.2. Rscript.exe window

At each step of PSLoadEsT in which an R script is executed the user will see a black Rscript.exe window (fig. 6) displayed with the R script name, PSLoadEsT version, USGS software disclaimer, and code progress messages. This window should never be closed manually by the user, when the R script is complete, the window will close and the PSLoadEsT\Rscripts\Bounce.accdb file will open to return the user to the PSLoadEsT interface.



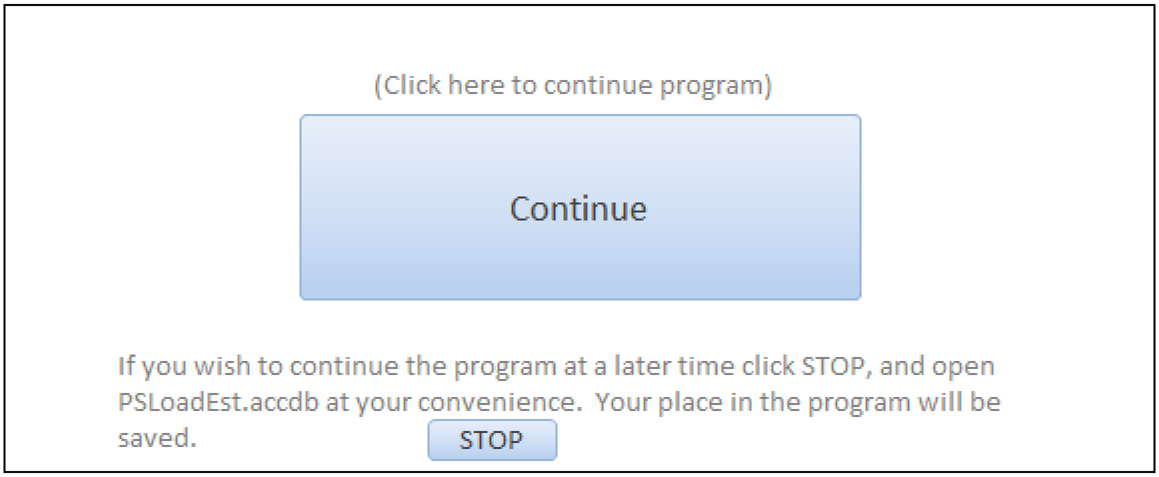
\\gs.doi.net\baltimoremd-w\Users\Baltimore\water\lgormansanisaca\NAWQA_SAS\PSLoadEsT\R-3.3.0\bin\x64\Rscript.exe

```
PSLoadEsT version 1.0
Please Wait for runScript.R Rscript to complete
DO NOT close this window!


This software has been approved for release by the U.S. Geological Survey

(USGS). Although the software has been subjected to rigorous review, the

USGS reserves the right to update the software as needed pursuant to

further analysis and review. No warranty, expressed or implied, is made

by the USGS or the U.S. Government as to the functionality of the software

and related material nor shall the fact of release constitute any such

warranty. Furthermore, the software is released on condition that neither

the USGS nor the U.S. Government shall be held liable for any damages

resulting from its authorized or unauthorized use.


Please Wait for loadFunctions.R Rscript to complete
DO NOT close this window!


Please Wait for in32bit.R Rscript to complete
DO NOT close this window!


Please Wait for LoadPrep_part2.R Rscript to complete
DO NOT close this window!
```

Figure 6. Example of the Rscript.exe window that appears at each step of PSLoadEsT execution.

4.3. The Bounce File

The sole purpose of the PSLoadEsT\Rscripts\Bounce.accdb file is to return the user to the primary program interface, the PSLoadEsT.accdb file. In order for the R scripts to update the PSLoadEsT.accdb interface file, the file must be closed, therefore the Bounce.accdb is called at the end of every R script so that the user can automatically be returned to the interface without having to manually reopen the program. If the Bounce.accdb file is never triggered and the interface must be re-opened manually, then the program is experiencing a CRITICAL ERROR, and the user should verify that all minimum requirements (section 2.1) including the installation of the required Microsoft Access driver are met. The only screen in the Bounce.accdb file is shown in figure 7.



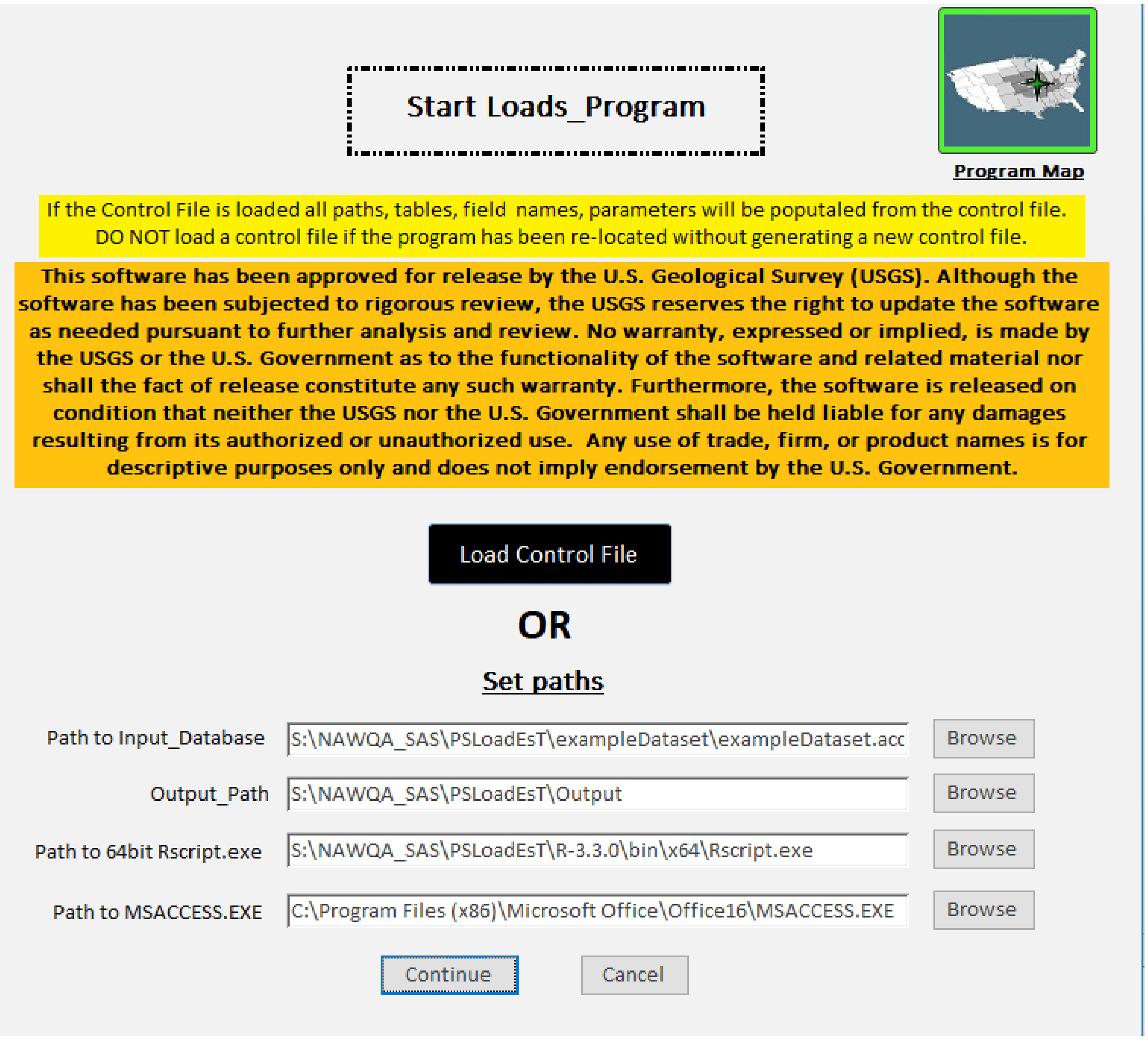
If you wish to continue the program at a later time click STOP, and open PSLoadEst.accdb at your convenience. Your place in the program will be saved.

Figure 7. The only screen in the Bounce.accdb file, which serves the sole purpose of returning the user to the PSLoadEsT.accdb file. This screen will appear after the execution of every R script.

5. Executing PSLoadEsT

5.1. Start PSLoadEsT

Click “Start” on the Program MAP (fig. 3) to execute PSLoadEsT. The “Start PSLoadEsT” screen (fig. 8) will be triggered and the Program MAP will close. There are two methods to start execution of PSLoadEsT; the user can either run through the setup screens (setting user specific paths and table and field name crosswalks) or, if the program has been previously executed from the same location and all paths and crosswalks have been exported to a ControlFile.accdb file, the user can click “Load Control File” and skip all setup screens and go directly into the Import.R step in section 5.5.



If the Control File is loaded all paths, tables, field names, parameters will be poputaled from the control file.
DO NOT load a control file if the program has been re-located without generating a new control file.

This software has been approved for release by the U.S. Geological Survey (USGS). Although the software has been subjected to rigorous review, the USGS reserves the right to update the software as needed pursuant to further analysis and review. No warranty, expressed or implied, is made by the USGS or the U.S. Government as to the functionality of the software and related material nor shall the fact of release constitute any such warranty. Furthermore, the software is released on condition that neither the USGS nor the U.S. Government shall be held liable for any damages resulting from its authorized or unauthorized use. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Path to Input_Database: S:\NAWQA_SAS\PSLoadEsT\exampleDataset\exampleDataset.acc

Output_Path: S:\NAWQA_SAS\PSLoadEsT\Output

Path to 64bit Rscript.exe: S:\NAWQA_SAS\PSLoadEsT\R-3.3.0\bin\x64\Rscript.exe

Path to MSACCESS.EXE: C:\Program Files (x86)\Microsoft Office\Office16\MSACCESS.EXE

Figure 8. The "Start" screen begins execution of PSLoadEsT by setting up user specific paths. If a PSLoadEsT ControlFile.accdb file is available, it can be loaded from this screen by clicking "Load Control File."

5.1.1. First Execution of PSLoadEsT

When executing PSLoadEsT for the first time, the user must designate the required program paths on the Start PSLoadEsT screen (fig. 8). Control files cannot be loaded on the first execution. Invalid path designations will result in program failure.

5.1.1.1. User Designated Paths

The user must designate the “Path to Input_Database,” which is the path to the Microsoft Access input data file formatted as an .accdb file with the six required tables from section 3 and a path to an existing directory for all output files (“Output_Path”). It is recommended that the user execute PSLoadEsT with the provided PSLoadEsT\exampleDataset\exampleDataset.accdb input data file to learn how the program operates prior to loading new data.

Using the “browse” buttons to designate paths is highly recommended as it eliminates the possibility of program errors due to typing errors in the required paths. Both user designated paths must not contain spaces or special characters. Note that the results from PSLoadEsT can be output to any directory on the user’s system that does not contain spaces or special characters; using the included output directory is not required.

5.1.1.2. Auto Paths

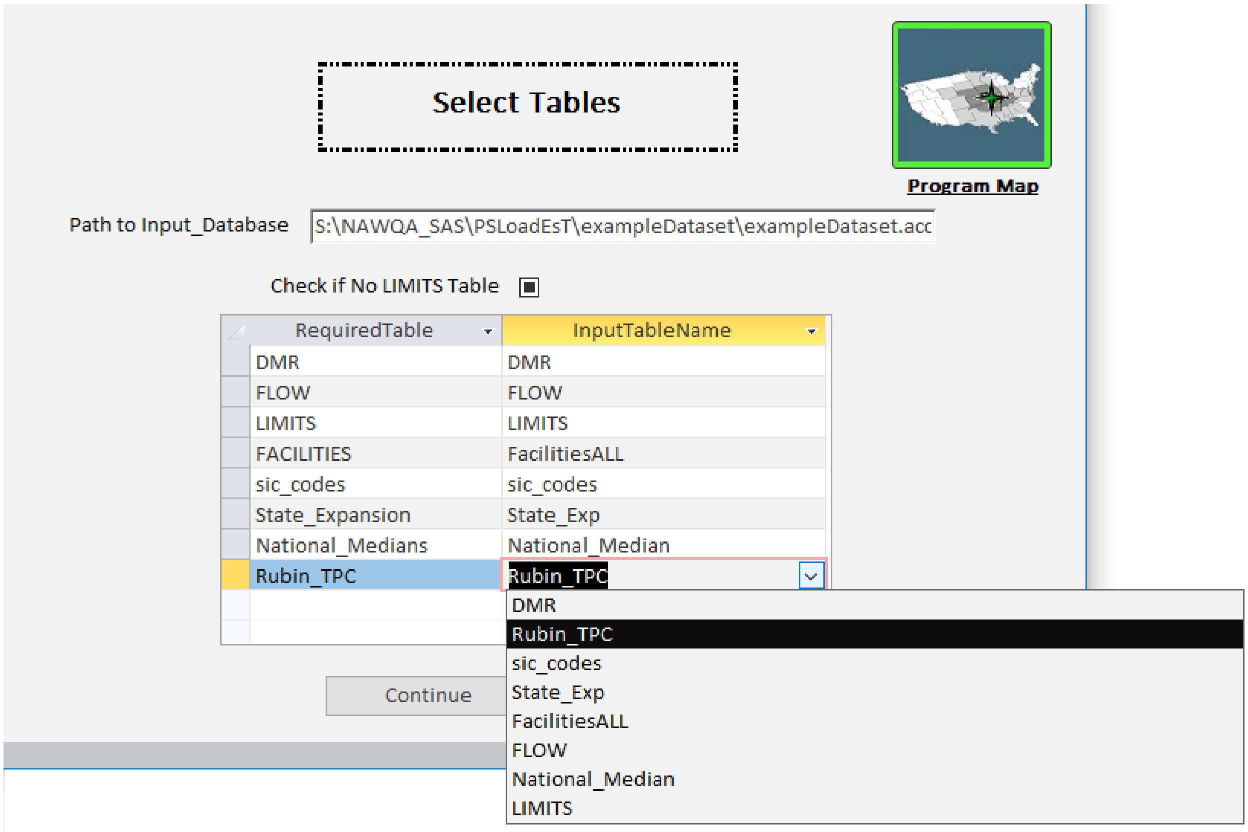
The “Path to 64-bit Rscript.exe” and the “Path to MSACCESS.EXE” should be automatically determined by PSLoadEsT when the “Start PSLoadEsT” screen loads. However, the user should always double check that these paths are accurate. The “Path to 64-bit Rscript.exe” is the path to the Rscript.exe file within the PSLoadEsT program directories here—PSLoadEsT\R-3.3.0\bin\x64\Rscript.exe. The “Path to MSACCESS.EXE” is the path to the user’s version of Microsoft Access, which should be a 32-bit version. This file is typically located here—C:\Program Files (x86)\Microsoft Office\Office(version number)\MSACCESS.EXE.

