ATTAINS Upload File Creation

Tetra Tech

March 27, 2024

## Introduction

The following user guide pertains to the [*create\_attains\_Assessment.R*](https://github.com/KateriSalk/Alaska_IR_Automation/blob/main/Code/7_Results/create_attains_Assessment.R) and [*create\_attains\_AU.R*](https://github.com/KateriSalk/Alaska_IR_Automation/blob/main/Code/7_Results/create_attains_AU.R) scripts developed by Tetra Tech for the Alaska Department of Environmental Conservation (AK DEC). This guide assumes that the user has relatively recent versions of R and [RStudio](https://posit.co/download/rstudio-desktop/) installed and that they are familiar with the R coding language. Please direct any questions regarding the usage of these scripts to Amber Bethe Crawford, AK DEC (amber.crawford@alaska.gov).

## Purpose

The purpose of the *create\_attains\_Assessment.R* script is to take the outputs of previous scripts and create required the ATTAINS Assessment upload csv files. Similarly, the purpose of the *create\_attains\_AU.R* script is to take the outputs of previous scripts and create required the ATTAINS Assessment Unit upload csv files. The csv outputs of each script are reflective of the ATTAINS template version 2022-07-18.

## Required Packages

The required packages for the *data\_processing.R* code are [*tidyverse*](https://www.tidyverse.org/) and [*readxl*](https://cran.r-project.org/web/packages/readxl/index.html)*.* Installation for the *tidyverse* package can be found in the Data Pull Guide. Below is a code chunk that demonstrates how to install and load these packages.

####Load Packages####  
library(tidyverse)  
  
#install.packages('readxl')  
library(readxl)

## Required Inputs

This code has specific exterior files that are required for it to run. The inputs are listed below. All input files are loaded at the beginning of the script.

| File Name | Purpose | Source |
| --- | --- | --- |
| WQ\_metadata\_trimmed\_with\_data\_sufficiency\_20240117.csv | Provides necessary metadata for ATTAINS csv outputs  \*\*For Assessment Units Only | *data\_processing.R* |
| AUID\_crosswalk.csv | Connects previous AU IDs to current AU IDs. Required to connect to previous year ATTAINS information | AK DEC |
| ATTAINS\_AK\_Asessments\_DataDownload\_20240126.xlsx | Provides information from previous years ATTAINS  \*For Assessment Only | ATTAINS |
| ATTAINS\_AK\_AsessmentUnits\_DataDownload\_20240126.xlsx | Provides information from previous years ATTAINS  \*\*For Assessment Units Only | ATTAINS |
| WQ\_data\_trimmed\_long\_withAU20240117.csv | Provides sample information such as date range that is required in the ATTAINS csv outputs | *data\_processing.R* |
| categorized\_aus\_20240222.csv | Provides necessary metadata for ATTAINS csv outputs | *categorize\_au.R* |

## Combine all AU IDs

In order to connect the current IR cycle ATTAINS information to previous years, we need to use the *AUID\_crosswalk.csv* information. The next section of code pulls previous ATTAINS information with the current AU IDs, previous AU IDs, and historical AU IDs. None of the AU ID numbers are overridden in this code. This information is then combined together to provide previous metadata to the current year’s upload. The following code shows the process in *create\_attains\_Assessment.R*.

#Combine sample data with AUID crosswalk  
data\_category\_AUID\_added <- categorized\_aus %>%  
 left\_join(au\_id\_crosswalk, by = c('AUID\_ATTNS' = 'Active\_AUID'))  
  
  
#Join with previous AU ATTAINS by each AUID type  
data\_current\_AU <- data\_category\_AUID\_added %>%  
 mutate(assessmentUnitId = AUID\_ATTNS) %>%  
 right\_join(previous\_assessment\_attains, by = c('assessmentUnitId'))  
  
#Find retired AUs  
data\_retired\_AU <- data\_category\_AUID\_added %>%  
 mutate(assessmentUnitId = Retired\_AUID) %>%  
 filter(!is.na(assessmentUnitId)) %>%  
 right\_join(previous\_assessment\_attains, by = c('assessmentUnitId'))  
  
