



Standard Operating Procedure 2.3.05

Coastal Wetland Elevation Data QA/QC and Data Publication—Version 1.0

Please cite this as:

Wright, W., D. Skidds, E. Cheng, and A. Kozlowski. 2023. Coastal Wetland Elevation Data QA/QC and Data Publication—Version 1.0. Inventory and Monitoring Division Standard Operating Procedure NPS/IMD/SOP—2.3.05. National Park Service Inventory and Monitoring Division. Fort Collins, Colorado.

Summary

This standard operating procedure (SOP) provides instructions for quality assurance (QA), quality control (QC), and distribution of monitoring data associated with the protocol for wetland elevation monitoring. Additionally, guidance is provided for use of the QA/QC dashboard in R, loading code from GitHub, and generating metadata using Ecological Metadata Language (EML). All wetland elevation monitoring data are stored in the multi-network surface elevation table data management system managed and maintained by the National Park Service (NPS). The data management system is utilized by five NPS Inventory and Monitoring Networks along the Atlantic seaboard: the Northeast Coastal and Barrier Network (NCBN), the Northeast Temperate Network (NETN), the National Capital Region Network (NCRN), the Southeast Coast Network (SECN), and the South Florida/Caribbean Network (SFCN). The data system encompasses data from 20 national park units (Table 1). It is the responsibility of the network data managers and wetland elevation (Surface Elevation Table or SET) protocol leads and staff to be thoroughly familiar with the standards and procedures within the protocol and companion SOPs, and ensure this SOP is updated to reflect any substantial changes made to that plan. All proposed changes to this SOP must be routed through the appropriate data managers for approval.

Revision Log

Revision Date	Author	Changes Made	Reason for Change	New Version #

Contents

	Page
Summary.....	1
Revision Log.....	2
Overview.....	5
Expected Audience	5
Data Stewardship and Data Workflow.....	6
Roles and Responsibilities.....	8
Exporting Data.....	9
Granting Permission to Export Data via the User Interface	9
Data Export Option 1: Using the Standalone R Script.....	10
Data Export Option 2: Using the SET-Viz Dashboard	10
Analytical Data Views.....	11
Metadata Views.....	11
Data QA/QC Dashboard	13
Overview	13
Installation and Use Requirements.....	13
Install and Configure Git.....	14
Clone the SET-viz Project to your Local Computer	14
Open the Dashboard	16
Connect the Dashboard to SET Data.....	17
Update the QA/QC Dashboard.....	17
Suggest Changes to the Dashboard	19
Collaborations and Notifications.....	20
References & Tutorials for Git.....	20
Data Package Publication Preparation	21
Create Draft DataStore Records	21
Create a Draft DataStore Record for the Data Package.....	21

Create a Draft DataStore Record for the Metadata Creation Package	22
Write EML Metadata	22
Getting Started.....	22
Install Required Applications	22
Make Sure R is Current	23
Collect Previous Files from DataStore	23
Create and Edit EML Metadata Using NPS R Scripts	24
Before Running the NPS EML Creation and NPS EML Editor R Scripts.....	24
The EMLassemblyline Metadata Templates	24
Run the NPS EML Assemblyline Script	27
Overview of the NPS EML Editor Script.....	29
Run the NPS EMLeditor Script.....	29
Compile the Metadata Creation Package	30
Publish the Data Package on DataStore.....	31
Upload Data Package to DataStore	31
Upload the Metadata Creation Package to DataStore	31
Literature Cited	32

Overview

Surface Elevation Table (SET) monitoring data are stored and processed using a multi-network, multi-region data management system. This system consists of three applications: a data entry user interface, a data summary and visualization dashboard to assist with QC, and a back end data storage and reporting system. The data entry application is designed for the manual entry of field data and application of event-level QA/QC processes. It consists of a web-based user interface located on the NPS Integrated Resource Management Applications site ([IRMA](#)). The QA/QC Dashboard is an R-based application that, when connected to a provisional data set, can be used to provide graphical overviews to identify data errors or anomalies. The back end storage application consists of an SQL Server database that manages all data storage and reporting required for both internal and external use. All three applications (the front end, back end, and dashboard) can be accessed by multiple users simultaneously.

The purpose of this SOP is to provide an instructional overview of the processes used to export data from the database, import data into the QA/QC dashboard, apply standards to the data, move data to an ‘accepted’ status, and export and publish a data package for publication in the [NPS DataStore](#). Technical documentation and instruction for use of the web-based data entry interface is provided in SOP 2.3.06 *Coastal Wetland Elevation Database User Manual and Technical Documentation—Version 2.0* (Wright et al. 2023). Technical documentation for the SQL back end database is available in SOP 2.3.07 *Monitoring Coastal Wetland Elevation Data Management and Database Technical Documentation—Version 2.1* (Wright et al. 2022a). Links to documentation relevant to the SET database system are consolidated in a [SET Project](#) (NPS 2023) on the DataStore.

Expected Audience

This SOP is written specifically for data managers, data interns, protocol leads, project crew members, and cooperating partners who wish to access existing data to support implementation of this protocol. This SOP is intended to reduce the overall time needed to accomplish required tasks. To accomplish this goal:

- Data managers need to read the entire SOP and be familiar with all aspects of the document. Data managers looking for ways to improve or add to the system must first understand the system in use.
- Protocol leads and anyone under their supervision need to read and be familiar with the requirements of the Data Stewardship and Data Workflow section of this SOP. The SQL database plays a central role assisting protocol leads by advancing data through defined data processing levels (raw to certified), providing transactional data for the QA/QC Dashboard, and generating the data sets used to publish the annual data package.

Table 1. List of parks where surface elevation table (SET) monitoring is conducted.

Network Code	Park Name	Park Code
NCBN	Assateague Island National Seashore	ASIS
NCBN	Cape Cod National Seashore	CACO
NCBN	Colonial National Historical Park	COLO
NCBN	Fire Island National Seashore	FIIS
NCBN	Gateway National Recreation Area	GATE
NCRN	George Washington Memorial Parkway	GWMP
NCRN	National Capital Parks-East	NACE
NETN	Acadia National Park	ACAD
NETN	Boston Harbor Islands National Recreation Area	BOHA
SECN	Canaveral National Seashore	CANA
SECN	Cape Hatteras National Seashore	CAHA
SECN	Cape Lookout National Seashore	CALO
SECN	Cumberland Island National Seashore	CUIS
SECN	Fort Frederica National Monument	FOFR
SECN	Fort Matanzas National Monument	FOMA
SECN	Fort Pulaski National Monument	FOPU
SECN	Timucuan Ecological and Historic Preserve	TIMU
SFCN	Biscayne National Park	BISC
SFCN	Salt River Bay National Historical Park and Ecological Preserve	SARI
SFCN	Virgin Islands National Park	VIIS

Data Stewardship and Data Workflow

Two data sets are being collected as a part of this surface elevation table (SET) monitoring effort: SET pin data and marker horizon data. Storage of QA/QC information for these datasets is included in this data structure (e.g., data validation and flagging). An overview of the data management processes associated with the monitoring is shown in Figure 1.