5.1.2. Control File Execution

To run PSLoadEsT using a previously generated ControlFile.accdb file, click “Load Control File” on the “Start” screen and browse to the location of the previously generated ControlFile.accdb file. It is not necessary to input user paths on the “Start” screen as these paths will be loaded from the ControlFile.accdb. The tables in the ControlFile.accdb file must be formatted exactly as they were output from PSLoadEsT; altering tables could cause program failure. Only import a ControlFile.accdb file that contains information from a PSLoadEsT run using the same import database with the same table and field names, from the same location on the user’s system. After the ControlFile.accdb file is imported, PSLoadEsT will navigate to the Import.R step (section 5.5).

5.2. Select Tables

The “Select Tables” screen (fig. 9) allows the user to crosswalk (field map) the import table names for the six required input tables and optional LIMITS table from section 3. The input data file may contain additional tables and the required tables (and optional LIMITS table) can have user-specific names. The “RequiredTable” field designates the type of table in PSLoadEsT, and the “InputTableName” field designates the user specific table name in the input data file. When the user clicks in the “InputTableName” field a dropdown menu will appear with all tables in the user’s input data file. PSLoadEsT comes with these fields populated with tables from the exampleDataset.accdb file, therefore, if executing new data for the first time, the user must select tables from the new data file. If the optional LIMITS table does not exist, the user must check the “Check if No LIMITS Table” checkbox above the table name selection table.



Path to Input_Database S:\NAWQA_SAS\PSLoadEsT\exampleDataset\exampleDataset.acc

Check if No LIMITS Table ■

RequiredTable	InputTableName
DMR	DMR
FLOW	FLOW
LIMITS	LIMITS
FACILITIES	FacilitiesALL
sic_codes	sic_codes
State_Expansion	State_Exp
National_Medians	National_Median
Rubin_TPC	Rubin_TPC

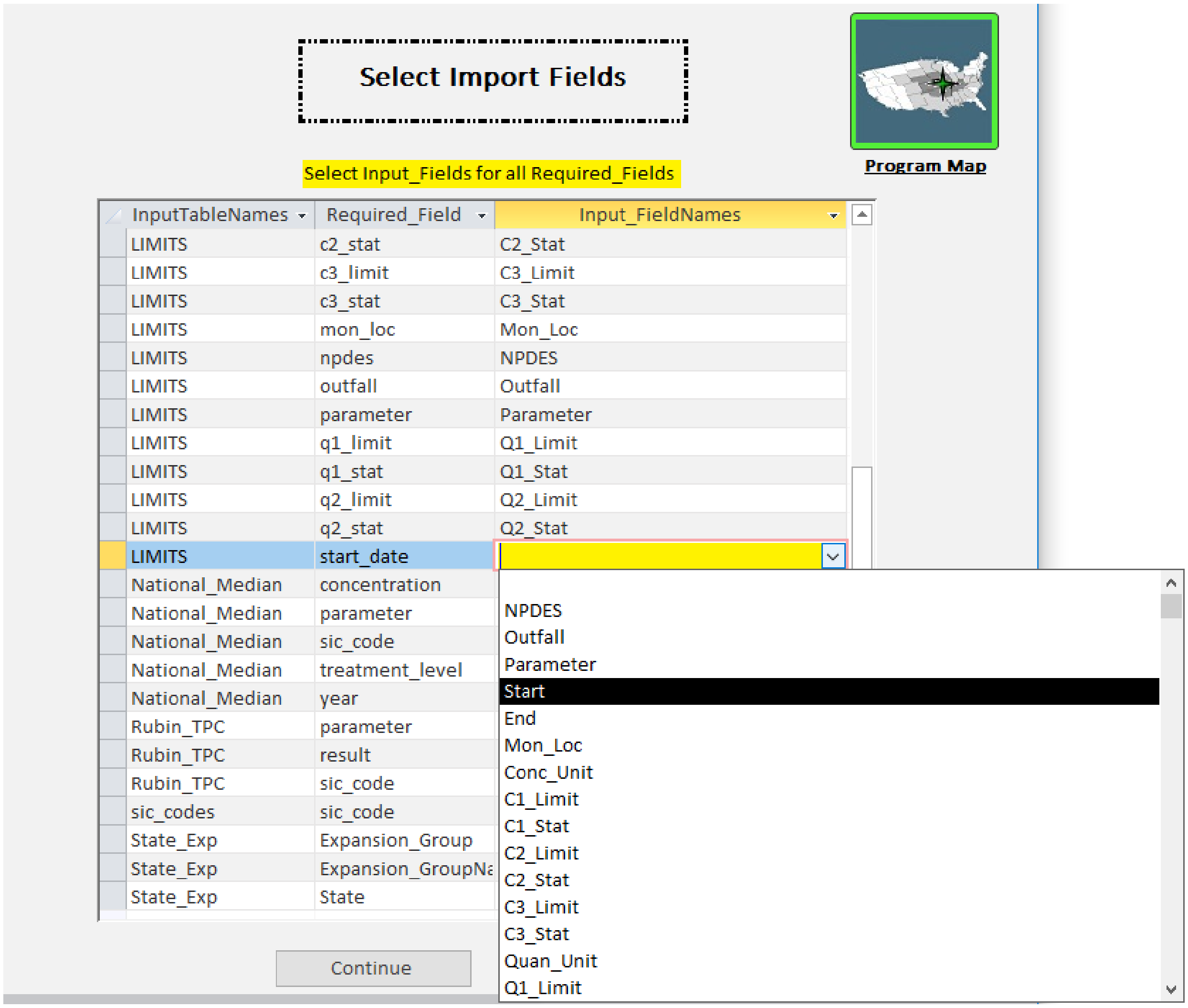
Figure 9. The Select Tables screen creates a crosswalk between the PSLoadEsT required tables ("RequiredTable") and the user-specific table names ("InputTableName") from the input database ("Path to Input_Database"). PSLoadEsT comes loaded with the table names from the PSLoadEsT\exampleDataset\exampleDataset.accdb file. To change the "InputTableName" click in the cell to activate the dropdown menu, which will auto-populate with all of the tables present in the user's input database.

5.2.1. Fields.R

When the user clicks "Continue" on the "Select Tables" screen, a black Rscript.exe window will appear as the first R script, Fields.R, executes. The Fields.R script reads in the user designated tables and attempts to find the required fields necessary for PSLoadEsT shown in table 1. However, it is not necessary to have the exact field names given in table 1 as long as the fields shown in table 1 are present in the input table. The Fields.R script outputs a temporary file within the PSLoadEsT\Rscripts directory called TempFields.csv that will be loaded and appear on the next screen of the PSLoadEsT interface.

5.3. Select Import Fields

The “Select Import Fields” screen (fig. 10) allows the user to create a crosswalk between the required fields from table 1 and the user-specific field names present in the input data file. The Fields.R script from section 5.2.1 populates as much of this crosswalk as can be populated automatically matching field names regardless of case. The “InputTableNames” column shows the user-specific table names from the input data file. The “Required_Field” column shows the general required fields for PSLoadEsT from table 1, and the “Input_FieldNames” column is where the user-specific field names are populated. Any fields that cannot be automatically matched in the Fields.R script will have the associated cell in the “Input_FieldNames” column highlighted in yellow. The user must select the appropriate field name from the dropdown menu on all Input_FieldNames highlighted in yellow. Once all Input_FieldNames have been populated, it is recommended that the user click in another column to save all selections from the Input_FieldNames column or press “Crtl+S.” No R script is triggered from the “Select Import Fields” screen “Continue” button.



InputTableNames	Required_Field	Input_FieldNames
LIMITS	c2_stat	C2_Stat
LIMITS	c3_limit	C3_Limit
LIMITS	c3_stat	C3_Stat
LIMITS	mon_loc	Mon_Loc
LIMITS	npdes	NPDES
LIMITS	outfall	Outfall
LIMITS	parameter	Parameter
LIMITS	q1_limit	Q1_Limit
LIMITS	q1_stat	Q1_Stat
LIMITS	q2_limit	Q2_Limit
LIMITS	q2_stat	Q2_Stat
LIMITS	start_date	
National_Median	concentration	
National_Median	parameter	
National_Median	sic_code	
National_Median	treatment_level	
National_Median	year	
Rubin_TPC	parameter	
Rubin_TPC	result	
Rubin_TPC	sic_code	
sic_codes	sic_code	
State_Exp	Expansion_Group	
State_Exp	Expansion_GroupNa	
State_Exp	State	

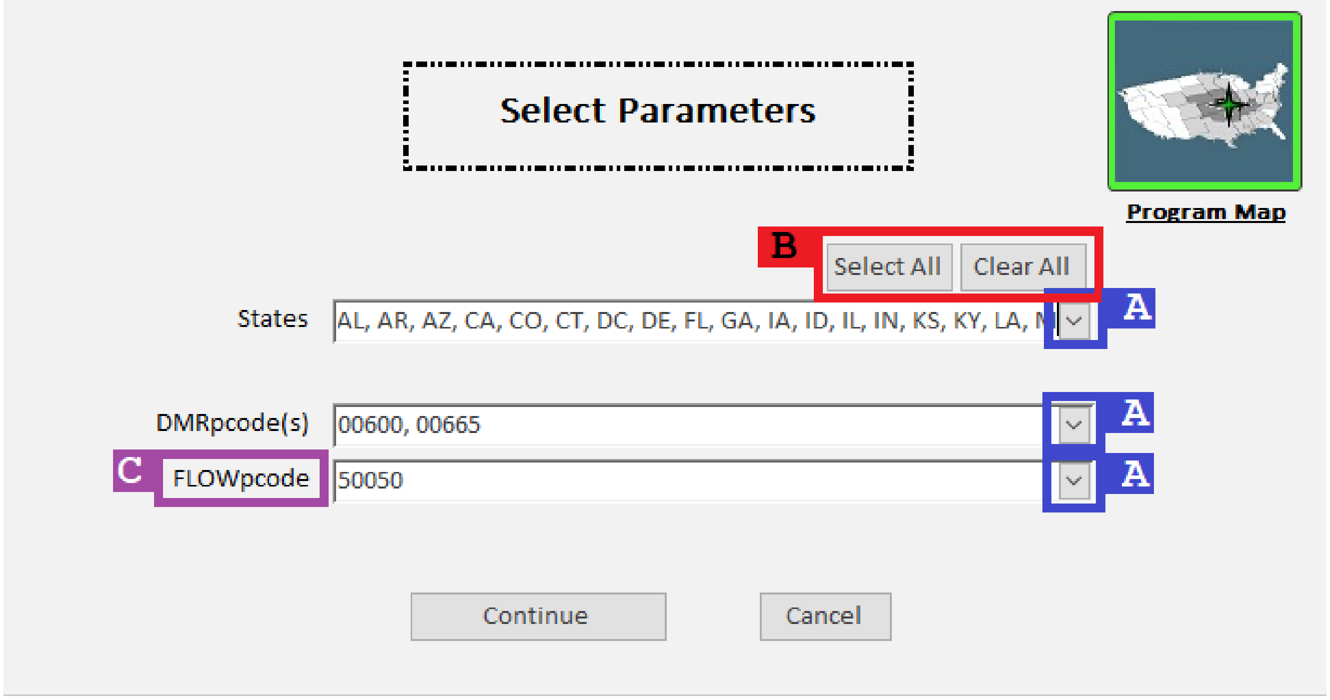
Figure 10. Select Fields screen is populated with the user-specific table names ("InputTableNames") and the "Input_FieldNames" that match the PSLoadEsT "Required_Field." Required fields without associated input field names will be highlighted in yellow and can be manually set by the user in a dropdown menu.