#Find historical AUs  
data\_historical\_AU <- data\_category\_AUID\_added %>%  
 mutate(assessmentUnitId = Historical\_AUID) %>%  
 filter(!is.na(assessmentUnitId)) %>%  
 right\_join(previous\_assessment\_attains, by = c('assessmentUnitId'))  
  
#Combine all together  
data\_all\_AUs1 <- data\_current\_AU %>%  
 rbind(data\_retired\_AU) %>%  
 rbind(data\_historical\_AU) %>%  
 unique()  
  
#Find AUs not in previous ATTAINS  
data\_current\_AU\_not\_listed <- data\_category\_AUID\_added %>%  
 mutate(assessmentUnitId = AUID\_ATTNS) %>%  
 full\_join(previous\_assessment\_attains, by = c('assessmentUnitId')) %>%  
 filter(!assessmentUnitId %in% data\_current\_AU$assessmentUnitId) %>%  
 filter(!assessmentUnitId %in% data\_retired\_AU$assessmentUnitId) %>%  
 filter(!assessmentUnitId %in% data\_historical\_AU$assessmentUnitId)  
  
#Bind rows together to ensure all current AUs represented  
#This becomes the starting point for each csv export process  
data\_all\_AUs <- data\_all\_AUs1 %>%   
 rbind(data\_current\_AU\_not\_listed)

\*For create\_attains\_AU.R, the steps are the same, but the *data\_sufficiency* variable replaces the *categorized\_aus* variable.

## Assessment Outputs

The following sections outline the process to create the csvs required for ATTAINS Assessment uploads. The output of each section is exported as a csv to a folder location of the user’s choosing.

### Assessments

The ‘Assessments’ csv output requires Assessment Unit ID, Agency, and Cycle Last Assessed. The following code include all the optional columns as NA values.

assessments <- data\_all\_AUs %>%  
 select(AUID\_ATTNS, organizationId, reportingCycle, Use\_Category, Use) %>%  
 mutate(ASSESSMENT\_COMMENT = Use,  
 AGENCY\_CODE = 'S',  
 YEAR\_LAST\_MONITORED = '2023',  
 ASSESSMENT\_RATIONALE = NA,  
 TROPHIC\_STATUS = NA) %>%  
 rename(ASSESSMENT\_UNIT\_ID = AUID\_ATTNS,   
 CYCLE\_LAST\_ASSESSED = reportingCycle, #Pulling from previous year's ATTAINS  
 ASSESSMENT\_UNIT\_STATE\_IR\_CAT = Use\_Category) %>%  
 unique() %>%  
 select(!Use) %>%  
 select(ASSESSMENT\_UNIT\_ID, AGENCY\_CODE, CYCLE\_LAST\_ASSESSED, YEAR\_LAST\_MONITORED,  
 ASSESSMENT\_UNIT\_STATE\_IR\_CAT, ASSESSMENT\_COMMENT, ASSESSMENT\_RATIONALE,  
 TROPHIC\_STATUS)

### Uses

The ‘Uses’ csv output requires the Assessment Unit ID, Use Name, Use Support, and Agency. The Use Support (USE\_ATTAINMENT\_CODE) column required a character version of the state-level numeric system. This change is in the *case\_when* statement. Some optional columns were filled out including: Monitoring Start Date, Monitoring End Date, and Use Level State IR Category Code. The following code includes the rest of the optional columns as NA values.

monitoring\_dates <- samples %>%  
 select(AUID\_ATTNS, ActivityStartDate) %>%  
 group\_by(AUID\_ATTNS) %>%  
 reframe(AUID\_ATTNS = AUID\_ATTNS,  
 USE\_MONITORING\_START = min(ActivityStartDate),  
 USE\_MONITORING\_END = max(ActivityStartDate)) %>%  
 unique()  
  