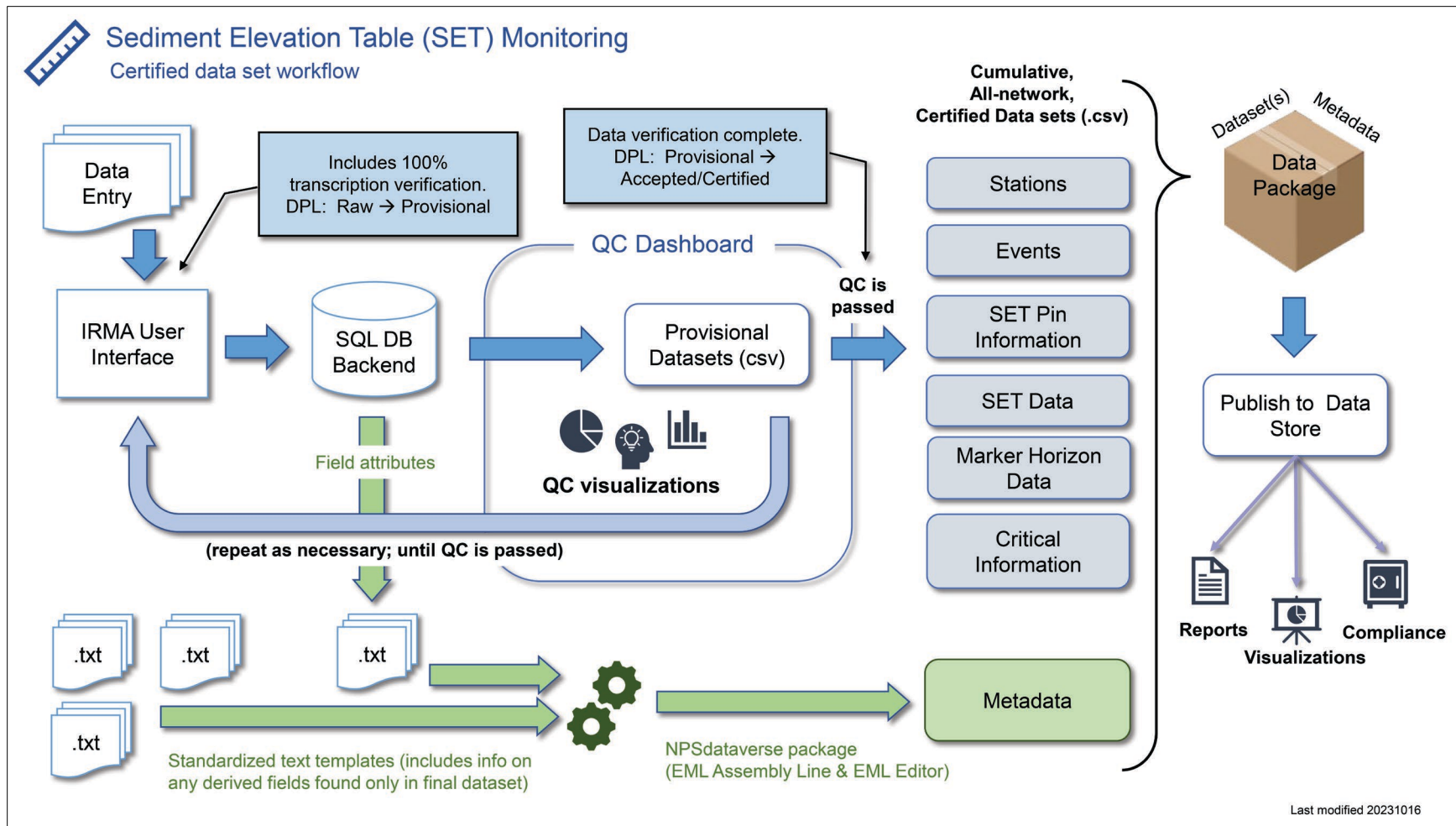


Figure 1. Surface elevation table monitoring workflow diagram.

Change in surface elevation (pins) and accretion (marker horizons) are measured at regular intervals in 20 national park units (see Table 1). Upon return from the field, the original paper data sheets (data processing level = raw) are entered into the data management system using the web-based user interface by the protocol lead, technician, or other staff member. Staff then conduct a check of the field data forms against the digital entries to ensure that transcribed data are complete and accurate (data processing level = provisional). Provisional data sets (in the format of SQL views) undergo a second round of review using a QA/QC Dashboard to identify contextual and logic-based data issues. Errors or deviations from standard practices that are discovered at this stage are corrected, flagged, and/or documented using the web-based user interface. Provisional data are iteratively corrected and reviewed until the protocol lead confirms the data meet the quality assurance standards defined in the protocol (data processing level = accepted).

Roles and Responsibilities

Data stewardship is the responsibility of all network staff, from the field crew who collect, verify, and enter data; to the protocol lead and technician who validate, analyze, and summarize data; to the network data manager, who ensures that master datasets are of high quality and proper data management standards and practices are in place and followed. General roles and responsibilities of key personnel are described in Table 2. Database permissions for the user interface for data entry are outlined in Table 3. In small networks, individual staff may perform one or more of these roles.

Table 2. Wetland Elevation Monitoring protocol roles and responsibilities for core personnel.

Roles	Responsibilities
Protocol lead	Operational lead for SET monitoring. Data analysis and review. Publication and report development. Lead protocol development and publication. Establish partnerships and create agreements as needed. Data and metadata review prior to publication. Ensures data quality.
Technician	Management and maintenance of SET operation. Equipment management, field data collection, and field crew training. Data analysis and review. Publication and report development. Ensures protocols are followed. Data and metadata review prior to publication.
Partners	Field data collection, equipment calibration and management.
Data Manager	Technical point of contact for all tasks concerning network and NPS database and information management standards, including (but not limited to) security, data quality, data documentation, database structure, reporting, and linkages to other NPS information management systems. Technical point of contact for all tasks concerning database, application development, and database administration and publication. Ensures data quality. Publishes data packages in alignment with IMD standards.

Table 3. Permissions by user role for the SET user interface.

Action	Data Contributor	Project Manager	Administrator
Manage Staff	No	Yes	Yes, for all networks
Manage Sites/Stations	Yes, for their network	Yes, for their network	Yes, for all networks
Add or delete raw field data	Yes, for their network	Yes, for their network	Yes, for all networks
Set Raw data to Provisional	Yes, for their network	Yes, for their network	Yes, for all networks
Edit Provisional Data	Yes, for their network	Yes, for their network	Yes, for all networks
Delete Provisional Data	No	Yes, for their network	Yes, for all networks
Set Provisional to Accepted/Pre-Certified/Certified	No	Yes, for their network	Yes, for all networks
Edit Accepted/Pre-Certified/Certified Data	No	Yes, for their network	Yes, for all networks
Edit Lookup Tables	No	No	Yes, for all networks
Manage Data Contributors	No	Yes, for their Network	Yes, for all networks
Manage Project Managers	No	Yes, for their Network	Yes, for all networks
Manage Bureau Data Managers	No	No	Yes, for all networks
Grant access to dashboard views	No	No	Yes, for all networks

Exporting Data

In most cases, data generated by the SET protocol should be obtained by downloading the [latest published data set from DataStore](#). However, in certain cases, it may be necessary to obtain data directly from the database (e.g., when the database contains data collected after the most recent formal publication). There are two ways for NPS staff to export data from the SET database system. Both methods utilize standardized SQL views stored in the back end of the database. Data views in the SQL back end exist to keep data exports uniform across use cases (e.g., dashboards, visualizers, data packages). For a list of the views and fields see SOP 2.3.07 (Wright et al. 2022a).

Granting Permission to Export Data via the User Interface

To gain access to the schema and views for export, users must first be granted access to the database back end via the SET User Interface (UI). Access can only be granted by a database administrator (see Figure 2).

Edit User [X]

* indicates required field

* Role: Data Contributor ▼

* Active: ☒

☒ Grant - Database Access

Unassign User: ☐

First Name: Abigail

Last Name: Gonzalez

Email: abigail_gonzalez@nps.gov

User Code: abigailgonzalez@nps.gov

Cancel Save

Figure 2. Access to data export is granted through the ‘Manage Users’ tab in the SET UI.

Data Export Option 1: Using the Standalone R Script

An [R code](#) script (Wright et al 2022b) is available on DataStore to all current database users. The R code will connect directly to the SET SQL database back end and download six .csv files (as detailed in the ‘Analytical Data Views’ section below) into a folder of your choosing. This script also allows for exporting data for any combination of networks. You must have access (supplied separately) to the server address information. If you do not have access to the script or server information, contact a SET database administrator or your network data manager.

Data Export Option 2: Using the SET-Viz Dashboard

Data can also be exported using the SET-viz dashboard tool, available via GitHub. For more information, see the ‘Data Visualizer’ section below.

Analytical Data Views

To provide SET data of consistent content and structure, six views were created in the SQL database. These views are used to pass formatted data to all downstream QA/QC applications as well as to provide the data used to publish annual, public, machine-readable data packages (Figure 3). The analytical data views are:

1. vw_dbx_marker_horizon_data—Exports as marker_horizon_data. Contains information about the parks, sites, marker horizon plots, visit dates, core types and notes, and data processing levels for marker horizon data.
2. vw_dbx_SET_pin_information—Exports as pin_group_data. Contains information about the pin collections.
3. vw_dbx_SET_data_FINAL—Exports as pin_data. Contains pin reading data, data processing levels, pin length and notes, and validation status. This view is built on several sub views.
4. vw_dbx_stations—Exports as station_data. Contains information about the sites and stations in each park, along with locations, active status, and protected status.
5. vw_dbx_critical_information—Exports as critical_information_data. Contains information about event level deviations from protocol (category and description) likely to affect data quality or consistency.
6. vw_dbx_events—Exports as event_data. Contains information about individual events, including network and park information, site and station information, date and time information, and if data were collected.

Details on view fields and aliases are found in SOP 2.3.07 *Monitoring Coastal Wetland Elevation Data Management and Database Technical Documentation—Version 2.1* (Wright et al. 2022a).

Metadata Views

In addition to the analytical data views mentioned in the previous section, data about the architecture of the database itself is summarized in two metadata views. Data returned by these views are used to assist with database documentation (e.g., used to generate Appendices A–D) and metadata creation for the annual data package. These views centralize key descriptive information about the database and are easily refreshed as the database is updated. The metadata views are:

1. vw_Metadata_Table—A list of all database tables, their fields, and attributes (table description, field descriptions, field metrics).
2. vw_Metadata_View—A list of the database fields selected for the transactional data views (table description, field descriptions, field metrics).