5.3.1. Select Import Fields Example Dataset Execution

The only “Input_Field” highlighted in yellow as a field not in the exampleDataset.accdb file is the required start_date field from the LIMITS table. The user should designate the “Start” field from the dropdown menu in the “Input_Field” column in the LIMITS table for the “Required_Field” start_date in the LIMITS table.

5.4. Select Parameters

The “Select Parameters” screen (fig. 11) allows the user to select which States, nutrient-concentration parameter codes, and effluent-flow parameter codes to run through PSLoadEsT. States listed in the dropdown menu are the unique States determined by the first two letters of the NPDES field in the user’s specified DMR table; only States within the DMR table will be available for analysis. The user may select as many or as few States as desired, however, selecting a few States or even just one State will severely limit the data available to make median substitutions for missing nutrient-concentration data discussed in section 5.9.1. The “DMRpcodes” shown in the dropdown are the nutrient-concentration as parameter codes from the DMR input table that the user wants to use in the final load calculation; the user can select multiple “DMRpcodes.” The “FLOWpcode” is the single effluent-flow parameter code to be used in the final load calculation; only those effluent-flow parameter codes in the user’s FLOW table are given as options. The Select Parameters step allows the user to subset the input data to be analyzed in the current run from within the program, which allows the user to use the same master input file with all compiled data when analyzing different regions and/or constituents without creating a separate input data file. No R script is executed from the “Select Parameters” screen “Continue” button.


Select Parameters

States: AL, AR, AZ, CA, CO, CT, DC, DE, FL, GA, IA, ID, IL, IN, KS, KY, LA, M

DMRpcode(s): 00600, 00665

FLOWpcode: 50050

Figure 11. Select Parameters screen allows the user to designate which States, DMR (concentration) parameter codes, and FLOW parameter codes to analyze in the current run. The down arrows (blue boxes, A) trigger dropdown menus listing all States, DMR parameter codes, and FLOW parameter codes in the input data file. The “Select All” and “Clear All” (red box, B) buttons operate only on the States selected. Only one FLOW parameter code (purple box, C) can be designated for each run.

5.4.1. Select Parameters Example Dataset Execution

Although the user may select any combination of States and parameter codes when running the exampleDataset.accdb file, it is recommended in the first run that the user select all States and "DMRpcodes" available in the exampleDataset.accdb input tables; the only FLOWpcode available is 50050, so that results are comparable to those included within the PSLoadEsT\Output directory and match the screens shown and described within the documentation, which can be helpful when learning the program.

5.5. Run Import.R

The "Run Import.R" screen allows the user to select which output tables they want to save from the Import.R script output (appendix 2). At this step, the user can select which tables to output to .csv and the Results.accdb file. All tables output to the Results.accdb file will also be output to .csv files, and the "Table" field will be the name given to the .csv file and the table name in the Results.accdb file. The Results.accdb file will also be created at this step. For more detailed instructions on how to select output tables see appendix 2-B.

5.5.1. Import.R

The Import.R script loads all user's input tables and replaces original user-specific names for "Required_Fields" with the generic "Required_Fields" names used in PSLoadEsT (table 1), and subsets the tables so that only the States, DMRpcodes, and FLOWpcode selected by the user in section 5.4 are present. The State field is added to the DMR, FLOW, FACILITIES and LIMITS (optional) tables using the first two letters from the "Required_Field" NPDES. Year and month fields are created based on the date field using the functions from the lubridate R package (Grolemund and Wickham, 2011). The R script also creates a seasons field using the seasons() function with breaks at February, May, August, and November from the smwrBase R package (Lorenz, 2015). The zeroPad() function from the dataRetrieval R package (Hirsch and De Cicco, 2015) is used to ensure that the leading zeros from the 5-digit parameter codes and 3-digit outfall designations are not lost in import. Only monitoring locations (mon_loc "Required_Field") designated as the letter "l" are considered for PSLoadEsT execution. A mon_loc designation of "1" indicates that the flow and/or sample was measured/collected where the facility discharges to a stream; all other mon_loc designations are removed from all output tables in the Import.R script.

5.6. Run DataTest.R

The DataTest.R script checks all the required input tables (and optional LIMITS table) from section 3 for joining errors (mismatches between tables) and duplicate records. On the "Run DataTest.R" screen the user can select output tables produced in the DataTest.R script to save within the Results.accdb output database. By default, all tables with records from the DataTest.R script are output in .csv format to the users output directory; tables without records are not output to .csv files or the Results.accdb output database. Following the execution of DataTest.R, the user will be guided through a series of screens each containing output tables from the DataTest.R script. If no tables are found, the user will see a "DataTest.R Complete" screen. The user also can navigate to the output screens from the DataTest.R from the DataTest.R Output MAP detailed in section 4.1.1.

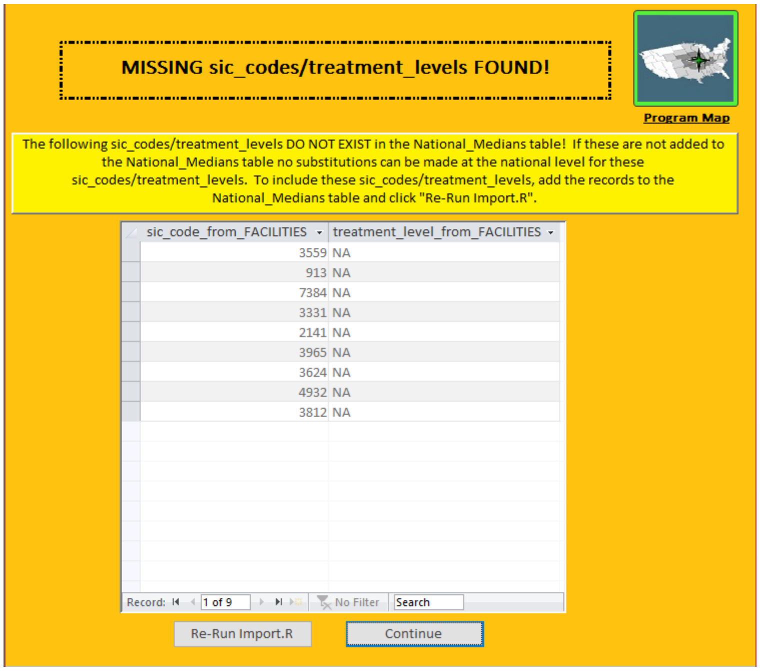
5.6.1. DataTest.R

PSLoadEsT uses the joining functions in the R package dplyr (Wickham and Francois, 2016) to find joining errors (records that cannot be linked between tables due to mismatches in the fields used to join the tables) between the required tables (and optional LIMITS table). Joining errors can cause facilities, sic_codes, and/or treatment_levels to be omitted from consideration by PSLoadEsT, and therefore must be evaluated by the user. However, the user can still continue executing PSLoadEsT with the understanding that the facilities, sic_codes, and/or treatment_levels found to be joining errors will be omitted from consideration.

Duplicates found in any of the required tables must be selected for removal by the user. Duplicates are critical data errors that can result in highly inaccurate final load calculations. Duplicates in multiple input tables will multiply when the tables are joined creating an exponentially larger problem as execution continues. The DataTest.R script highlights these common data problems so that the user can either address the issues within the program (in the case of duplicates) or fix the input tables and re-import the data (required for joining errors).

5.6.2. Joining Errors

Joining errors are the first type of data error highlighted by PSLoadEsT because these errors could require the user to re-import the required input tables after errors have been addressed. All facilities, sic_codes, and/or treatment_levels shown on "_JoinTest" screens (appendix 1) will be omitted from PSLoadEsT execution from this point forward, which may or may not be the desired effect. The user may select either to re-import the required tables or to continue thereby removing facilities, sic_codes, and /or treatment_levels on all "_JoinTest" screens. An example of "_JoinTest" screen is shown in figure 12.



MISSING sic_codes/treatment_levels FOUND!

The following sic_codes/treatment_levels DO NOT EXIST in the National_Medians table! If these are not added to the National_Medians table no substitutions can be made at the national level for these sic_codes/treatment_levels. To include these sic_codes/treatment_levels, add the records to the National_Medians table and click "Re-Run Import.R".

sic_code_from_FACILITIES	treatment_level_from_FACILITIES
3559	NA
913	NA
7384	NA
3331	NA
2141	NA
3965	NA
3624	NA
4932	NA
3812	NA

Record: 1 of 9 No Filter Search

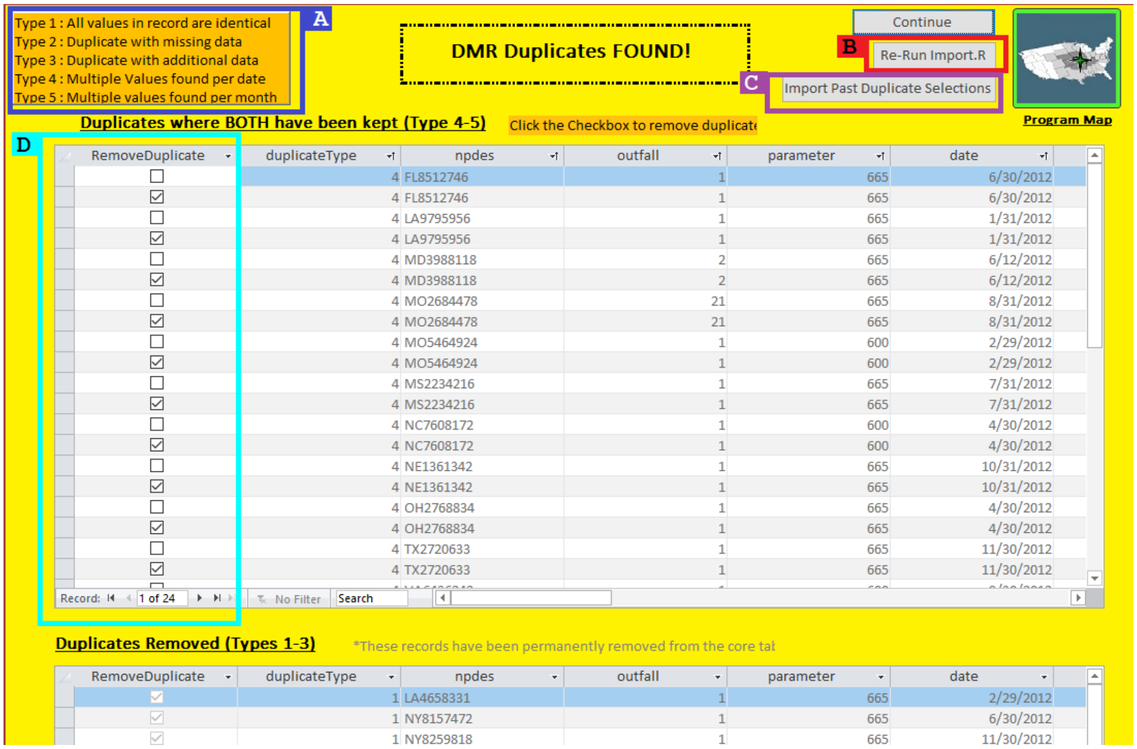
Figure 12. The National_Medians_JoinTest Screen as it will appear when running the included exampleDataset.accdb file. When running the exampleDataset.accdb file, the user should assume that these sic_codes are meant to be removed from consideration when substituting concentration values using the National_Medians table.

5.6.2.1. Joining Errors in the Example Dataset Execution

When running the exampleDataset.accdb file, joining errors are in the LIMITS and National_Medians tables. Sic_code/treatment_level combinations found in the FACILITIES table and not found in the National_Medians table will not have National_Medians applied as substitutions for missing nutrient concentration values as discussed in section 5.9.1. Facilities not found in the LIMITS table have little impact on the program and will simply not be output in the LIMITS table found in the output directory; more detail on the LIMITS table is provided in section 3.3. The user should assume that all joining errors are intentional, and that it is not necessary to make any changes to the input tables in the exampleDataset.accdb file.

5.6.3. Duplicates

Duplicates are considered critical errors in PSLoadEsT and should never be included in the load calculation. The five types of duplicates found in DataTest.R are shown in table 2. The “Duplicates FOUND” screen (fig. 13) displays Type 4 and Type 5 duplicates in the first table; the user must evaluate and remove all Type 4 and Type 5 duplicates from the required input tables either within the PSLoadEsT interface (recommended for the exampleDataset.accdb) or by removing duplicates from the input tables and re-importing the data. Duplicates flagged as Types 1-3 (displayed in the bottom table) are automatically removed by PSLoadEsT and saved in a “(tablename)_DuplicatesRemoved.csv” table.



Type 1 : All values in record are identical
Type 2 : Duplicate with missing data
Type 3 : Duplicate with additional data
Type 4 : Multiple Values found per date
Type 5 : Multiple values found per month

DMR Duplicates FOUND!