uses <- data\_all\_AUs %>%  
 select(AUID\_ATTNS, AUID\_ATTNS, Use, Use\_Category) %>%  
 left\_join(monitoring\_dates, by = 'AUID\_ATTNS') %>%  
 unique() %>%  
 rename(ASSESSMENT\_UNIT\_ID = AUID\_ATTNS,  
 USE\_NAME = Use) %>%  
 mutate(USE\_ATTAINMENT\_CODE = case\_when(Use\_Category == 5 ~  
 "N", #Not supporting  
 Use\_Category == 2 ~  
 "F", #Fully supporting  
 Use\_Category == 3 ~  
 "I", #Insufficient Information  
 T ~  
 "X"), #Not assessed  
 USE\_AGENCY\_CODE = "S",  
 USE\_TREND = NA,  
 USE\_THREATENED = NA,  
 USE\_ASMT\_BASIS = NA,  
 USE\_ASMT\_DATE = NA,  
 USE\_ASSESSOR\_NAME = NA,  
 USE\_COMMENT = NA,  
 USE\_STATE\_IR\_CAT = Use\_Category,  
 USE\_ORG\_QUALIFIER\_FLAG = NA) %>%  
 select(!Use\_Category) %>%  
 select(ASSESSMENT\_UNIT\_ID, USE\_NAME, USE\_AGENCY\_CODE, USE\_TREND, USE\_THREATENED,  
 USE\_ASMT\_BASIS, USE\_MONITORING\_START, USE\_MONITORING\_END, USE\_ASMT\_DATE,  
 USE\_ASSESSOR\_NAME, USE\_COMMENT, USE\_STATE\_IR\_CAT,   
 USE\_ORG\_QUALIFIER\_FLAG)

### Assessment Types

The ‘Assessments Types’ csv output is not required. Assessment Unit ID, Use Name, and Type were provided anyway. The Confidence column is optional and assigned a value of NA.

assessment\_types <- data\_all\_AUs %>%  
 select(AUID\_ATTNS, Use, assessmentTypes) %>%  
 rename(ASSESSMENT\_UNIT\_ID = AUID\_ATTNS,  
 USE\_NAME = Use,  
 USE\_ASMT\_TYPE = assessmentTypes) %>% #PULL FROM PREVIOUS ATTAINS assessmentTypes  
 mutate(USE\_ASMT\_CONFIDENCE = NA) %>%  
 unique()

### Parameters

The ‘Parameters’ csv output requires Assessment Unit ID, Parameter Name, Associated Use, Parameter Status, and Attainment. The following code include all the optional columns as NA values.

parameters <- data\_all\_AUs %>%  
 select(AUID\_ATTNS, TADA.CharacteristicName, Use, Use\_Category, Individual\_Category) %>%  
 mutate(PARAM\_STATUS\_NAME = case\_when(Use\_Category == 5 ~ #CATEGORIES ARE UNCLEAR  
 "N",   
 Use\_Category == 2 ~  
 "Meeting Criteria",   
 Use\_Category == 3 ~  
 "Insufficient Information",   
 T ~  
 "X"),  
 PARAM\_TREND = NA,  
 PARAM\_COMMENT = NA,  
 PARAM\_AGENCY\_CODE = NA,  
 PARAM\_POLLUTANT\_INDICATOR = NA, #Should be Y/N  
 PARAM\_YEAR\_LISTED = NA,  
 PARAM\_TARGET\_TMDL\_DATE = NA,  
 PARAM\_EXPECTED\_TO\_ATTAIN = NA,  
 PARAM\_PRIORITY\_RANKING = NA,   
 PARAM\_CONSENT\_DECREE\_CYCLE = NA,  
 PARAM\_ALT\_LISTING\_ID = NA,  
 PARAM\_ORG\_QUALIFIER\_FLAG = NA,  
 PARAM\_DELISTING\_REASON = NA,  
 PARAM\_DELISTING\_COMMENT = NA,  
 PARAM\_DELISTING\_AGENCY = NA) %>%  
 rename(ASSESSMENT\_UNIT\_ID = AUID\_ATTNS,  
 PARAM\_NAME = TADA.CharacteristicName,  
 PARAM\_USE\_NAME = Use,  
 PARAM\_ATTAINMENT\_CODE = Individual\_Category,  
 PARAM\_STATE\_IR\_CAT = Use\_Category) %>%  
 unique() %>%  
 select(ASSESSMENT\_UNIT\_ID, PARAM\_NAME, PARAM\_USE\_NAME, PARAM\_STATUS\_NAME,  
 PARAM\_ATTAINMENT\_CODE, PARAM\_TREND, PARAM\_COMMENT, PARAM\_AGENCY\_CODE,  
 PARAM\_POLLUTANT\_INDICATOR, PARAM\_YEAR\_LISTED, PARAM\_TARGET\_TMDL\_DATE,  
 PARAM\_EXPECTED\_TO\_ATTAIN, PARAM\_PRIORITY\_RANKING,  
 PARAM\_CONSENT\_DECREE\_CYCLE, PARAM\_ALT\_LISTING\_ID,   
 PARAM\_ORG\_QUALIFIER\_FLAG, PARAM\_DELISTING\_REASON,   
 PARAM\_DELISTING\_COMMENT, PARAM\_DELISTING\_AGENCY)