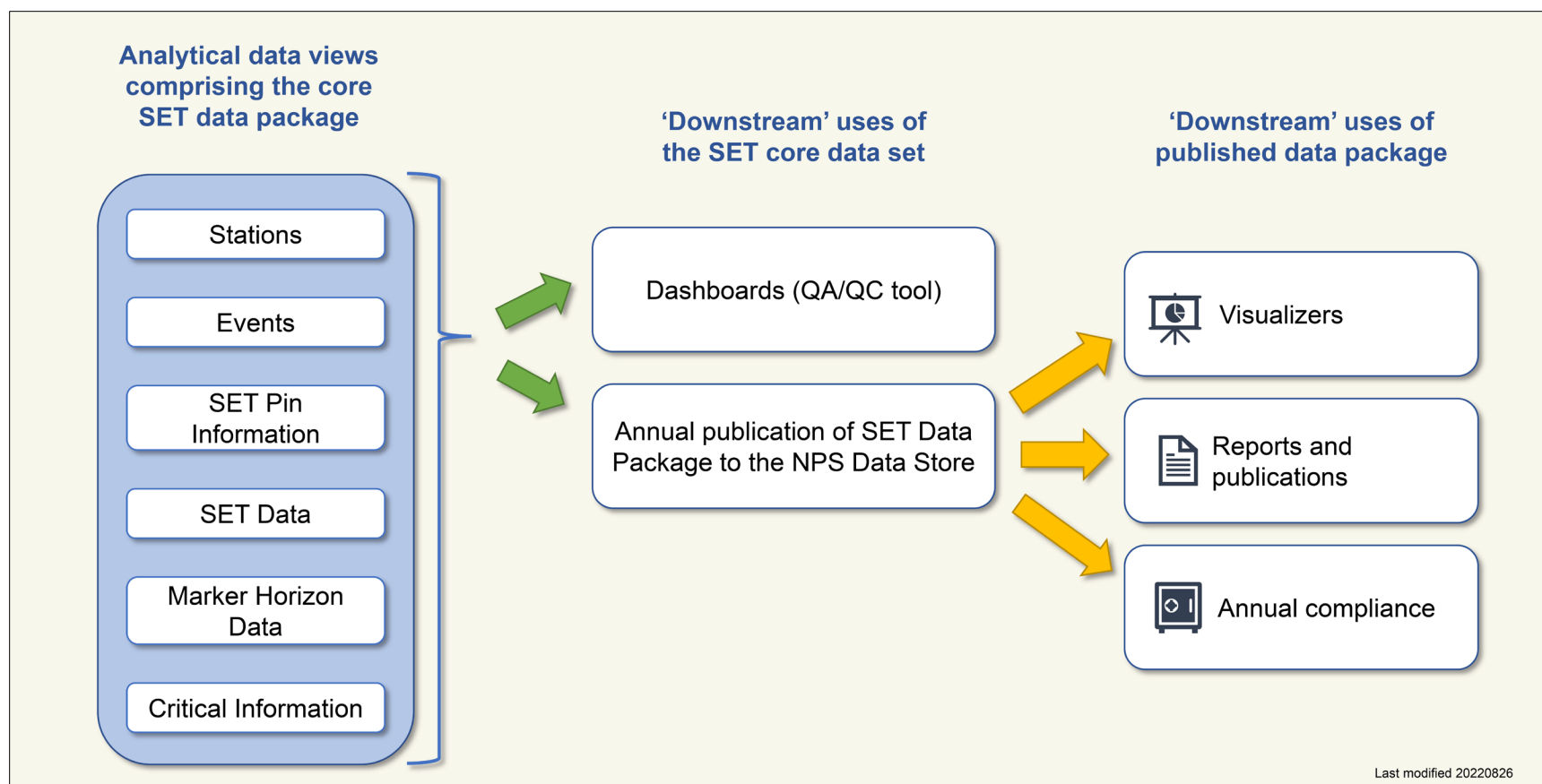


Figure 3. Graphic illustrating expected data reporting workflows in the SET database.

Data QA/QC Dashboard

Overview

The NPS Wetland Elevation QA/QC dashboard is an interactive tool to facilitate QA/QC and visualization of unpublished wetland data collected in National Park units. The dashboard, available [publicly on GitHub](#), utilizes a specifically formatted data set exported from the NPS SET database to generate its summary tables and graphs. You must have database access OR the downloaded data files to run the dashboard (see Data Exporting section above). This document covers the technical requirements and instructions for configuring and running the Wetland Elevation QA/QC Dashboard and includes the following steps (each discussed in detail below):

1. Install and configure Git
2. Clone the SET-viz project to your local computer
3. Open the Dashboard
4. Connect the Dashboard to SET data
5. Update the Dashboard
6. Suggest changes to the Dashboard

Steps 1–2 should only need to be completed once per local computer/user. Steps 3–6 are required per use or as updates to the data or dashboard application dictate.

Installation and Use Requirements

1. A reliable DOI network connection (direct connection or VPN).
2. Administrative rights to download software on your computer.
3. Read-only rights to the NPS IMD Surface Elevation Table (SET) Database; to obtain database access contact your network data manager or SET database administrator.
4. R installed. RStudio is a helpful interface but is not strictly required.
5. Valid 64-bit 'ODBC Driver for SQL Server' installed on your computer. To check this:
 - a. Click the Windows Start button.
 - b. Type 'ODBC Data Sources (64-bit)'
 - c. 'Drivers' tab.
 - d. Scroll down to verify 'SQL Server' exists.

NOTE: You will not need a GitHub account to clone a repository to your computer, but you will need one to post 'issues' (i.e., report bugs or make requests for fixes or new features) to the project or

receive notifications when the repository is updated. To get a GitHub account, follow the instructions at this [GitHub Help link](#) (use your NPS email address):

Install and Configure Git

1. Download and install the latest 64-bit version of [Git for Windows here](#):
2. Double-click the downloaded '.exe' file to start the installation.
3. You will have numerous customization choices to make during the Git installation. To use Git from RStudio (per instructions in this document), accept the default settings provided on each window, except:
 - a. When 'Choosing HTTPS transport backend' make sure you select to use the native Windows Secure Channel library (Figure 4).

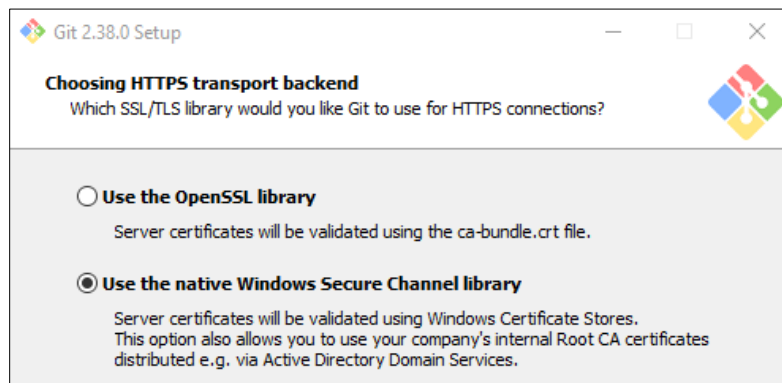


Figure 4. Choosing the HTTPS transport back end.

4. Verify that Git installed correctly:
 - a. Go to your computer's Command Prompt (Press Windows + r to open the 'Run' box). Type 'cmd' and then click 'OK.'
 - b. Type 'git' in the terminal. If Git installed successfully, you should see a list of common Git commands. If instead you get a message that Git wasn't found, try installing again.

Clone the SET-viz Project to your Local Computer

1. If this is your first time cloning a GitHub repository, first enable Git in RStudio:
 - a. Click Tools → Global Options → Git/SVN → 'Enable version control interface for RStudio projects' → (when the path to Git shows up) click 'Apply' → 'OK.'
2. In RStudio, click File → New Project → Version Control → Git.

- Using a web browser, go to the [NPS GitHub page](#). Use 'Find a repository' to search for 'SET-viz.' Select SET-viz to advance to its repository page and use the green 'Code' button to copy the link you will use to clone the repository onto your computer (Figure 5).

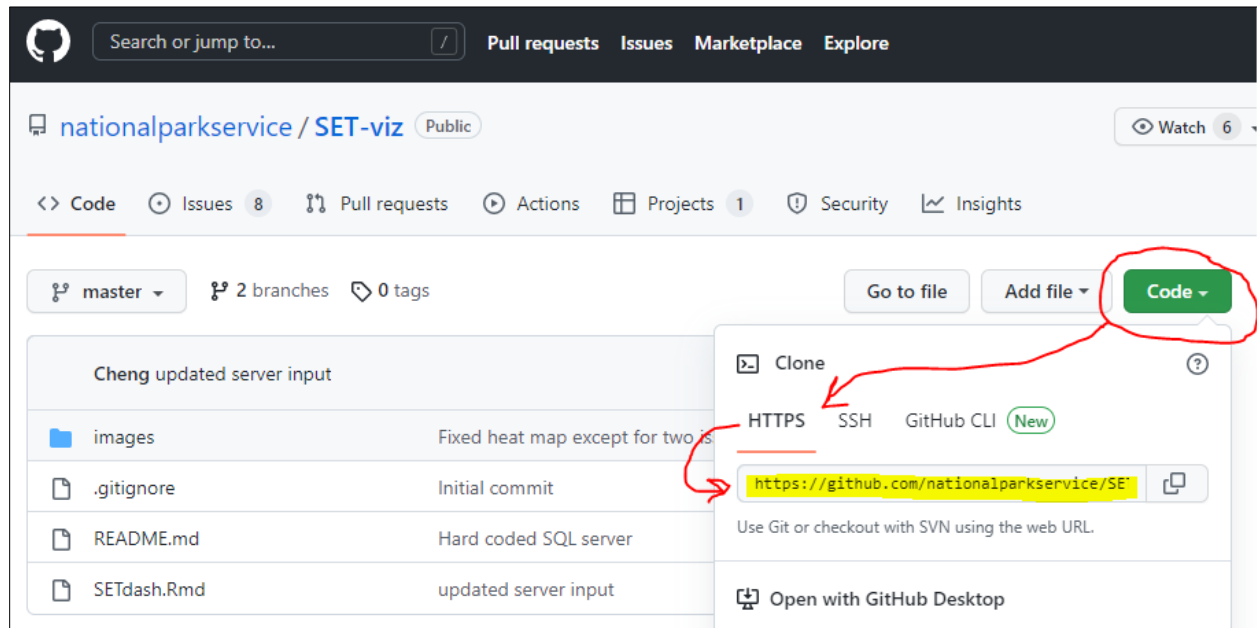


Figure 5. SET-viz repository window with clone link.