Duplicates where BOTH have been kept (Type 4-5)

Click the Checkbox to remove duplicat

RemoveDuplicate	duplicateType	npdes	outfall	parameter	date
☐	4	FL8512746	1	665	6/30/2012
☑	4	FL8512746	1	665	6/30/2012
☐	4	LA9795956	1	665	1/31/2012
☑	4	LA9795956	1	665	1/31/2012
☐	4	MD3988118	2	665	6/12/2012
☑	4	MD3988118	2	665	6/12/2012
☐	4	MO2684478	21	665	8/31/2012
☑	4	MO2684478	21	665	8/31/2012
☐	4	MO5464924	1	600	2/29/2012
☑	4	MO5464924	1	600	2/29/2012
☐	4	MS2234216	1	665	7/31/2012
☑	4	MS2234216	1	665	7/31/2012
☐	4	NC7608172	1	600	4/30/2012
☑	4	NC7608172	1	600	4/30/2012
☐	4	NE1361342	1	665	10/31/2012
☑	4	NE1361342	1	665	10/31/2012
☐	4	OH2768834	1	665	4/30/2012
☑	4	OH2768834	1	665	4/30/2012
☐	4	TX2720633	1	665	11/30/2012
☑	4	TX2720633	1	665	11/30/2012

Duplicates Removed (Types 1-3)

*These records have been permanently removed from the core tal

RemoveDuplicate	duplicateType	npdes	outfall	parameter	date
☑	1	LA4658331	1	665	2/29/2012
☑	1	NY8157472	1	665	6/30/2012
☑	1	NY8259818	1	665	11/30/2012

Figure 13. DMR Duplicates FOUND screen from the exampleDataset.accdb file. Duplicate types are shown in the upper left (blue box, A). To remove duplicates outside of PSLoadEsT by altering the input data, click “Re-Run Import.R” (red box, B). To apply previously saved duplicate removal selections (DMR_DuplicatesKept.csv file) for Type 4 and Type 5 duplicates, click “Import Past Duplicate Selections” (purple box, C). Select duplicates to remove in the first column of the top table (light blue box, D). Duplicates of Types 1-3 (bottom table) will be automatically removed.

Table 2. Types of duplicates identified in by the DataTest.R script in PSLoadEsT.

Duplicate type	Description	Removal action
Type 1	All values in record are identical.	Automatic
Type 2	Duplicate record with one or more missing key data and at least one non-missing for same NPDES, outfall, date, and parameter code (e.g., missing c2 value in the DMR table).	Automatic
Type 3	Duplicate with additional non-key data (e.g., single c2 value in multiple records with different entries in a comment field in the DMR table for a given NPDES, outfall, date, and parameter code.	Automatic
Type 4	Duplicate with multiple values for key data (e.g. multiple values for c2 in the DMR table) for a given NPDES, outfall, date, and parameter code.	User selected
Type 5	Duplicate with multiple values for key data by NPDES, outfall, month, and parameter code.	User selected
Type 6	Duplicate resulting from joining the DMR and FLOW tables, where the effluent-flow and nutrient concentration values occur at different dates but within the same month. These duplicates are evaluated in the Sites.R step (section 5.7.5) not in the DataTest.R step.	Not removed, Date from nutrient-concentration value replaced with date of effluent flow value

The records selected by the user to keep from the Type 4 and Type 5 duplicates are saved in a “(tablename)_DuplicatesKept.csv” file. To import a previously saved “(tablename)_DuplicatesKept.csv” file with duplicate selection from a previous run of PSLoadEsT, click “Import Past Duplicate Selections” in the upper right-hand corner of a “Duplicates FOUND” screen. Only “(tablename)_DuplicatesKept.csv” files saved by PSLoadEsT in their original format should be imported to select the Type 4 and Type 5 duplicates for removal.

If only duplicates flagged as Types 1-3 are found, the user will be shown a table of duplicates removed, however, no action is required to continue executing PSLoadEsT. If no duplicates of any type are found, PSLoadEsT will navigate to the next step in the program “Run Sites.R.”

5.6.3.1. Duplicates in the Example Dataset Execution

The DMR table in the exampleDataset.accdb file contains duplicates flagged as Types 1-5 duplicates. Duplicates flagged as Types 1-3 are automatically removed, but the user must select which records to remove from the Type 4 and Type 5 duplicates table. The user can either select these recorded manually within the PSLoadEsT interface or import the included DMR_DuplicatesKept.csv file to select Type 4 and Type 5 duplicates for removal. Failing to remove Type 4 and Type 5 duplicates is not recommended as it will lead to larger calculated loads than should be found.

The FACILITIES table includes Type 1 duplicates that are automatically removed by PSLoadEsT and require no action from the user to complete execution.

5.7. Sites.R

The Sites.R step applies the user’s selections for removal of Type 4 and Type 5 duplicates, generates “(tablename)_remark” tables, evaluates missing effluent-flow values, joins the DMR and FLOW tables, and generates summary statistics for nutrient-concentration and effluent-flow fields.

5.7.1. Duplicate Removal

The Sites.R script also executes the DupFix.R script (if duplicates were found), which applies the user’s selections for removal of Type 4 and Type 5 duplicates prior to all other actions. If duplicates were found in a given table, the DupFix.R script will create the “(tablename)_remark” table with added columns: duplicateFound (binary), duplicateType (numeric), and duplicateRemoved (binary). The Sites.R script creates the “(tablename)_remark” tables for the DMR and FLOW tables assuming no duplicates were found in those tables.

5.7.2. (tablename)_remark Output Tables

The Sites.R script also clears all qualifying text from required numeric fields for nutrient-concentration and effluent flow (table 1). Remark codes are as follows: (1) “<”, (2) “>”, (3) non-numeric or negative, (0) numeric, non-missing, and (NA) missing (see table 2-1 for details on remarkFunc()). These codes are stored in the (tablename)_remark tables for DMR and FLOW, and only numeric values without the accompanying qualifying text are output from the DMR and FLOW tables from this point forward in PSLoadEsT execution. Additional fields (not found in the “Required_Fields” list in table 1) are stripped out of the required tables at this point in the execution.

5.7.3. Missing “q1” Effluent-Flow Values

Missing “q1” values from the FLOW table are replaced with “q2” values, and the FLOW table is subset to only show records where “q1” is not missing. Loads are only computed in PSLoadEsT if a value for effluent flow is present, therefore, if both “q1” and “q2” are missing, no load will be calculated.

5.7.4. Joining DMR and FLOW Tables and Type 6 Duplicates

The DMR and FLOW tables after Types 1-5 duplicates have been removed are joined in the Sites.R step, which can result in what is referred to as a Type 6 duplicate. In some cases, there may be an effluent flow recorded for a given NPDES, outfall, month, year, and parameter code on a different date from the nutrient concentration with the same NPDES, outfall, month, year, and parameter code. PSLoadEsT will flag the duplicate in the “dmr_remark” table and use the date of the effluent flow measurement as the measurement date for the load calculation. “OrigDate” (date field type, original date from the DMR input table) and “dateFromFLOW” (binary field type, 1 if date from FLOW table was applied) columns will be added to the “dmr_remark” table if any Type 6 duplicates are found.

5.7.5. Summary Statistics for Nutrient Concentration and Effluent Flow

The DMR and FLOW tables are combined into the “dmr_flow” output table and summary statistics (count and median) are run on the nutrient concentration and effluent flow data, and output to the “dmr_flow_stat” table. Note that field names for nutrient concentration and effluent-flow data have been changed to generic field names; “c1”, “c2”, “c3”, “q1”, and “q2” have become “conc1”, “conc 2”, “conc 3”, “quan1”, and “quan2”, respectively. The FACILITIES table is merged with the sic_codes table creating the “wi_pcs” table, which is used to store all qualitative data unique to each facility that will be merged with output tables throughout the execution of PSLoadEsT.

5.8. LoadPrep_part1.R

The LoadPrep_part1.R script reformats the dmr_flow combined table, outputs summary statistics, flags original data, generates flowclass, and flags high-flow values. See appendix 2 for a list of all output tables and descriptions.

5.8.1. Reformatting of dmr_flow Table

The dmr_flow table is reformatted such that the effluent flow data (quan1 and quan2) from the FLOW table replaces the quan1 and quan2 data from the DMR table for all nutrient concentration data (conc1, conc2, and conc3) with matching NPDES, outfall, and date. The “quarter” field for fiscal quarter is added using the R package lubridate (Grolemund and Wickham, 2011). The “pcs_data” binary field flags records with original non-missing data from the DMR or FLOW tables. The “type” text field is set equal to “original” if conc2 (average monthly concentration) is not missing; this distinguishes these data from the substitutions made in this field in section 5.9.1, and the “Expansion_Group” text field is set equal to “none” as original data do not have a State_Expansion group designation.

5.8.2. Summary Statistics Output by LoadPrep_part1.R

The “stat1” output table gives summary statistics (count and median) of the conc2 and quan1 fields. The conc2 and quan1 fields are the nutrient concentration and effluent flow fields that will be used in PSLoadEsT to calculate loads.

The “concStats” output table contains additional summary statistics on both conc2 and quan1, including counts, medians, quantiles, and means. Quantiles are given probabilities of 0.01, 0.1, 0.25, 0.5, 0.75, 0.9, and 0.99. Seasonal medians are also output to the “median1” output table.

5.8.3. “flowclass” Field

The “flowclass” numeric field is added to all output files from this point forward to classify levels of flow according to the floclassFunc() in the loadFunctions.R script (appendix 2-A), which follows table 3 below. The flowclassFunc() is based on the flow class designations used by McMahon and others (2007). The “flowstat” table outputs summary statistics (median and count) on effluent flow according to flowclass.

Table 3. Flowclass field values assigned by the flowclassFunc() in loadFunctions.R from PSLoadEsT, where "quan1" is the effluent flow field.

flowclass value	quan1 logical test (typically units of MGD)
1	$0 < \text{quan1} \leq 0.05$
2	$0.05 < \text{quan1} \leq 0.2$
3	$0.2 < \text{quan1} \leq 1$
4	$1 < \text{quan1} \leq 5$
5	$5 < \text{quan1}$
NA	quan1 missing

5.8.4. High Effluent Flow Values

High effluent flow values can result in very large nutrient loads in the final calculation, therefore, it is necessary to verify that these values are true values and not values reported in the incorrect unit or are otherwise inaccurate (McMahon and others, 2007). The "flowcheck" table flags these values so that the user can verify accuracy. The "highflow" text field assigns values according to the highFlow() function in loadFunctions.R script (table 2-1), which compares each quan1 value to the median quan1 by NPDES and outfall according to equation 1. The "flowcheck" table only contains records where highflow is not equal to "NA." All records in the "flowcheck" table should be evaluated by the user to ensure that the effluent flow values used in the load calculation are accurate.

$$\begin{aligned} highflow = \; & ifelse(x > (100 * med\_x), "100 * medflow", \\ & ifelse(x > (10 * med\_x), "10 * medflow", \\ & ifelse(x > 100, "flow > 100", NA))) \end{aligned} \qquad (1)$$

where

x is "q1" flow value (typically in MGD, from table 1);
med_x is median flow value by NDPES, outfall, and flow parameter code; and
NA is blank output.

If inaccurate effluent flow data are found, the user must fix the inaccuracies and return to the beginning of PSLoadEsT and re-import the data; program execution should be completed in order starting with the Import.R step as the user should never skip steps in program execution. It is recommended that the user always select the "flowcheck" table for output at the LoadPrep_part1.R step.

5.8.4.1. High Effluent Flow Values Example Dataset Execution

Although it is best to evaluate high flows when running real data, the user should assume when running the exampleDataset.accdb file that all high-flow values are accurate and proceed without making alterations to the exampleDataset.accdb file.

5.9. LoadPrep_part2.R

The LoadPrep_part2.R script runs through a stepped substitution of missing nutrient concentration values where an effluent flow measurement is present, counts missing values after all substitutions have been made, generates summary statistics, and flags high nutrient concentration values. This is also the step where PSLoadEsT and the methods applied by McMahon and others (2007) differ the most.

5.9.1. Missing Nutrient Concentration Data Stepped Substitution

McMahon and others (2007) use a series of substitutions beginning with a seasonal median aggregated by NPDES, outfall, year, season, and parameter code, followed by medians aggregated by region and (1) flowclass, sic_code and season, (2) flowclass and sic_code, and (3) sic_code, and ending with a substitution from the Rubin_TPC table of literature-derived concentration values.

PSLoadEsT has an added layer of complexity with the implementation of the State_Expansion table and the added "treatment_level" field. Substitutions are applied according to the order shown in table 4, which shows substitutions being applied in the same order as the substitutions applied by McMahon and others (2007) with the added treatment_level field at each level of expansion given in the State_Expansion table. Aggregating by treatment level as well as sic_code, groups similar facilities more effectively, thereby giving a more accurate median for the purposes of substitution.

Prior to running the LoadPrep_part2.R script, the user must indicate the number of nutrient concentration (conc2) values and/or number of facilities with conc2 data that must be present to generate a median value to be substituted for a missing conc2 value. Only original data are used to generate medians or determine if sufficient data exist to generate a median.