- Back in R-Studio's 'New Project Wizard' window complete the following:
 - Repository URL: Enter the URL copied from the SET-viz repository page.
 - Project directory name: will autofill as "SET-viz" if you used the correct URL.
 - Create project as subdirectory of: This is where the SET-viz repository files will be saved to. The path will default to your RStudio working directory but can be changed to suit your preferences.
- When ready, click 'Create Project.' You should see downloaded files from the remote GitHub repository.
Error handling: If you get an error message about an SSL certificate, go to your computer's Command Prompt (Press Windows + r to open the 'Run' box. Type 'cmd' and then click 'OK') and type in 'git config --global http.sslBackend channel' (without the quotes and with all spaces as shown), then press 'Enter' and attempt entering the project URL again in RStudio.
- You should now see a 'Git' tab in the upper right pane (aka Workspace Browser pane) and SET-viz project files in the lower right pane (aka Plots pane).
- If you do not see the Git tab, and you have verified Git installed correctly (Step 4b), verify that RStudio has the correct path to Git:

- a. Go to your computer's Command Prompt (Press Windows + r to open the "Run" box. Type 'cmd' and then click 'OK').
 - b. Determine the Git path, type 'where git' (for Mac, 'which git') – this will return the path to the cmd git.exe file.
 - c. In RStudio, go to Tools → Global Options → Git/SVN – this will return the path of the bin git.exe.
 - d. Verify that the root paths to both files is the same.
8. If everything above looks fine, restart RStudio and look again for the Git tab (Figure 6).

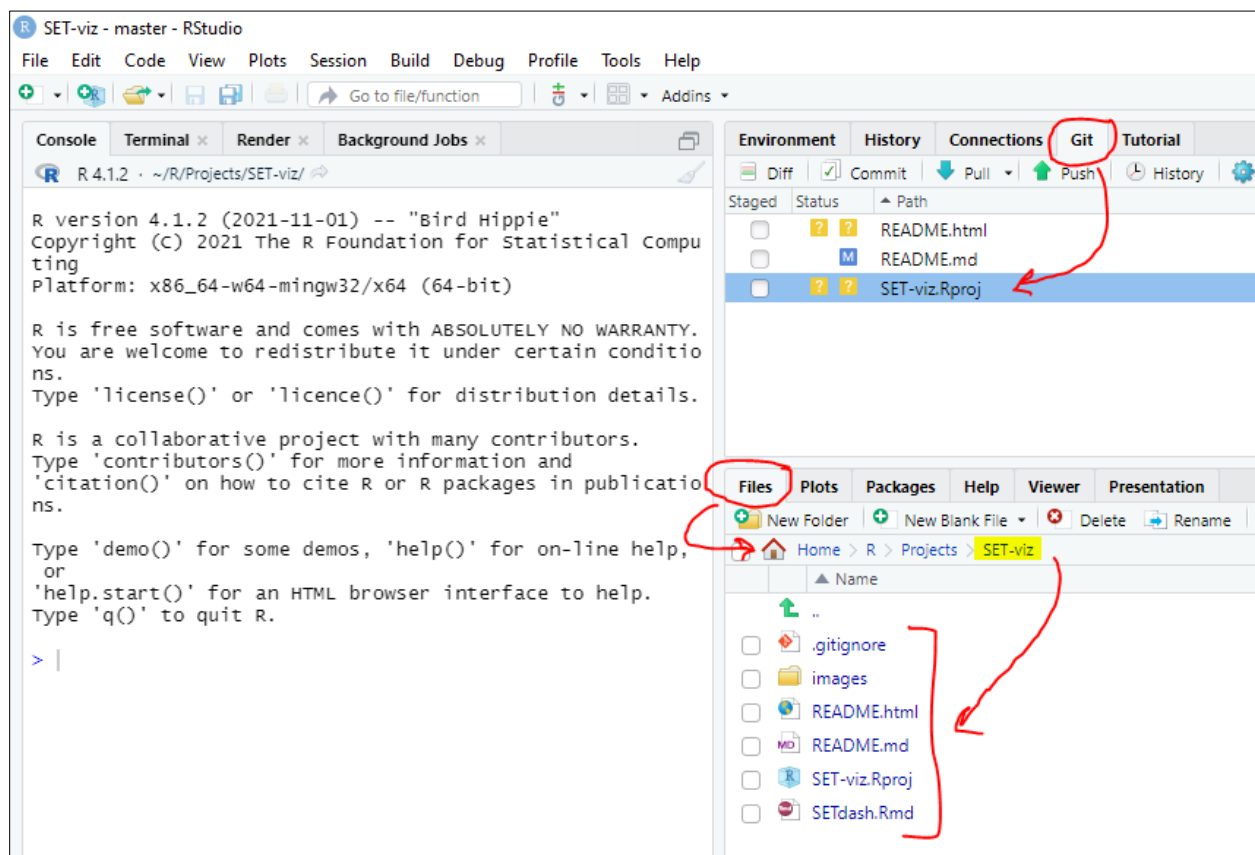


Figure 6. RStudio window showing Git tab.

Open the Dashboard

Now that the project files are loaded locally on your computer, it is time to open the Dashboard. The README.md file that downloaded with the repository provides basic instructions on getting started (click 'Preview' on the source pane after it opens to see formatted version), but the basic steps are as follows:

1. In RStudio, if you don't already have it, install the flexdashboard package. To do this, go to your console window type 'install.packages("flexdashboard")' (without the single quotes) and press Enter.
2. In RStudio → Plots pane → Files tab → double-click the .Rmd dashboard file to open it.
3. Click the 'Run Document' icon near the top of the RStudio window. The SET Dashboard will open in RStudio.
4. In the upper left of the Dashboard click 'Open in Browser' for a full-screen view.

Connect the Dashboard to SET Data

The Dashboard is an interactive tool for QA/QC and visualization. The data set used by the Dashboard must follow the very specific format generated by the transactional SQL database that NPS uses to store wetland elevation data. Tools built into the Dashboard itself are used to retrieve the latest version of the data directly from the SQL database. The Dashboard will format the data and combine the formatted data into a single .RDS file that the user can save to a local computer. Each time the Dashboard is used the user can choose to either load a previously saved .RDS file or to download the latest data directly from the SQL database. Instructions for both methods are provided below and on the 'Home' tab of the Dashboard.

1. **Option 1:** Pull fresh data from SQL.
 - a. Enter the SQL Driver: Usually this is simply "SQL Server." If this fails, verify that you have a valid SQL Server ODBC driver installed on your computer (see Requirements in the Data Visualizer—Overview section).
 - b. Enter the SQL database server path: Enter the 'Connection' path listed under SET Back end, Production found in [this PDF](#). NOTE: Syntax is important, path should use the following format: servername\share (i.e., INPXXXXXX01\ntwk).
NOTE: If you do not have access to the DataStore help document (the above PDF) with the server addresses, please contact your network data manager.
 - c. Select network(s) to import data from.
 - d. Click the green 'Import/Summarize Data from SQL' button. NOTE: You must be on a DOI direct network connection (such as VPN).
2. **Option 2:** Import an existing .RDS file.
 - a. Follow the instructions on the Dashboard's 'Home' tab to browse to and import an .RDS file.

Update the QA/QC Dashboard

The dashboard will periodically undergo planned updates to improve its utility. Whenever the dashboard is updated in the remote Git repository (e.g., as issues are addressed), you can 'pull' the

current version of the dashboard by clicking the blue down-arrow icon on the ‘Git’ tab of your local dashboard repository (See Figure 7).

1. RStudio → Set-viz project → ‘Git’ tab in the upper right pane (aka Workspace Browser pane)
2. Click on the blue ‘Pull’ down-arrow. Note: If your Push/Pull arrows are disabled (‘grayed out’), type ‘git push -u origin master’ in your computer terminal/shell (see Install and Configure Git, Step 4).