If the existing data are insufficient to generate a median substitution according to the user's selections at all levels of area expansion in the State_Expansion table, a national median (from the National_Medians table) is applied. If a national median is not available, substitutions are made from the Rubin_TPC table, however, if data are not present in the Rubin_TPC table, no load is calculated for a facility with missing nutrient concentration data.

Substitutions will continue until all missing data has been filled or all possible substitutions according to the user's selections and the steps in table 5 have been made. As substitutions are completed for each State in the LoadPrep_part2.R script, the two-letter State abbreviation and the word "complete" will print in the Rscript.exe black-box window so that the user can view the execution progress.

Table 4. Substitutions in PSLoadEsT are applied in the order below expanding the available data using in a median calculation from the original State to the national level according to Expansion_Groups given in the State_Expansion table. The "type" values and "Expansion_Group" values are output with the nutrient concentration data.

type	Expansion_group	Median aggregated by
original	none	no substitution
medsea	none	NPDES, outfall, year, season, parameter code
flowsicsea	none	state, year, flowclass, sic_code, treatment_level, season
flowsic	none	state, year, flowclass, sic_code, treatment_level
sic	none	state, year, sic_code, treatment_level
flowsicsea	SSExp1[1]	group of states designated by first Expansion_Group, year, flowclass, sic_code, treatment_level, season
flowsic	SSExp1[1]	group of states designated by first Expansion_Group, year, flowclass, sic_code, treatment_level
sic	SSExp1[1]	group of states designated by first Expansion_Group, year, sic_code, treatment_level
flowsicsea	SSExp2[1]	group of states designated by second Expansion_Group, year, flowclass, sic_code, treatment_level, season
flowsic	SSExp2[1]	group of states designated by second Expansion_Group, year, flowclass, sic_code, treatment_level
sic	SSExp2[1]	group of states designated by second Expansion_Group, year, sic_code, treatment_level
national	all	year, sic_code, treatment_level
tpc	all	sic_code

[1] Expansion_Group names are from the State_Expansion table included in the exampleDataset.accdb file. Names of expansion groups can be changed and/or additional groups added.

5.9.2. Final Dataset Checks

After all possible substitutions have been made, the final nutrient concentration and effluent flow table is summarized; existing missing values are counted, counts and medians of flow are aggregated by year and quarter, and high concentration values are flagged.

5.9.2.1. Summary Count of Remaining Missing Concentration Values

The "pcs8Missing" output table gives a count of all missing nutrient concentration values per state prior to substitutions in section 5.9.1, and the "missingALL" output table gives the count of all missing nutrient concentration values per state after all possible substitutions have been made. The "pcs9" output table has the substituted nutrient-concentration (conc2) and effluent flow (and medseaflo if applicable) values with their accompanying flags (type and Expansion_Group.)

5.9.2.2. Summary Count of Effluent Flow Data

The “mon_flow” and “qtr_flow” output tables give counts of existing effluent flow data (quan1) by NPDES, outfall, year, and month and NPDES, outfall, year, and quarter respectively. These tables are joined with pcs9 in the output table pcs10.

5.9.2.3. High Nutrient Concentration Flags

High conc2 (flagged with “1” in the highConcflag binary field) is defined as any conc2 value that is greater than 10 times the median conc2 by NPDES, outfall, and parameter code or conc2 values that are greater than the 95th percentile for the State. Median conc2 values by NPDES, outfall, and parameter code are given in the “highConc” output table, and 95th percentiles by State are given in the “percentConc” output table. The output table “pcs11” includes the highConcflag field, conc2 field values replaced with a median conc2 by NPDES, outfall, and parameter code if highConcflag equals 1, and all fields from output table “pcs10”. The “pcs11” output table has the final conc2 and quan1 values that will be used in the final load calculation.

5.10. LoadCalc.R

The LoadCalc.R script makes the final load calculations by month and by season using the pcs11 output table with the substituted nutrient concentration and effluent flow values, compares load calculations for multiple years if more than 1 year of data is present, and summarizes the concentration and flow type fields. Prior to running the LoadCalc.R script, the user must designate a load calculation formula in the PSLoadEsT interface; the default load calculation formula is shown in equation 2. Units for nutrient concentration of milligrams per liter and effluent flow of million gallons per day are required if using the default formula; if using data with different units, the formula will have to be updated by the user. PSLoadEsT determines the number of days in a month (“n” from Eq. (2)) using the days_in_month() function from the R package lubridate (Grolemund and Wickham, 2011), which takes leap year into account.

$$(c * (q * 3785000) * .000001) * n \qquad (2)$$

where

c is nutrient concentration (conc2) value in milligrams per liter,
q is effluent flow (quan1) value in million gallons per day, and
n is number of days.

5.10.1. Monthly Loads Output Tables

If 12 months of effluent flow data are present (summarized in the mon_flow output table from section 5.9.2), a monthly load calculation is made and then summed by year for each facility and can be found in the “12_months” and “load_summary_by12Month” tables, nutrient loads are saved in the kg_12 and kgmgl12_(parameter code) fields, respectively. The type and Expansion_Group fields flag load calculations based on substituted values. Annual load calculations based on 12 months of data are also found in the “load_summary_by_discharger” output table, which contains all load calculations; load calculations based on 12 months of data are flagged in the calc12_(parameter code) field and annual loads are saved in the kg_(parameter code) field.

5.10.2. Seasonal Loads Output Tables

If less than 12 months but more than three-quarters of the flow data are present in the “pcs11” output table, seasonal loads (“gt_3-4_qtrs” output table) and seasonal loads summed by year (“load_summary_bySeason” output table) are calculated. Seasonal load calculation values and seasonal loads summed by year are stored in the kg_34qtr and kg_34qtr_(parameter code) fields, respectively. Median seasonal-flow values from section 5.8.2 are used in place of quan1 in equation 2 for this calculation. Annual load calculations based on greater than three-quarters but less than 12 months of data are also found in the “load_summary_by_discharger” output table; seasonal load calculations are flagged in the calc34qtr_(parameter code) field and annual loads are saved in the kg_(parameter code) field.

5.10.3. Monthly Loads with Less Than Three-Quarters of Flow Data Output Tables

Facilities with less than three-quarters of flow data present will have monthly loads calculated and summed by year; however, annual loads are flagged as having less than three-quarters of the flow data present. The output tables for these facilities are “all_the_rest” and “load_summary_byMonth_lt34” and loads are saved in the kgqtrlt3 and kgmgllt34_(parameter code) fields, respectively. Annual loads are also output to the “load_summary_by_discharger” table with the calcqtrlt3_(parameter code) field flagged.

5.10.4. Summary of Type of Nutrient-Concentration and Effluent-Flow Data

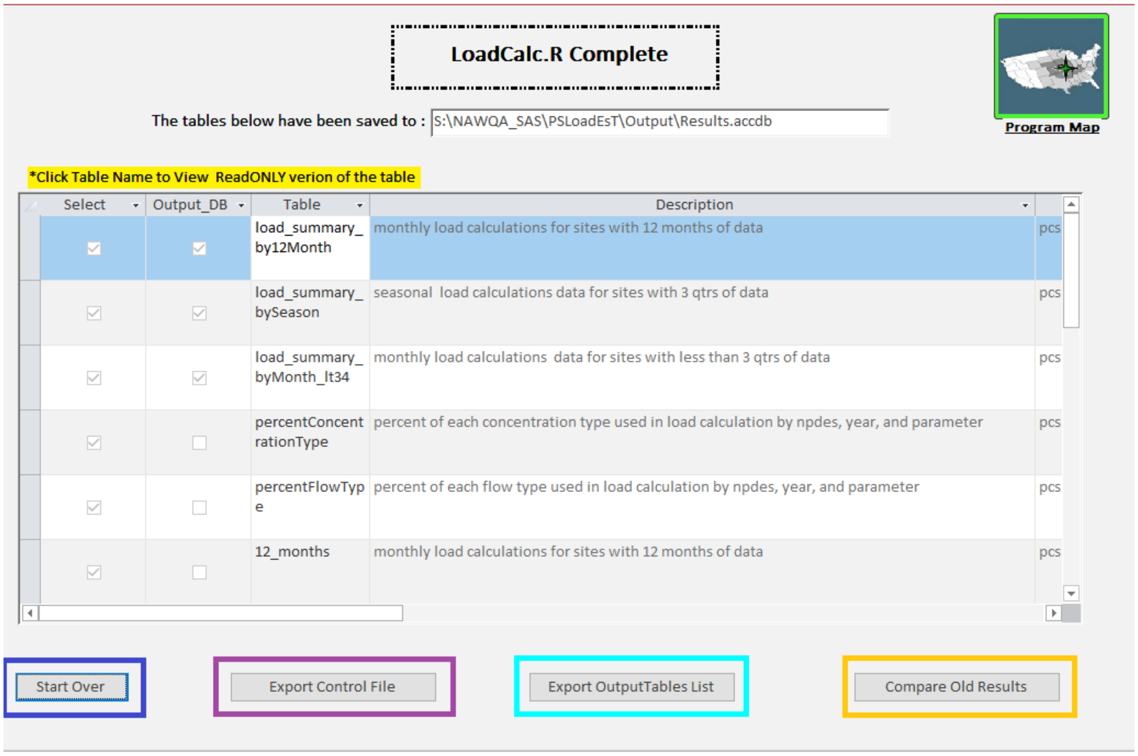
Knowing how much of each type of nutrient-concentration (conc2) and effluent-flow (quan1) data are used in the final load calculation is an important way to qualify the final load value. The “percentConcentrationType” output table summarizes the type and Expansion_Group fields for each NPDES, year, and parameter code as a percent of the total number of conc2 values. Ideally, one would want to see a large percentage of type equal to “original” and Expansion_Group equal to “none” as this is the original input data and not a substituted value; however, this is not always the largest percentage of data used to calculate the load. The “percentFlowType output table is the effluent-flow data equivalent of the “percentConcentrationType” table and gives the percentage of original flow data by NPDES, year, and parameter code.

5.10.5. Comparing Loads across Multiple Years

If more than 1 year of data is input into PSLoadEsT, the “compareYears” table will be available for output. The percent change per year is output and flagged if greater than 50 percent, and the 5th and 95th percentiles are calculated. Only 1 year of data is included in the exampleDataset.accdb file.

5.11. Post Load Calculation Output and Control File Export

From the LoadCalc_Complete screen (fig. 14), the user can export the Control File, the list of Output Tables and descriptions shown at each step in the program and compare loads to past load calculations generated by PSLoadEsT.



LoadCalc.R Complete

The tables below have been saved to : S:\NAWQA_SAS\PSLoadEsT\Output\Results.accdb

*Click Table Name to View ReadONLY verion of the table

Select	Output_DB	Table	Description	
☑	☑	load_summary_by12Month	monthly load calculations for sites with 12 months of data	pcs
☑	☑	load_summary_bySeason	seasonal load calculations data for sites with 3 qtrs of data	pcs
☑	☑	load_summary_byMonth_lt34	monthly load calculations data for sites with less than 3 qtrs of data	pcs
☑	□	percentConcentrationType	percent of each concentration type used in load calculation by npdes, year, and parameter	pcs
☑	□	percentFlowType	percent of each flow type used in load calculation by npdes, year, and parameter	pcs
☑	□	12_months	monthly load calculations for sites with 12 months of data	pcs

Figure 14. The LoadCalc.R Complete screen shows available output tables and options for post processing and program archiving. The user can output a control file (purple) with all user selections to be used as a program archiving file and/or to re-run the analysis with the current specifications. All output tables and descriptions (light blue) can be output to a .csv file. Results can be compared to results from previous years (orange). The user can also be returned to the beginning of the program to run another dataset (dark blue).

5.11.1. Exporting Control File

Exporting the Control File allows the user to execute PSLoadEsT from the same location using the same input file (possibly updated, but in the same location and with the same field names) without running through the setup screens in sections 5.1.1, 5.1.1.2, 5.2, 5.3, and 5.4, allowing the user to start PSLoadEsT execution at the Import.R step in section 5.5. However, if any changes to the user input from sections 5.1.1, 5.1.1.2, 5.2, 5.3, and 5.4 have changed with an updated input file and/or location of the PSLoadEsT or Microsoft Access program, the user must run through the setup screens in sections 5.1.1, 5.1.1.2, 5.2, 5.3, and 5.4 and not import a ControlFile.accdb file with old user selections that are no longer valid.

To export the Control File, click “Export Control File” at the bottom of the LoadCalc_Complete screen and enter a directory path in the following screen. A Microsoft Access file named ControlFile.accdb will be output to the directory specified when the ExportControl.R script is executed. Tables from the PSLoadEsT.accdb saved to the ControlFile.accdb are shown in table 5 Making alterations to the ControlFile.accdb is not recommended and could result in program failure or inaccurate load calculations.

Table 5. Tables saved to the ControlFile.accdb file when the user exports the Control File for PSLoadEsT.