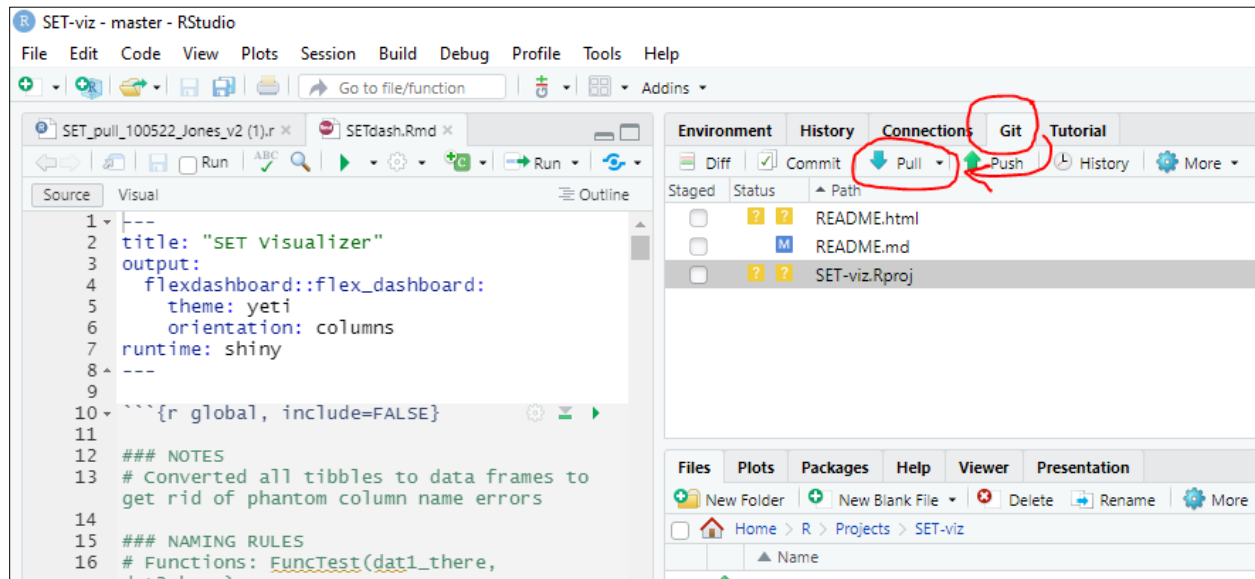


Figure 7. Graphic showing the steps to pull a new version of the dashboard.

3. You should see a pop-up window like this, indicating which files in your local Dashboard repository have been replaced with newer versions:

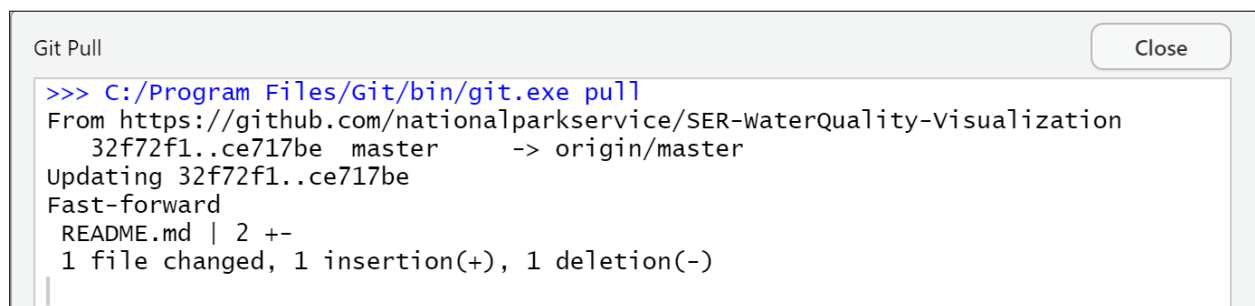


Figure 8. Window showing versions.

If you already have the most up-to-date version of the Dashboard, the message in the window will simply say ‘Already up to date.’ If the Dashboard is being actively updated, you may want to update your local Dashboard repository regularly to keep up with Dashboard fixes and enhancements.

Should I sign-up for a GitHub account? Project tracking and management for the development of the SET-viz project is handled through GitHub. To participate in the governance steps outlined below you must have a GitHub account. [Apply for an account here](#).

Suggest Changes to the Dashboard

Suggesting changes to the Dashboard requires a GitHub account. Once you have had a chance to test out the Dashboard, you can use the GitHub 'issues' tab to generate a feature request, ask a question, or report a bug in the Dashboard. A feature request, for example, would be something like “It would be useful to be able to rule the world (or other helpful feature) with the Dashboard.” If you are reporting a bug in the Dashboard, provide specific details about the dashboard page/tab that is problematic, the specific data set used when you found the bug (if the problem appears only with certain data sets), and any R error message that was generated.

To do this:

1. Using a web browser, go to the [NPS GitHub page](#) and use ‘Find a repository’ to search for ‘SET-viz.’ Select SET-viz to advance to its repository page.
2. Check the list of pending issues under the 'Issues' tab (Figure 8), to make sure no one else has already posted the issue/suggestion.

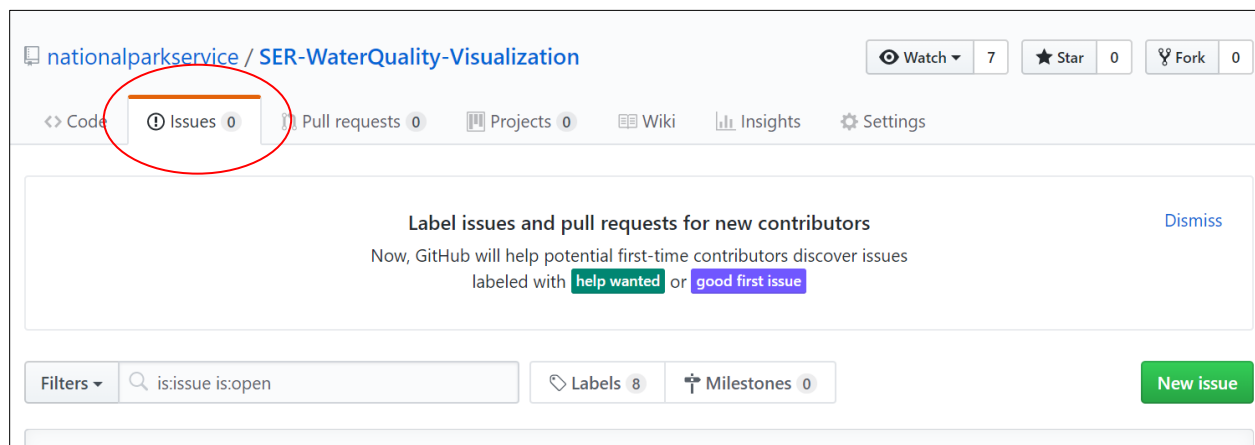


Figure 8. Graphic showing the 'issues' tab.

3. To submit a new issue, click the green ‘New issue’ button (Figure 9).
 - a. Provide a short informative title, then add a more detailed explanation in the box below the title box.
 - b. Click ‘Submit new issue’.
4. Once the issue is addressed or fixed, it will be marked as ‘closed’ on the remote repository issues tab. You should receive an email informing you that the issue you posted has been

resolved and the dashboard has been updated in the remote 'Git' repository. The email will be sent to the email address you set up your GitHub account with.

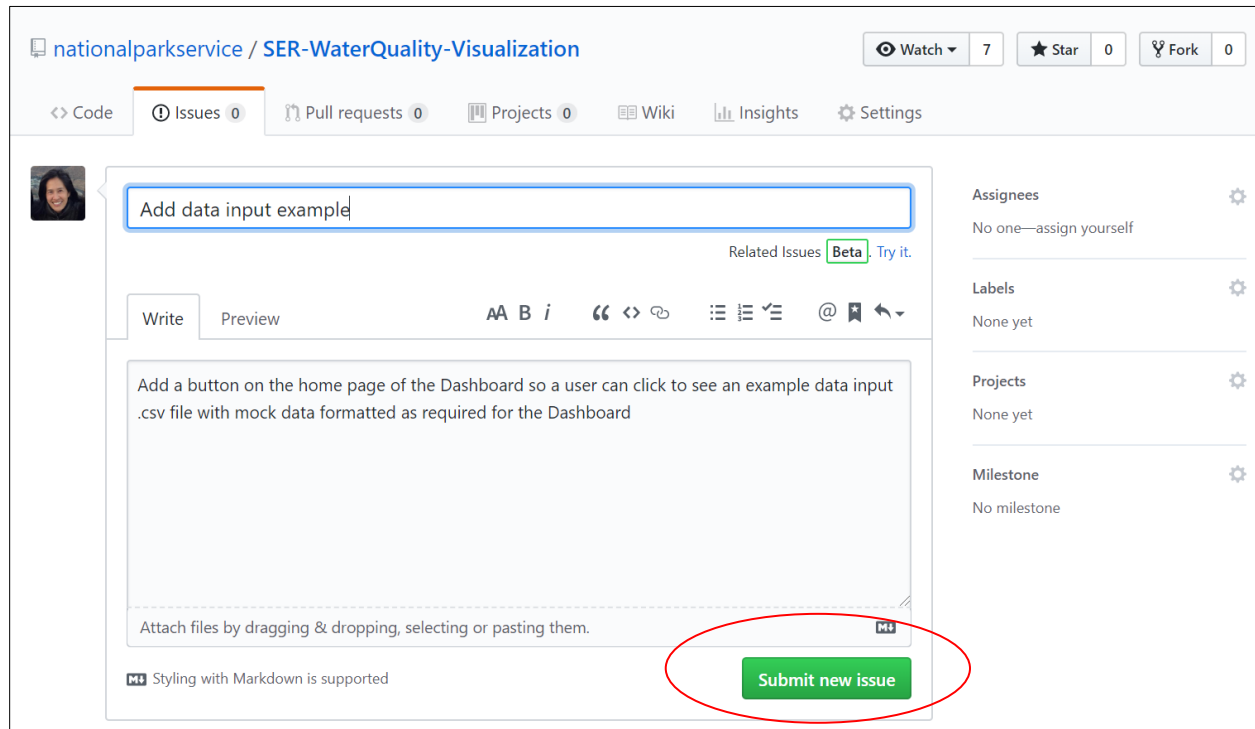


Figure 9. Graphic showing the 'Submit new issue' button.