Table	PSLoadEsT section No.	Description
InputTables	5.2	User-designated table names for the Required Tables (and optional LIMITS table) from section 3 in the user's Input database.
OutputTables	Appendix 3	List of PSLoadEsT Output Tables and descriptions found at each step of the program execution.
Paths	5.1.1.1 and 5.1.1.2	User-designated and auto paths to the input database, output directory, Rscript.exe file within the PSLoadEsT program, and Microsoft Access program.
TableFields	6.3	The Required Fields from table 1 and the user's field names from the Input database.
tblINfields	6.3	All field names found in the Required Tables from section 3 in the user's Input database.
tblSelectFields	6.3	Crosswalk between the Required Fields from table 1 and the user's field names from the Input database.
currentScript	Appendix 2	Current step of PSLoadEsT execution for use in interface.
LoadCalcFormula	5.10	User-designated formula for load calculation.
runScript	Appendix 2	name of R script to execute output to the Rscripts\runScript.csv file.
TablesOUTput	Appendix 3	List of all tables output by most recently run R script. If the list does not match what the user selected, the "An Error has Occurred" screen will appear with a list of tables that were not output.
TestComplete	Appendix 2	Imported from Rscripts\TestComplete.csv. Each R script will output this file if the R script runs successfully. If this file does not exist, the "An Error has Occurred" screen will appear.
Organize	5.11.2	States if the output files have been reorganized by PSLoadEsT execution step.
Parameters	5.4	Nutrient-concentration parameter codes, effluent-flow parameter code, and States selected for load calculation.

5.11.2. Exporting Output Tables List

To save the list of Output Tables and descriptions shown on each step of PSLoadEsT, click "Export OutputTables List" at the bottom of the LoadCalc.R Complete screen; the user is then asked whether the Output Tables should be organized (sorted) by step of PSLoadEsT execution, which is recommended but not required. The Organize.R script is executed and will output the list of Output Tables to the OutputTables.csv file within the user-designated output directory. Exporting the Output Tables List is recommended as a reference guide to the many output tables from PSLoadEsT.

5.11.3. Compare Old Results with Compare.R

If the user has previously calculated loads generated in PSLoadEsT, the user can compare the current load calculations to the previously calculated loads. Only un-altered output tables from PSLoadEsT with original file and field names can be used for this comparison. To execute the Compare.R script, the user must designate a path to the directory where the previously generated .csv output files are stored; files required for Compare.R are: “load_summary_by_discharger”, “load_summary_byMonth_lt34”, “load_summary_bySeason”, and “load_summary_by12Month”. Differences (new minus old) and percent differences for all numeric fields in the load_summary tables will be output by the Compare.R script.

6. Summary

Developing reproducible estimates for point-source nutrient loads throughout a given study region using nutrient-concentration and effluent-flow data reported according to permit-specific regulations requires a method that addresses all the potential data challenges including duplicate, missing, non-conformable, and erroneous data. PSLoadEsT provides a user-friendly method to estimate nutrient loads using the open-source programming language R with a Microsoft Access® interactive interface that guides the user through addressing common problems found in compiled nutrient-concentration and effluent-flow data across a study region where reporting regulations vary by permit, for use as calibration data in regional water-quality models.

7. References Cited

Dowle, M., and Srinivasan, A., 2016, data.table—Extension of 'data.frame' (R package version 1.10.0 ed.): The Comprehensive R Archive Network, accessed February 24, 2017, at https://cran.r-project.org/web/packages/data.table/index.html.

Grolemund, G., and Wickham, H., 2011, Dates and times made easy with {lubridate}: Journal of Statistical Software, v. 40, no. 3, p. 1–25.

Grothendieck, G., 2014, sqldf—Perform SQL selects on R data frames (R package version 0.4-10 ed.): accessed February 24,2017, at The Comprehensive R Archive Network, https://cran.r-project.org/web/packages/sqldf/sqldf.pdf.

Hirsch, R.M., and De Cicco, L.A., 2015, User guide to Exploration and Graphics for RivEr Trends (EGRET) and dataRetrieval—R packages for hydrologic data (version 2.0, February 2015: U.S. Geological Survey Techniques and Methods, book 4, chapter A10, 93 p., https://pubs.usgs.gov/tm/04/a10/.

Lorenz, D.L., 2015, smwrBase—An R package for managing hydrologic data (version 1.0.1): U.S. Geological Survey Open-File Report 2015-1202, 7 p., https://doi.org/10.3133/ofr20151202.

Maupin, M.A., and Ivahnenko, T., 2011, Nutrient loadings to streams of the continental United States from municipal and industrial effluent: Journal of the American Water Resources Association, v. 47, no. 5, p. 950–964, accessed August 21, 2018, at https://doi.org/10.1111/j.1752-1688.2011.00576.x.

McMahon, G., Tervelt, L., and Donehoo, W., 2007, Methods for estimating annual wasterwater nutrient loads in the southeastern United States: U.S. Geological Survey Open-File Report 2007-1040, p. 81.

R Core Team, 2016, R—A language and environment for statistical computing (R package version 3.3.0 ed.): Vienna, Austria, R Foundation for Statistical Computing.

Ripley, B., and Lapsley, M., 2016, RODBC: ODBC Database Access (R package version 1.3-14 ed.), accessed February 24, 2017, at https://CRAN.R-project.org/package=RODBC.

Schwarz, G.E., Hoos, A.B., Alexander, R.B., and Smith, R.A., 2006, The SPARROW surface water-quality model—Theory, application, and user documentation: U.S. Geological Survey Techniques and Methods, book 6, chap. B3, https://pubs.usgs.gov/tm/2006/tm6b3/.

Skinner, K.D., and Maupin, M.A., 2018, Point-source nutrient loads to streams of the conterminous United States, 2012: U.S. Geological Survey Data Series 1101, 13 p., https://doi.org/10.3133/ds1101.

U.S. Environmental Protection Agency, 1990, Permit compliance system: U.S. Environmental Protection Agency, 20W-4001, 21 p., accessed November 30, 2018, at https://nepis.epa.gov/Exe/ZyPDF.cgi?Dockey=910167QT.PDF.

Walker, A., 2017, openxlsx—Read, write and edit XLSX Files (R package version 4.0.0. ed.): Comprehensive R Archive Network, accessed February 24, 2017, at https://cran.r-project.org/web/packages/openxlsx/index.html.

Wickham, H., 2007, Reshaping data with the {reshape} package: Journal of Statistical Software, v. 21, no. 12, p. 1–20.

Wickham, H., 2011, The split-apply-combine strategy for data analysis: Journal of Statistical Software, v. 40, no. 1, p. 1–29.

Wickham, H., 2015, pryr—Tools for computing on the language (R package version 0.1.2 ed.): Comprehensive R Archive Network, accessed February 24, 2017, at https://cran.r-project.org/web/packages/pryr/index.html.

Wickham, H., 2016a, httr—Tools for working with urls and http (R package version 1.2.1 ed.): Comprehensive R Archive Network, accessed February 24, 2017, at https://cran.r-project.org/web/packages/httr/index.html.

Wickham, H., 2016b, stringr—Simple, consistent wrappers for common string operations (R package version 1.1.0 ed.): Comprehensive R Archive Network, accessed February 24, 2017, at https://cran.r-project.org/web/packages/stringr/index.html.

Wickham, H., and Francois, R., 2016, dplyr—A grammar of data manipulation (0.5.0 ed.): The Comprehensive R Archive Network, accessed February 24, 2017, at https://cran.r-project.org/web/packages/dplyr/index.html.

8. Appendixes

Appendix 1. PSLoadEsT Directory Structure

PSLoadEsT directory structure within the PSLoadEsT parent directory. Directory structure of PSLoadEsT should not be changed. Files within the directories should not be deleted or altered, with the exception of the Output directory.

Directory	Contents
exampleDataset	exampleDataset.accdb file and the Type 4-5 duplicates selected to be used in the exampleDataset.accdb DMR table stored in the DMR_DuplicatesKept.csv (for quick import of duplicate selections)
Output	an optional output directory to store results. All output tables from appendix 1 for the exampleDataset.accdb run are stored in this directory.
R-3.3.0	included approved version of R (R Core Team, 2016) for use in PSLoadEsT with all required libraries installed
Rscripts	PSLoadEsT R scripts, .RData files used to load workspace from previous step with exampleDataset.accdb run stored, temporary .csv files used to track PSLoadEsT progress, .accdb files used to trigger the interface (Bounce.accdb), test size of output tables prior to output to Results.accdb (sizeTest.accdb), and generate the ControlFile.accdb file (Blank.accdb).

Appendix 2. Structure of PSLoadEsT Execution

A number of internal steps are executed at each major step of PSLoadEsT execution beginning with the first R script executed, the Fields.R script. The process detailed here is the same for all program steps where an R script is executed (i.e. when the black Rscript.exe window, fig. 6, appears).

A. Loading Functions and Sourcing the R script for Execution

First the name of the script to execute is saved to the ‘runScript” table in the PSLoadEsT, which is then exported to the runScript.csv file stored in the PSLoadEsT\Rscripts\ folder. Then the runScript.R script is executed via shell command using the 64-bit Rscript.exe executable stored in the R-3.3.0 directory of PSLoadEsT, which opens the black box (Rscript.exe window) visible to the user that should never be manually closed by the user during script execution.

The runScript.R script locates the PSLoadEsT\Rscripts directory from which it is being executed and then executes the loadFunctions.R script in the same directory, which loads all custom functions and common vectors and variables necessary for successful execution of PSLoadEsT. Custom functions and descriptions are shown in table 2-1. The loadFunctions.R script also runs the openPackages.R script, which initializes all required R packages used by PSLoadEsT; required R packages are given in appendix 3.

Table 2-1. Custom functions found in the loadFunctions.R script.

Function	Description
removeDFs	Removes objects from the active R environment to free memory available for further execution
outRemove	Outputs user selected tables to .csv files (and the Results.accdb file as designated by the user.) The size of the output table and the space available in the Results.accdb file are tested, if space exists in the Results.accdb the output table is saved to that file otherwise the table will be saved only in .csv format. After all tables are output, the objects are removed from the R workspace to free memory for additional operations.
convertClass	Converts columns from the original class to the class specified in the "cls" argument
remarkFunc	Generates remark codes for all non-numeric input in columns that should be numeric. Remark codes are as follows : (1) "<", (2) ">", (3) non-numeric or negative, (0) numeric, non-missing, and (NA) missing
valueFunc	This strips text ('<', '>') out of numeric fields after remark codes have been saved. Negative values become NA.
niceFuncs	R function calls are given plain English names (i.e. "length(na.omit(x))" becomes "count")
getStats	Aggregates data by "group" argument and applies functions from "funcs" argument. Typically "funcs" are from the common vectors and variables section of the loadFunctions.R script (statFuncs, quantiles, and moreFuncs)
floclassFunc	Generates flowclass variable according to table 3.
highFlow	High effluent flow values are flagged as follows : "100*medflow" if flow is 100 times the median, "10*medflow" if flow is 10 times the median, "flow>100MGD" if flow>100, and "NA" if none of the previous conditions are met.
replaceType	Replaces missing values with substitutions and flags the substitutions in the "type" and "Expansion_Group" columns
countMissing	Counts number of missing values aggregated by state and parameter
seasonDays	Calculates number of days in each season taking leap year into account

Once all functions and packages have been loaded, PSLoadEsT searches for data from previously completed steps found in the .RData files stored in the PSLoadEsT\Rscripts directory and loads all available data into the workspace. All .csv files within the PSLoadEsT\Rscripts directory are then deleted, as well as any output .csv files that will be regenerated in the current step. The .csv files in the PSLoadEsT\Rscripts directory are primarily used to test for code completion allowing the interface to alert the user of incomplete execution due to errors.

The runScript.R script then runs the R script for the current step within a try() function that catches errors in the code execution to be output to the user interface.

B. Successful Execution of Current Step

Input and Output in 32-bit

Each step of PSLoadEsT execution is completed in the 64-bit Rscript.exe executable, however, since the interface and the users input dataset are both stored in 32-bit Microsoft Access .accdb files the first section of each R script pulls the data from the Microsoft Access files by using a system command to run the PSLoadEsT\Rscripts\in32bit.R script using the 32-bit Rscript.exe executable included in the program. Tables and query strings to pass to the in32bit.R R script are stored in the PSLoadEsT\Rscripts\temp.RData file. The in32bit.R R script then saves the additional user input tables as R objects within the same temp.RData file to be loaded into the 64-bit R environment in the execution of the current step. Similarly, any output tables saved to the Results.accdb file are saved using the PSLoadEsT\Rscripts\out32bit.R script.

Output Files

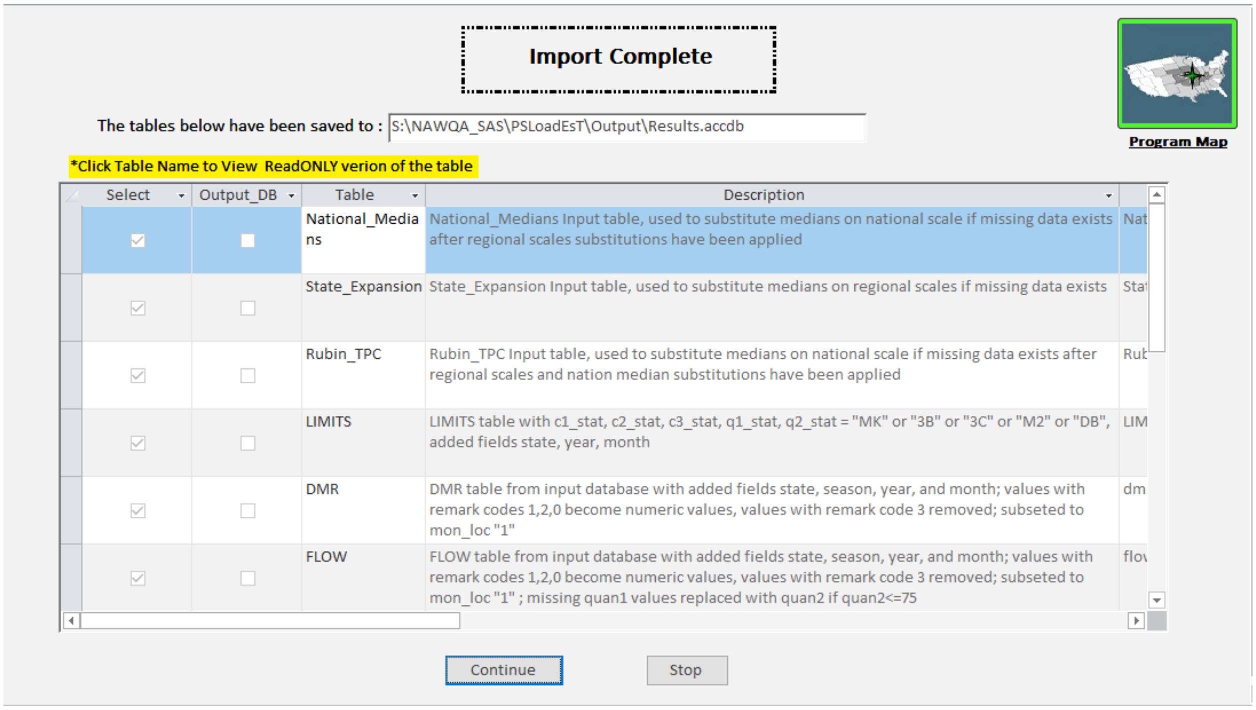
The two methods of output for PSLoadEsT are .csv files for each table and a single Results.accdb Microsoft Access file containing all tables designated by the user to output to a results database. All tables selected for output will automatically be output to .csv files with the name shown in the “Table” field in the user interface, and only those tables selected under “Output_DB” will be output to a Results.accdb file in the user’s selected output directory. Since 32-bit Microsoft Access files have a size limit of 2GB, the tables selected for output to the Results.accdb file will be saved to the Results.accdb file until the size limit has been reached, after which time all files will only be output in .csv format. The PSLoadEsT\Rscripts\sizeTest.accdb file is used to determine whether output tables can be saved to the Results.accdb file. Due to the size constraint, it is recommended that the user select “Output_DB” only for tables that will be manipulated within the Results.accdb file using Microsoft Access tools.

From R to PSLoadEsT Interface

Once all operations are complete, the TestComplete.csv file will be generated indicating that the R script was executed without errors, and a (nextstepname).RData file is saved to the PSLoadEsT\Rscripts\ directory. A list of all tables successfully output to the user’s output directory is saved to the TablesOUTput.csv file within the PSLoadEsT\Rscripts\ directory, so the PSLoadEsT interface can compare this list to the requested output tables in the event of an error. Finally, the Bounce.accdb file is opened via shell.exec(). Clicking the “Continue” button within the Bounce.accdb file returns the user to the PSLoadEsT interface.

Successful Execution Interface Output Screen

Assuming successful execution of the entire R script for the current step, the user will see a screen with a list of output tables saved at the current step (example from Import.R step shown in figure 2-1). The user can then click on any output table name (the “Table” field) to view a temporary version of the table within the interface; this may take a few seconds to complete as many tables are large. Leading zeros in the “outfall” and “parameter” fields may not appear in the temporary tables shown in the interface, but are present in the .csv files in the user’s output directory, however, many programs (i.e., Microsoft Excel®) will strip the preceding zeros from the “outfall” and “parameter” fields so the user should be mindful of this effect when viewing output files. All output files will read correctly (with preceding zeros) in a text editor or when imported into R.



Import Complete

The tables below have been saved to : S:\NAWQA_SAS\PSLoadEsT\Output\Results.accdb

*Click Table Name to View ReadONLY verion of the table

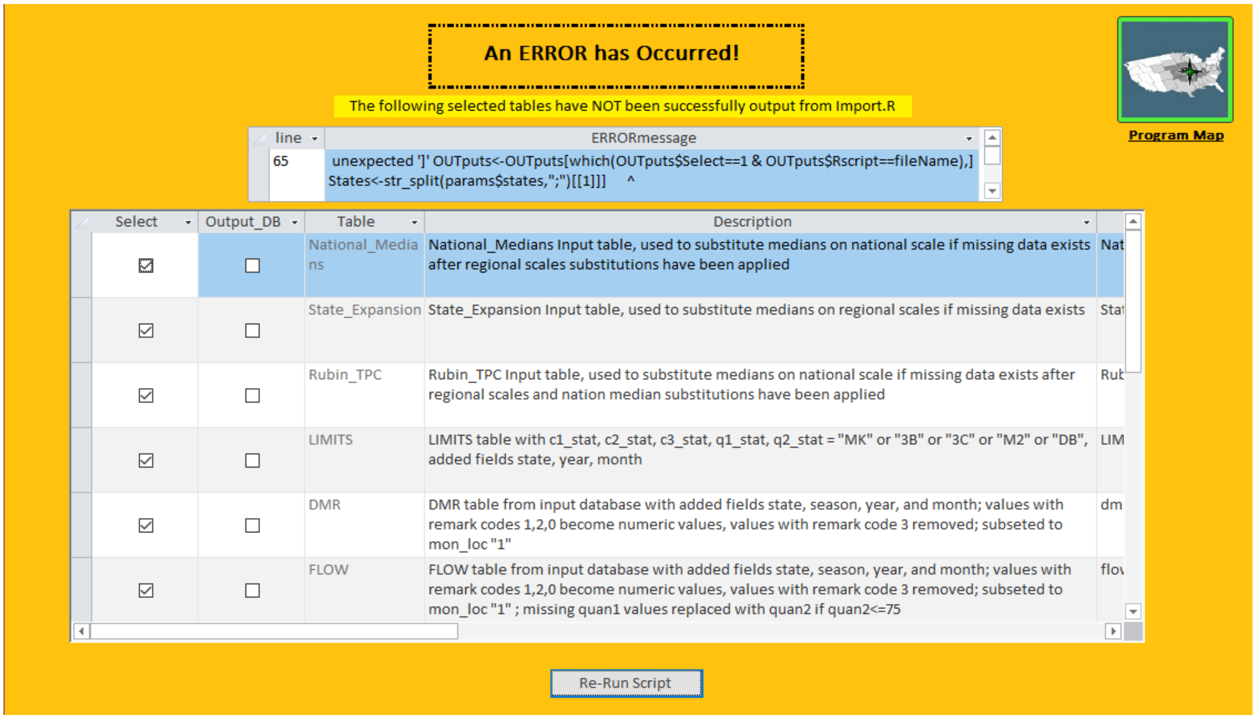
Select	Output_DB	Table	Description	
☑	□	National_Medians	National_Medians Input table, used to substitute medians on national scale if missing data exists after regional scales substitutions have been applied	Nat
☑	□	State_Expansion	State_Expansion Input table, used to substitute medians on regional scales if missing data exists	Stat
☑	□	Rubin_TPC	Rubin_TPC Input table, used to substitute medians on national scale if missing data exists after regional scales and nation median substitutions have been applied	Rub
☑	□	LIMITS	LIMITS table with c1_stat, c2_stat, c3_stat, q1_stat, q2_stat = "MK" or "3B" or "3C" or "M2" or "DB", added fields state, year, month	LIM
☑	□	DMR	DMR table from input database with added fields state, season, year, and month; values with remark codes 1,2,0 become numeric values, values with remark code 3 removed; subseted to mon_loc "1"	dm
☑	□	FLOW	FLOW table from input database with added fields state, season, year, and month; values with remark codes 1,2,0 become numeric values, values with remark code 3 removed; subseted to mon_loc "1" ; missing quan1 values replaced with quan2 if quan2<=75	flov

Figure 2-1. Example of .R script Complete output screen. Path to the output database is shown at the top, but all files selected for output are also output to the same directory as the .csv files. Clicking on the table name will open a temporary read-only version of the table within the PSLoadEsT interface. Clicking “Continue” at the bottom will trigger the next step. To return to the Program MAP, click the Program MAP icon in the upper right corner.

To continue to the next step in the program simply click “Continue” below the list of output tables. If the PSLoadEsT program is closed, upon re-opening it will automatically display the same screen that was last open allowing the user to pick up execution at the next step without returning to the Program MAP.

C. An ERROR Has Occurred

The PSLoadEsT interface will display “An ERROR has Occurred” screen (fig. 2-2) in the event of an R script failing to fully execute. The error message, along with the line number that resulted in the error, is displayed along with a list of tables that failed to save to the users output directory. Errors can occur for many reasons, the most common being incorrect user input in the PSLoadEsT interface and improperly formatted input data. In the event of an error, the user should double check all interface input from the previous steps as well as formatting of the input datasets.



An ERROR has Occurred!

The following selected tables have NOT been successfully output from Import.R

line	ERRORmessage
65	unexpected ']' OUTputs<-OUTputs[which(OUTputs$Select==1 & OUTputs$Rscript==fileName),] States<-str_split(params$states,";")[[1]]] ^

Select	Output_DB	Table	Description	
☑	☐	National_Medians	National_Medians Input table, used to substitute medians on national scale if missing data exists after regional scales substitutions have been applied	Nat
☑	☐	State_Expansion	State_Expansion Input table, used to substitute medians on regional scales if missing data exists	Stat
☑	☐	Rubin_TPC	Rubin_TPC Input table, used to substitute medians on national scale if missing data exists after regional scales and nation median substitutions have been applied	Rub
☑	☐	LIMITS	LIMITS table with c1_stat, c2_stat, c3_stat, q1_stat, q2_stat = "MK" or "3B" or "3C" or "M2" or "DB", added fields state, year, month	LIM
☑	☐	DMR	DMR table from input database with added fields state, season, year, and month; values with remark codes 1,2,0 become numeric values, values with remark code 3 removed; subseted to mon_loc "1"	dm
☑	☐	FLOW	FLOW table from input database with added fields state, season, year, and month; values with remark codes 1,2,0 become numeric values, values with remark code 3 removed; subseted to mon_loc "1" ; missing quan1 values replaced with quan2 if quan2<=75	flov

Figure 2-2. The Error has Occurred screen displays the line number associated with the execution error as well as the tables that were not output as a result of the error.

If a mistake in the interface input is found, the user should return to the step at which the mistake was made and re-run PSLoadEsT from that point. Do NOT correct the mistake and use the Program MAP to skip ahead to a subsequent step as the corrected interface selections are only saved in the .RData files after that step is executed. These .RData files are used by the R scripts to access user interface input, therefore, if the step in which the interface input was corrected is not run, the .RData file accessed by subsequent steps will still contain the previously saved incorrect interface input leaving the mistake in place. If an error is a result of incorrectly formatted user input data, the user must correct the input data file and return to the beginning of the program.