Collaborations and Notifications

To get email notifications when issues, etc. are posted to the repository you will need to 'watch' the SET-viz project and configure notifications in your GitHub profile ([requires a GitHub account](#)).

1. Sign into your GitHub account
2. [How to manage your 'watched' repositories](#)
3. [Choose the type of activity on GitHub that you want to receive notifications for and how you want these updates delivered](#)

References & Tutorials for Git

<https://beta.rstudioconnect.com/jennybc/happy-git-with-r/>

<https://github.com/maptime/getting-started-with-git-and-github/blob/master/README.md>

<https://guides.github.com/introduction/git-handbook/>

https://jennybc.github.io/2014-05-12-ubc/ubc-r/session03_git.html

[NPS IMD Advanced R Training \(katemiller.github.io\)](#)

Data Package Publication Preparation

The SET data publication process is a multi-step process, consisting of exporting the data files from the database, preparing the DataStore records, applying metadata to the data files, and publishing the final data package. The process is outlined in Figure 11.

NOTE: The data package publication process has been updated considerably since the last SET data package was published and this SOP was initially reviewed. It is imperative to download a new version of NPS Dataverse and become familiar with any updates BEFORE publishing a new data package each year.

The most current template can be obtained by clicking **File > New File > R Markdown > From Template > Editable_EML_Creation_Script**.

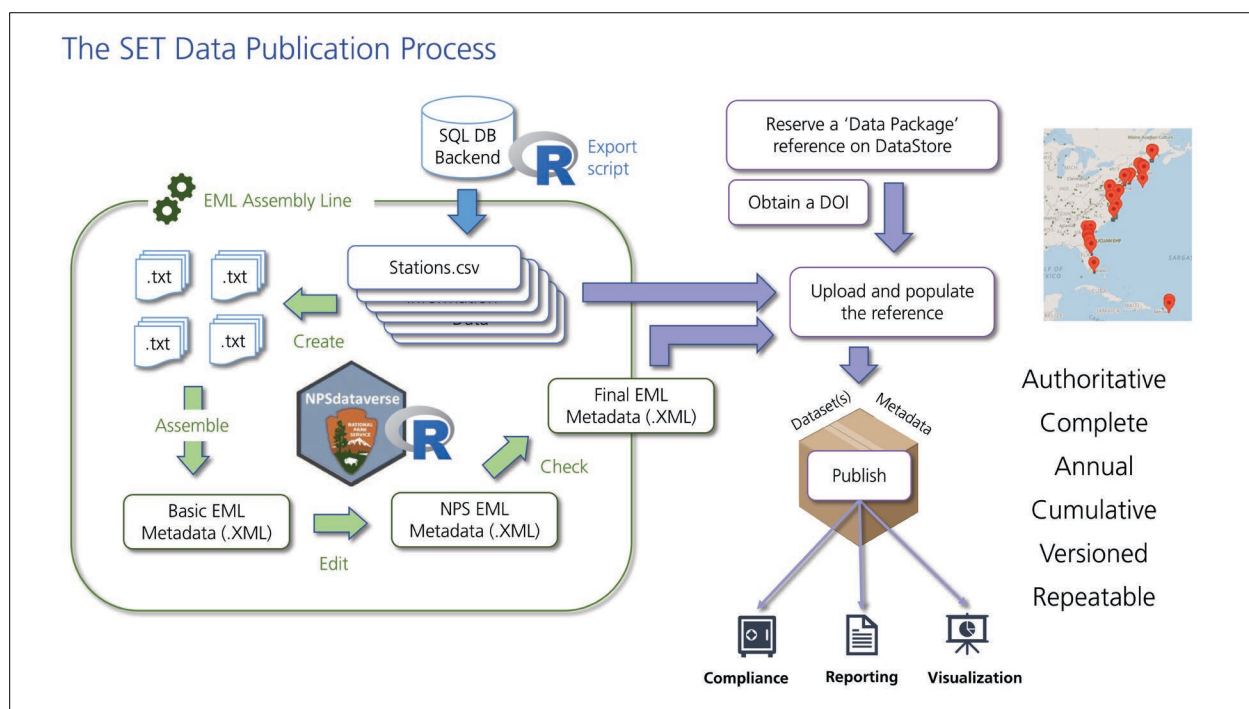


Figure 11. Graphic showing the SET data publication process.

Create Draft DataStore Records

Create a Draft DataStore Record for the Data Package

Before you can apply metadata to the data package, you will need a draft DataStore reference number to insert into the metadata scripts. To do this:

1. Open the [SET Project in DataStore](#).
2. Find the link to and open the reference containing the current data package.
3. Verify you have access to edit the current data package reference.

4. Click Actions → Clone. Make sure NOT to activate the record but leave it in draft form. Avoid filling in the DOI number until the end, as this can lock the record prematurely.
5. Note the draft DataStore reference number for use in the metadata scripts.

Create a Draft DataStore Record for the Metadata Creation Package

Repeat Steps 1–5 above for the metadata creation package. The draft record for the metadata creation package is not used in the R scripts but will be needed in Part II when the new Metadata Creation Package is published.

Write EML Metadata

Once the cumulative data sets have undergone all desired QA and QC checks and all pertinent records have been appropriately flagged, they are ready to be published as an annual Data Package via the NPS IRMA DataStore. In addition to the data files, an additional required element of the Data Package is an [Ecological Metadata Language](#) (EML; Jones et al. 2019) metadata file in .xml format describing the data. While there are numerous online and desktop tools available for creating EML-compliant metadata, the process for the SET Monitoring Program uses a workflow that leverages the freely available R package [EMLassemblyline](#) (Smith 2022) and several R Tools developed by the National Park Service.

Getting Started

To prepare for data package publication, it's helpful to gather all the prerequisite files and install all required software in advance.

Install Required Applications

Metadata creation requires many software applications and R packages. Installing and updating them prior to running the R scripts will simplify the process. You will need the following tools:

Applications (these are generally available for download by the user on Software Center. For assistance, reach out to your local IT Helpdesk).

- R (R Core Team 2022)
- RStudio (RStudio Team 2019)

Packages from CRAN

To install these packages, open RStudio, and on the 'Tools' tab, click "install package" and type in the package names separated by a comma. If you have issues installing packages, try changing the CRAN to the 'Global (CDN) – Rstudio' connection (Tools → Global Options → Packages, and change the Primary CRAN repository) and using a wifi connection instead of a VPN or direct DOI Network connection.

- [devtools](#) (Wickham et al 2022)
- [Lubridate](#) (Grolemund and Wickham 2011)

- [tidyverse](#) (Wickham et al 2019)
- [Stringr](#) (Wickham 2022)

Packages from GIT

- [EMLAssemblyline](#) (Smith 2022; included in NPSdataverse)
- [NPS Dataverse](#) (Baker and Patterson 2023), which includes:
 - QCkit (Baker, Patterson, DeVivo, Quevedo, and Sherman 2023)
 - EML Editor (Baker and Patterson 2023)
 - DPChecker (Baker and Wright 2023)
 - NPSutils (Baker, DeVivo, and Patterson 2023)
 - EMLassemblyline (Smith 2023)
 - EML (Boettiger 2022)

To install packages from GIT, make sure Devtools is installed, and use the following lines of code in RStudio:

```
devtools::install_github("EDIorg/EMLassemblyline")
```

```
devtools::install_github("NationalParkService/NPSDataVerse", force = TRUE)
```

Scripts

Note: If you are publishing a new version of an existing data package, it is highly recommended to use the previous scripts that will already be customized and require minimal edits (refer to the ‘Collect Previous Files from DataStore’ section below). To start a new data package from scratch, download the following:

[EML Creation Script](#)

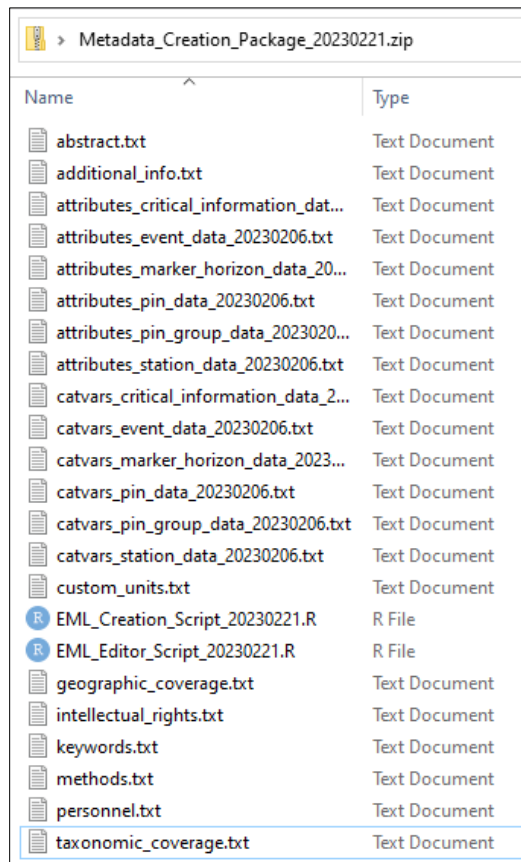
Make Sure R is Current

Open R Studio. Go to ‘Tools’ → ‘Check for Package Updates.’ Make sure your packages are current. Make sure your working directory is set where you want it.