Appendix 3. Output Tables and Descriptions by R script

Output Table Name	Description	Input Table(s)	R script	R script Documentation Section
DMR	DMR table from input database with added fields state, season, year, and month; subset where mon loc = "1"	DMR_input	Import.R	6.5
FACILITIES	FACILITIES table from input database, fields renamed using PSLoadEsT required field names	FACILITY_input	Import.R	6.5
FLOW	FLOW table from input database with added fields state, season, year, and month; subset where mon_loc = "1"	FLOW_input	Import.R	6.5
LIMITS	LIMITS table with c1_stat, c2_stat, c3_stat, q1_stat, q2_stat = "MK" or "3B" or "3C" or "M2" or "DB", added fields state, year, month	LIMITS_input	Import.R	6.5
National_Medians	National_Medians Input table	National_Medians_	Import.R	6.5
Rubin_TPC	Rubin_TPC Input table	Rubin_TPC_input	Import.R	6.5
sic_codes	sic_codes table from input, fields renamed to show PSLoadEsT required field names	sic_codes_input	Import.R	6.5
State_Expansion	State_Expansion Input table	State_Expansion	Import.R	6.5
DMR_DuplicatesKept	Type 4 and 5 duplicates where more than one conc2 value exists in DMR table, user must select duplicates for removal	DMR	DataTest.R	6.6
DMR_DuplicatesRemoved	Type 1, 2 and 3 duplicates removed from the DMR table. After user's selection of Type4-5 duplicates this table will be updated by the DupFix.R script in the Sites.R step	DMR	DataTest.R	6.6
FACILITIES_DuplicatesKept	Type 4 duplicate where there are more than one value for all fields, user must select duplicates for removal	FACILITIES	DataTest.R	6.6
FACILITIES_DuplicatesRemoved	Type 1 duplicates removed from the FACILITIES table	FACILITIES	DataTest.R	6.6

Output Table Name	Description	Input Table(s)	R script	R script Documentation Section
FACILITIES_JoinTest	shows facilities found in either the DMR, FLOW, (or LIMITS table) that do not exist in the FACILITIES table. If these are not added to the FACILITIES table they will be removed from the analysis.	DMR,FLOW,LIMITS,FACILITIES	DataTest.R	6.6
FLOW_DuplicatesKept	Type 4 and 5 duplicates where more than one quan1 value exists in FLOW table, user must select duplicates for removal	FLOW	DataTest.R	6.6
FLOW_DuplicatesRemoved	Type 1, 2, and 3 duplicates removed from the FLOW table	FLOW	DataTest.R	6.6
LIMITS_DuplicatesKept	Type 4 duplicate where this is more than one value for c2_stat and q1_stat, user must select duplicates for removal	LIMITS	DataTest.R	6.6
LIMITS_DuplicatesRemoved	Type 1, 2, and 3 duplicates removed from the LIMITS table	LIMITS	DataTest.R	6.6
LIMITS_JoinTest	shows ndpes, outfall, parameter combinations found in either the DMR or FLOW table that do not exist in the LIMITS table. This has no consequence on further analysis completed within the program	DMR,FLOW,LIMITS	DataTest.R	6.6
National_Medians_DuplicatesKept	Type 4 duplicate where there is more than one value for concentration, user must select duplicates for removal	National_Medians	DataTest.R	6.6
National_Medians_DuplicatesRemoved	Type 1, 2 duplicates removed from the National_Medians table	National_Medians	DataTest.R	6.6
National_Medians_JoinTest	shows sic_code/treatment_level combinations found in the FACILITIES table that do not exist in the National_Medians table. If these are not added to the National_Medians table no substitutions can be made at the national level for the sic_code/treatment_levels shown	FACILITIES, National_Medians	DataTest.R	6.6
Rubin_TPC_DuplicatesKept	Type 4 duplicate where there is more than one value for concentration, user must select duplicates for removal	Rubin_TPC	DataTest.R	6.6
Rubin_TPC_DuplicatesRemoved	Type 1, 2, and 3 duplicates removed from the Rubin_TPC table	Rubin_TPC	DataTest.R	6.6

Output Table Name	Description	Input Table(s)	R script	R script Documentation Section
sic_codes_DuplicatesKept	Type 4 duplicate where there is more than one value for sic_code, user must select duplicates for removal	sic_codes	DataTest.R	6.6
sic_codes_DuplicatesRemoved	Type 1 duplicates removed from the sic_codes table	sic_codes	DataTest.R	6.6
sic_codes_JoinTest	shows sic_codes found in the either the FACILITIES, National_Medians, or Rubin_TPC tables that do not exist in the sic_codes table. If these are not added to the sic_codes table, they will be removed from further	FACILITIES, National_Medians,Rubin_TPC,sic_codes	DataTest.R	6.6
State_Expansion_DuplicatesRemoved	Type 1 duplicates removed from the State_Expansion table	State_Expansion	DataTest.R	6.6
states_JoinTest	shows states found in the either the DMR, FLOW, or FACILITIES tables that do not exist in the State_Expansion table. If these are not added to the State_Expansion table, no substitutions can be made at the regional level.	DMR,FLOW,FACILITIES, State_Expansion	DataTest.R	6.6
dmr_flow	DMR and FLOW tables merged	DMR, FLOW	Sites.R	6.7
dmr_flow_stat	medians and counts of conc1,	dmr_flow_stat	Sites.R	6.7
dmr_remark	DMR table from input database with added fields state, season, year, and month; original values and remark codes; conc1, conc2, conc3, quan1, quan2 are fields carried on through the program. Duplicates are flagged if applicable	DMR_input	Sites.R	6.7
facilities_remark	FACILITIES table from input database with duplicates flagged if applicable	FACILITIES_input	Sites.R	6.7
flow_remark	FLOW table from input database with added fields state, season, year, and month; original values and remark codes; conc1, conc2, conc3, quan1, quan2 are fields carried on through the program. Duplicates are flagged if applicable	FLOW_input	Sites.R	6.7
limits_remark	LIMITS table from input database with duplicates flagged if applicable	LIMITS_input	Sites.R	6.7

Output Table Name	Description	Input Table(s)	R script	R script Documentation Section
national_medians_remark	National_Medians table from input database with duplicates flagged if applicable	National_Medians_input	Sites.R	6.7
rubin_tpc_remark	rubin_tpc table from input database with duplicates flagged if applicable	Rubin_TPC_input	Sites.R	6.7
sic_codes_remark	sic_codes table from input database with duplicates flagged if applicable	sic_codes_input	Sites.R	6.7
state_expansion_remark	State_Expansion table from input database with duplicates flagged if applicable	State_Expansion_input	Sites.R	6.7
wi_pcs	FACILITIES and sic_codes tables merged by sic_code, subset by npdes<=9 characters	FACILITIES, sic_codes	Sites.R	6.7
concStats	medians of conc2 and quan1, by npdes, outfall, year, season, parameter	pcs7	LoadPrep_part1.R	6.8
flowcheck	highflow variable added according to highFlow(), subset where quan1 >1 and highflow is not null	temp1, tempflow	LoadPrep_part1.R	6.8
flowstat	medians and counts for flow=quan1 by state and flowclass	pcs5	LoadPrep_part1.R	6.8
pcs1	dmr_flow table with added quarter(year) field with months 1-3 given quarter = 1, join parameter in dmr_pcodes to parameter in flow_pcodes, replace quan1 and quan2 from DMR with data from FLOW	DMR, FLOW	LoadPrep_part1.R	6.8
pcs2	if data exists for a month, set pcs_data==1	pcs1	LoadPrep_part1.R	6.8
pcs4	pcs2 and wi_pcs tables merged by npdes	pcs2	LoadPrep_part1.R	6.8
pcs5	add flowclass variable according to function floclassFunc()	pcs4	LoadPrep_part1.R	6.8
pcs7	if conc2 non-missing, type="original"	pcs5	LoadPrep_part1.R	6.8
stat1	medians and counts of conc1, conc2, conc3, quan1, quan2, by "npdes","outfall","year","quarter","month","parameter" from pcs1	pcs1	LoadPrep_part1.R	6.8
temp1	pcs7 with quan1>0	pcs7	LoadPrep_	6.8
tempflow	median of quan1, by npdes, outfall, parameter	tempflow	LoadPrep_part1.R	6.8

Output Table Name	Description	Input Table(s)	R script	R script Documentation Section
flowsic	medians and counts of conc2, by flowclass, sic_code, and parameter, subset to median(conc2)>0 and count(conc2)>5	pcs7	LoadPrep_ part2.R	6.9
flowsicsea	medians and counts of conc2, by flowclass, sic_code, season, and parameter, subset to median(conc2)>0 and count(conc2)>5	pcs7	LoadPrep_ part2.R	6.9
flowsub	subset of pcs8 where seasonal median flow values where substituted	pcs8	LoadPrep_ part2.R	6.9
highConc	median concentration values by npdes, outfall, parameter	pcs10	LoadPrep_ part2.R	6.9
missingALL	count of missing concentration values after all substitutions	pcs9	LoadPrep_ part2.R	6.9
mon_flow	count of flow values per npdes, outfall, year, parameter for use in LoadCalc.R	pcs9	LoadPrep_ part2.R	6.9
pcs10	pcs9 joined with mon_flow and qtrstat	pcs9, mon_flow, qtrstat	LoadPrep_ part2.R	6.9
pcs11	Final data for load calculation, pcs10 joined with highConc and percentConc, if concentration > 10*median(concentration) and >95th percentile for the state then replace with median by npdes, outfall, and parameter	pcs10, highConc, percentConc	LoadPrep_ part2.R	6.9
pcs8	seasonal medians by npdes, outfall, year, season, parameter replacing missing concentration data if count>5, median season flow values also calculated and used to replace missing flow values	pcs7, median1	LoadPrep_ part2.R	6.9
pcs8Missing	count of missing concentration values in pcs8	pcs8	LoadPrep_ part2.R	6.9
pcs9	pcs8 with missing concentration values replaced with flowsicsea, flowsic, sic, and tpc values utilizing state_expansion if applicable	pcs8, flowsicsea, flowsic, sic, Rubin_TPC, State_Expansion, sic_codes	LoadPrep_ part2.R	6.9
percentConc	95th percentile of concentration values by state and parameter	pcs10	LoadPrep_ part2.R	6.9
qtrstat	number of quarterly flow values by npdes, outfall, year, parameter	pcs9	LoadPrep_ part2.R	6.9

Output Table Name	Description	Input Table(s)	R script	R script Documentation Section
sic	medians and counts of conc2, by sic_code, and parameter, subsetted to median(conc2)>0 and count(conc2)>5	pcs7	LoadPrep_part2.R	6.9
12_months	monthly load calculations for sites with 12 months of data	pcs11	LoadCalc.R	6.10
all_the_rest	monthly load calculations for sites with data for < 3/4 of the year	pcs11	LoadCalc.R	6.10
compareYears	Comparison of loads from year to year, flagging large changes, this table is only an option if analyzing more than 1 year of data.	load	LoadCalc.R	6.10
gt_3-4_qtrs	seasonal load calculations for sites with data for >=3/4 of the year but less than 12 months	pcs11	LoadCalc.R	6.10
load_summary_by_discharger	Combine all types of load calculations into 1 column "kg", flag type of load calculation in calc_(type) fields	temp7	LoadCalc.R	6.10
load_summary_by12Month	monthly load calculations for sites with 12 months of data	pcs11	LoadCalc.R	6.10
load_summary_byMonth_lt34	monthly load calculations data for sites with less than 3/4 of the year data	pcs11	LoadCalc.R	6.10
load_summary_bySeason	seasonal load calculations data for sites with 3/4 of the year of data	pcs11	LoadCalc.R	6.10
percentConcentrationType	percent of each concentration type used in load calculation by npdes, year, and parameter	pcs11	LoadCalc.R	6.10
percentFlowType	percent of each flow type used in load calculation by npdes, year, and parameter	pcs11	LoadCalc.R	6.10
temp7	merge all load calculations	temp1, temp4,	LoadCalc.	6.10
compare_load_summary_by_discharger	compares the most recent year of the current load_summary_by_discharger table to a previous output from the PSLoadEsT	load_summary_by_discharger	Compare.R	6.11.3
compare_load_summary_by12Month	compares the most recent year of the current load_summary_by12Month table to a previous output from the PSLoadEsT	load_summary_by12Month	Compare.R	6.11.3

Output Table Name	Description	Input Table(s)	R script	R script Documentation Section
compare_load_summary_byMonth_lt34	compares the most recent year of the current load_summary_byMonth_lt34 table to a previous output from the PSLoadEsT	load_summary_byMonth_lt34	Compare.R	6.11.3
compare_load_summary_bySeason	compares the most recent year of the current load_summary_bySeason table to a previous output from the PSLoadEsT	load_summary_bySeason	Compare.R	6.11.3

Appendix 4. Information on R-Packages used in PSLoadEsT

The following R-packages are included with R version 3.3.0 (R Core Team, 2016) in the ~PSLoadEsT\ R-3.3.0\library directory. The “core packages” are directly used in the PSLoadEsT R scripts and are called in the openPackages.R script. All dependencies of the “core packages” and their dependent packages are also included in the ~PSLoadEsT\ R-3.3.0\library directory.

Package	Version	Reference
dataRetrieval	2.6.3	(Hirsch and De Cicco, 2015)
data.table	1.10.0	(Dowle and Srinivasan, 2016)
dplyr	0.5.0	(Wickham and Francois, 2016)
httr	1.2.1	(Wickham, 2016a)
lubridate	1.6.0	(Grolemund and Wickham, 2011)
openxlsx	4.0.0	(Walker, 2017)
plyr	1.8.4	(Wickham, 2011)
pryr	0.1.2	(Wickham, 2015)
reshape2	1.4.2	(Wickham, 2007)
RODBC	1.3-14	(Ripley and Lapsley, 2016)
smwrBase	1.1.2	(Lorenz, 2015)
sqldf	0.4-10	(Grothendieck, 2014)
stringr	1.1.0	(Wickham, 2016b)
xml2	1.1.0	(Wickham, 2017)

Publishing support provided by the U.S. Geological Survey
Science Publishing Network, Tacoma Publishing Service Center

For more information concerning the research in this report, contact the
Director, MD-DE-DC Water Science Center
U.S. Geological Survey
5522 Research Park Drive
Catonsville, MD 21228
https://www.usgs.gov/centers/md-de-dc-water

Gorman Sanisaca and others—**Annual Wastewater Nutrient Data Preparation and Load Estimation Using the Point-Source Load Estimation Tool**—Open-File Report 2019–1025

ISSN 2331-1258 (online)
https://doi.org/10.3133/ofr20191025