Collect Previous Files from DataStore

Everything you need will be listed in the [SET Database System and Data Publication Workflows Project](#) (NPS 2023). A link to the project can also be found pinned to the top of the database channel in the SET team. Download the most recent ‘Metadata template files.’ This will be a zip file that includes the metadata template files (.txt) and R script used for the most recent data package that was published. Note that the template file name after the prefixes ‘attributes_’ or ‘catvars_’ MUST match the file name of the .csv files that contain the data and will need to be modified before running the EML creation script (Figure 12). If they don’t match, the script will generate new empty templates.

Download all the metadata template files to your working directory.



Name	Type
abstract.txt	Text Document
additional_info.txt	Text Document
attributes_critical_information_dat...	Text Document
attributes_event_data_20230206.txt	Text Document
attributes_marker_horizon_data_20...	Text Document
attributes_pin_data_20230206.txt	Text Document
attributes_pin_group_data_2023020...	Text Document
attributes_station_data_20230206.txt	Text Document
catvars_critical_information_data_2...	Text Document
catvars_event_data_20230206.txt	Text Document
catvars_marker_horizon_data_2023...	Text Document
catvars_pin_data_20230206.txt	Text Document
catvars_pin_group_data_20230206.txt	Text Document
catvars_station_data_20230206.txt	Text Document
custom_units.txt	Text Document
EML_Creation_Script_20230221.R	R File
EML_Editor_Script_20230221.R	R File
geographic_coverage.txt	Text Document
intellectual_rights.txt	Text Document
keywords.txt	Text Document
methods.txt	Text Document
personnel.txt	Text Document
taxonomic_coverage.txt	Text Document

Figure 12. Example of files included in the metadata creation package zip file.

Note: if you plan to download and use the ‘Metadata template files’ from the previous data publication (listed in the ‘Collect Previous Files from DataStore’ section above), most of the following steps will be complete. If you need to start from scratch, or troubleshoot existing scripts, the following steps will walk you through the process.

Create and Edit EML Metadata Using NPS R Scripts

Before Running the NPS EML Creation and NPS EML Editor R Scripts

Place the six exported data set files (.csv) in your designated working folder (See the Exporting Data section above for methods available to produce these six files).

Note: Each line or set of code in the EML creation script should be run separately and sequentially. In particular, the `make_eml()` function should not be run until all previous steps have been completed successfully.

The EMLassemblyline Metadata Templates

The ‘EMLassemblyline Functions’ portion of EMLassemblyline R package creates an EML metadata file in part by querying the data sets and creating a series of intermediate text files to store such information as table and field names, codes and definitions for categorical variables, geographic

extent, and taxonomic information. Once generated, the user then updates these files with the appropriate metadata using a text editor (e.g., adds definitions for fields and categorical variables determined to be in the data set by the script). Subsequent EML creation script steps then harvest and assemble the information stored in these text templates to create the final .xml metadata file. Because the content of these text files rarely changes, they can usually be reused to generate metadata for subsequent annual data packages. Note: The EML Creation script will only generate text template files if they are missing from the working directory; it will not overwrite existing template files. Current template text files for the SET Protocol are described in Table 4 and can be found on the NPS DataStore (<https://irma.nps.gov/DataStore/Reference/Profile/2295516>).

Table 4. List of EML metadata template files for use with the EML AssemblyLine R package.

Template file	Description
abstract.txt	Describes the salient features of a data set in a concise summary much like an abstract does in a journal article.
additional_info.txt	Describes additional info not included elsewhere.
attributes_critical_information_data.txt	Describes columns of the “Critical Information Data” data set (classes, units, datetime formats, missing value codes).
attributes_event_data.txt	Describes columns of the “Event Data” data set (classes, units, datetime formats, missing value codes).
attributes_marker_horizon_data.txt	Describes columns of the “Marker Horizon Data” data set (classes, units, datetime formats, missing value codes).
attributes_pin_data.txt	Describes columns of the “Pin Data” data set (classes, units, datetime formats, missing value codes).
attributes_pin_group_data.txt	Describes columns of the “Pin Group Data” data set (classes, units, datetime formats, missing value codes).
attributes_station_data.txt	Describes columns of the “Station Data” data set (classes, units, datetime formats, missing value codes).
catvars_critical_information_data.txt	Describes categorical variables from the “Critical Information Data” data set (if any columns are classified as categorical in table attributes template).
catvars_event_data.txt	Describes categorical variables from the “Event Data” data set (if any columns are classified as categorical in table attributes template).
catvars_marker_horizon_data.txt	Describes categorical variables from the “Marker Horizon Data” data set (if any columns are classified as categorical in table attributes template).
catvars_pin_data.txt	Describes categorical variables from the “Pin Data” data set (if any columns are classified as categorical in table attributes template).

Template file	Description
catvars_pin_group_data.txt	Describes categorical variables from the “Pin Group Data” data set (if any columns are classified as categorical in table attributes template).
catvars_station_data.txt	Describes categorical variables from the “Station Data” data set (if any columns are classified as categorical in table attributes template).
custom_units.txt	Describes non-standard units used in a data table attribute template.
geographic_coverage.txt	Describes where the data were collected.
intellectual_rights.txt	Describes how the data may be used. Releasing without restriction (CC0) or with minimal attribution (CC BY) maximizes value and future use.
keywords.txt	Describes the data in a small set of terms. Keywords facilitate search and discovery on scientific terms, as well as names of research groups, field stations, and other organizations. When available, using a controlled vocabulary or thesaurus vastly improves discovery.
methods.txt	Describes the data creation methods. Includes enough detail for future users to correctly use the data. Lists instrument descriptions, protocols, etc.
personnel.txt	Describes the personnel and funding sources involved in the creation of the data. This facilitates attribution and reporting.
taxonomic_coverage.txt	Describes biological organisms occurring in the data and helps resolve them to authority systems. Not currently utilized by SET monitoring.

When text templates are created for the first time or when changes to the structure of the six .csv files require changes be made to the previous year's templates, edit the.txt files using a text editor or spreadsheet application as necessary. For details, see the NPS template editing guidelines at https://nationalparkservice.github.io/NPS_EML_Script/edit_tmpls.html.

Run the NPS EML Assemblyline Script

NOTE: The line numbering here applies only to this specific data package creation, and the line numbers may change in future data package creation scripts. They are included here as a general reference.

Open the EML_Creation_Script included in the zip file that was downloaded in the 'Collect Previous Files from DataStore' earlier in this document and follow the steps outlined below.

1. Load the packages you installed earlier (lines 25–27)

```
library(EMLassemblyline)
library(lubridate)
library(tidyverse)
```

2. Change the metadata_id file name to reflect the current date (line 34). This will be the name of the .xml metadata file produced by the R script. The filename MUST end in _metadata.

```
metadata_id <- "YYYYMMDD_SET_metadata"
```

3. Check the data package title and update if needed (line 37)

```
package_title <- "SET YYYY Cumulative Data Package"
```

4. Check the data type. For SET, the data type is 'ongoing' (line 40).

```
data_type <- "ongoing"
```

5. Check your working directory and make sure it is where you expected it (line 46). If you close R or work on another project the working directory may change.

6. Make sure the names on the .csv files match the date in the R code lines 48–53 (Vector of data set file names). Any template files with the prefix 'attributes_' or 'catvars_' MUST match the file name of the .csv files that contain the data and will need to be modified before running the EML Creation script. If they don't match, the script will generate new empty templates.

```
#Vector of data set file names.
```

```
data_files <- c("critical_information_data_YYYYMMDD.csv",
               "event_data_ YYYYMMDD.csv",
```

```
"marker_horizon_data_ YYYYMMDD.csv",
"pin_data_ YYYYMMDD.csv",
"pin_group_data_ YYYYMMDD.csv",
"station_data_ YYYYMMDD.csv")
```

7. Check the data descriptions on lines 69–74. If you are republishing a SET data package, these should not change. If you are running a data package from scratch, insert descriptions of your .csv data files.

8. Set the DSRefCode (line 79) to match the IRMA record number. This will set the DOI record number.

```
DSRefCode<-2298730
```

9. Make sure the start and end dates in the R Code (line 100–101) match the start and end date in the IRMA record.

```
startdate <- ymd("1998-10-14")
enddate <- ymd("2023-03-29")
```

10. There are several functions (1–3) that are commented out in the R code. These should not need to be run if you are republishing a SET data package. If you are starting a new data package, explanations in the code will help you proceed with these three steps.

11. Change the filename on the data_coordinates_table (line 156) to match the station_data .csv file name. This tells the script where to get the point information for the SET stations.

12. Run lines 156–167.

```
data_coordinates_table <- "station_data_20230406.csv"
data_latitude <- "station_latitude"
data_longitude <- "station_longitude"
data_sitename <- "station_code"
template_geographic_coverage(path = working_folder,
data.path = working_folder,
data.table = data_coordinates_table,
lat.col = data_latitude,
lon.col = data_longitude,
```

```
site.col = data_sitename,  
write.file = TRUE)
```

13. Function 5 is commented out because the SET data package does not contain taxonomic information.
14. At this point, if everything has run cleanly with no errors, you are ready to run the ‘Make EML’ code (lines 185–193).
15. If it runs successfully, run the `issues()` line to check for issues. If this is successful, you are ready to move on to the `EML_Editor_script`.

Overview of the NPS EML Editor Script

Once the valid EML metadata file (.xml) has been created, NPS-specific elements and fields can be added using the R `EMLeditor` package (https://github.com/nationalparkservice/EML_editor). The `EMLeditor` package should be installed as part of the `NPSdataverse` package.

Functions available in `EMLeditor` include `set.CUI`, `set.DOI`, `set.DRRdoi`, and `set.parkUnits`. Once any function is used, the tool will also automatically add the National Park Service as a publisher to the metadata unless you set `NPS=FALSE`. If you leave the default setting as `NPS=TRUE`, `EMLeditor` will also assume the data package is being created “by or for the NPS” and add that information to the metadata.

Run the NPS EMLeditor Script

Make sure you have followed all the installation and preparatory steps in the ‘Getting Started’ section.

1. Check your working directory to make sure it is the same as the previous script AND contains the .xml file produced by the previous script.
2. Open the EML editor script that was downloaded from the DataStore in the ‘Metadata Creation Package’ zip file.
3. Load the `NPSdataverse` library (line 3)

```
library(NPSdataverse)
```

4. Use the `EML::read_eml()` function to load your EML file into R as an R object. Check the .xml file name and make sure it matches the one just produced by the EML creation script (line 7).

```
my_meta2 <- EML::read_eml("YYYYMMDD_SET_metadata.xml", from = "xml")
```

5. Use the `set_cui()` function to add information about Controlled Unclassified Information (CUI; line 10)

```
my_meta2 <- set_cui(my_meta2, "public")
```

6. Use the `set_int_rights()` function to add information about Intellectual Rights (line 12)

```
my_meta2 <- set_int_rights(my_meta2, "public")
```

7. Use the `set_doi()` function to add a Digital Object Identifier (DOI) for your data set. This will correspond to the NPS DataStore Reference ID created above (line 16).

```
my_meta2 <- set_doi(my_meta2, 2298730)
```

8. Make sure the language is set to English (line 23).

9. Check the content unit links and producing units (lines 24–30).

```
# Add content unit links
```

```
park_units <- c("ACAD", "ASIS", "BISC", "BOHA", "CACO", "CAHA", "CALO",  
"CANA", "COLO", "CUIS", "FIIS", "FOFR", "FOMA", "FOPU", "GATE", "GWMP",  
"NACE", "SARI", "TIMU", "VIIS")
```

```
my_meta2 <- set_content_units(my_meta2, park_units)
```

```
# Add the producing unit(s)
```

```
network_units <- c("NCBN", "NCRN", "NETN", "SECN", "SFCN")
```

```
my_meta2 <- set_producing_units(my_meta2, network_units)
```

10. If the code up through line 31 runs cleanly, run lines 36–40 to validate the metadata and run the DPChecker to check the metadata schema.

```
# Validate your EML. If eml_validate returns errors, inspect them and  
fix them.
```

```
EML::eml_validate(my_meta2)
```

```
# Run DPChecker package to test whether your EML is schema-valid
```

```
DPchecker::test_validate_schema()
```

11. If everything looks good, write the `my_eml2` R object back to XML, remembering to use the NPS naming convention for metadata files (`*_metadata.xml`).

```
write_eml(my_meta2, "YYYYMMDD_SET_metadata.xml")
```

Compile the Metadata Creation Package

Because SET data are published annually for this protocol, most of the intermediate ‘helper’ files produced in the steps described above can be reused during subsequent publishing events. To make these files easier to find and reuse, they are bundled together and published on DataStore as a ‘Metadata Creation Package.’ Each Metadata Creation Package contains the versions of the text templates, EMLcreation, and EMLeditor scripts used to generate the Data Package associated with it.

The goal is to have a corresponding Metadata Creation Package for each actual Data Package published.

Metadata Creation Packages are generated once a satisfactory .xml metadata file has been produced by simply zipping up this publication round's version of the text templates, EMLcreation and EMLeditor scripts. The section below discusses how to publish them. See Figure 10 for an example of what needs to be included.

Publish the Data Package on DataStore

Upload Data Package to DataStore

To upload the data package files to DataStore:

1. Go back to the IRMA record you created for the data package in the 'Create Draft DataStore Records' section.
2. Go to the 'Files and Links' tab. Upload the six .csv files with the current date and the metadata (.xml) file.
3. Update the DOI (object identifier field) record number.
4. Send the DataStore reference number to the Central Support Office (Judd Patterson) for review.
5. After both IRMA records are active, make sure to go back and version the newest file.
 - a. Open the old record
 - b. Click 'Actions' → 'Edit'
 - c. Go to the 'Version History' tab
 - d. Click '+Add'
 - e. Use the NEW DataStore package reference number as the new version
6. Add the NEW metadata creation package as a cross reference.

Upload the Metadata Creation Package to DataStore

To upload the Metadata Creation Package to DataStore, repeat steps 1–6 above for the IRMA record you created earlier for the Metadata Creation Package. Add the NEW Data Package record number as a cross reference.

Literature Cited

- Baker, R., and J. Patterson. 2023. NPSdataverse: Tools and packages for data and metadata manipulation. R package version 0.0.0.9000. Available at: <https://github.com/RobLBaker/NPSdataverse> (last accessed October 2023)
- Grolemund, G., and H. Wickham. 2011. Dates and times made easy with lubridate. *Journal of Statistical Software*, 40(3), 1–25. Available at: <https://www.jstatsoft.org/v40/i03/> (last accessed October 2023).
- Jones, M. B., M. O’Brien, B. Mecum, C. Boettiger, M. Schildhauer, M. Maier, T. Whiteaker, S. Earl, and S. Chong. 2019. Ecological Metadata Language version 2.2.0. KNB Data Repository. Available at: [doi:10.5063/F11834T2](https://doi.org/10.5063/F11834T2) (last accessed October 2023).
- National Park Service (NPS). 2023. Surface Elevation Table (SET) Database Documentation (Project). Available at: <https://irma.nps.gov/DataStore/Reference/Profile/2294781> (last accessed October 2023).
- R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Available at: <https://www.R-project.org/> (last accessed October 2023)
- RStudio Team (2019). RStudio: Integrated Development for R. RStudio, Inc., Boston, Massachusetts. Available at: <https://www.posit.co/> (last accessed October 2023).
- Smith, Colin. 2022. EMLassemblyline: A tool kit for building EML metadata workflows. Available at: <https://github.com/EDIorg/EMLassemblyline>, <https://ediorg.github.io/EMLassemblyline> (last accessed October 2023).
- Wickham, H., M. Averick, J. Bryan, W. Chang, L. D. McGowan, R. François, G. Grolemund, A. Hayes, L. Henry, J. Hester, M. Kuhn, T. L. Pedersen, E. Miller, S. N. Bache, K. Müller, J. Ooms, D. Robinson, D. P. Seidel, V. Spinu, K. Takahashi, D. Vaughan, C. Wilke, K. Woo, and H. Yutani. 2019. Welcome to the tidyverse. *Journal of Open Source Software* 4(43):1686. Available at: <https://doi.org/10.21105/joss.01686> (last accessed October 2023).
- Wickham, H. 2022. stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.5.0. Available at: <https://CRAN.R-project.org/package=stringr> (last accessed October 2023).
- Wickham, H., J. Hester, W. Chang, and J. Bryan. 2022. devtools: Tools to make developing R packages easier. Available at: <https://devtools.r-lib.org/> or <https://github.com/r-lib/devtools> (last accessed October 2023).

Wright, W., A. Kozlowski, J. Patterson, A. Okunoye. 2022a. Monitoring Coastal Wetland Elevation Data Management and Database Technical Documentation—Version 2.1. Southeast Coast Network Standard Operating Procedure NPS/SECN/SOP—2.3.07. Southeast Coast Network, Athens, Georgia. Available at: <https://irma.nps.gov/DataStore/Reference/Profile/2294179> (last accessed October 2023).

Wright, W., D. Jones, and A. Kozlowski. 2022b. Surface Elevation Table monitoring export views data pull (R Script). Available at: <https://irma.nps.gov/DataStore/Reference/Profile/2294820> (last accessed October 2023).

Wright, W., K. Montenegro, A. Kozlowski, J. Patterson, and M. Prats. 2023. Coastal wetland elevation database user manual and technical documentation—Version 2.0. Southeast Coast Network Standard Operating Procedure NPS/SECN/SOP—2.3.06. National Park Service, Inventory and Monitoring Division. Athens, Georgia. Available at: <https://irma.nps.gov/DataStore/Reference/Profile/2293214> (last accessed October 2023).