Optional Outputs

The ‘Assessment Method Types’, ‘Seasons’, ‘Sources’, and ‘Associated-Actions’ csv outputs were not included since there was no required information and no evidence of AK DEC providing these in the past.

## Assessment Unit Outputs

The following sections outline the process to create the csvs required for ATTAINS Assessment Unit uploads. The output of each section is exported as a csv to a folder location of the user’s choosing.

### Assessment Units

The ‘Assessment Units’ csv output requires Assessment Unit ID, Name, State, Agency, and Location Description. Use Class is provided, despite being an optional column. The Comment column is assigned NA values.

assessment\_units <- data\_all\_AUs %>%  
 rename(ASSESSMENT\_UNIT\_ID = AUID\_ATTNS,  
 USE\_CLASS\_NAME = AU\_Type,  
 ASSESSMENT\_UNIT\_NAME = assessmentUnitName,  
 LOCATION\_DESCRIPTION = locationText) %>%  
 mutate(ASSESSMENT\_UNIT\_STATE = 'AK',  
 ASSESSMENT\_UNIT\_AGENCY = 'S', #S for state agency  
 ASSESSMENT\_UNIT\_COMMENT = NA) %>% #ASK ABOUT COMMENTS  
 select(ASSESSMENT\_UNIT\_ID, ASSESSMENT\_UNIT\_NAME, ASSESSMENT\_UNIT\_STATE,  
 ASSESSMENT\_UNIT\_AGENCY, ASSESSMENT\_UNIT\_COMMENT, LOCATION\_DESCRIPTION,  
 USE\_CLASS\_NAME) %>% #Change order to reflect template  
 unique()   
  
#Some AUs have duplicates from having use class and some having NA  
duplicate\_values\_na <- assessment\_units %>%   
 group\_by\_at(vars(-USE\_CLASS\_NAME)) %>% #Find duplicate rows based on one having  
 #a use class and another having NA  
 filter(n() > 1) %>%  
 #filter out NA values   
 filter(is.na(USE\_CLASS\_NAME))  
  
#Remove all NA duplicated values in the assessment  
combine\_assessment\_units <- assessment\_units %>%  
 anti\_join(duplicate\_values\_na, by = c('ASSESSMENT\_UNIT\_ID', 'ASSESSMENT\_UNIT\_NAME', 'ASSESSMENT\_UNIT\_STATE',  
 'ASSESSMENT\_UNIT\_AGENCY', 'ASSESSMENT\_UNIT\_COMMENT', 'LOCATION\_DESCRIPTION',  
 'USE\_CLASS\_NAME'))

### Water Types

The ‘Water Types’ csv output requires Assessment Unit ID, Water Type, Size, and Water Size Units. Size Source and Source Scale were pulled from previous ATTAINS records. The Estimation Method was assigned as NA.

water\_types <- data\_all\_AUs %>%  
 select(AUID\_ATTNS, waterType, waterSize, waterSizeUnits, sizeSource, sourceScale) %>%  
 mutate(ESTIMATION\_METHOD = NA) %>%  
 rename(ASSESSMENT\_UNIT\_ID = AUID\_ATTNS,  
 WATER\_TYPE = waterType,  
 WATER\_SIZE = waterSize,  
 WATER\_UNIT = waterSizeUnits,  
 SIZE\_SOURCE = sizeSource,  
 SOURCE\_SCALE = sourceScale) %>%  
 select(ASSESSMENT\_UNIT\_ID, WATER\_TYPE, WATER\_SIZE, WATER\_UNIT, SIZE\_SOURCE,   
 ESTIMATION\_METHOD, SOURCE\_SCALE) %>% #Reorder columns  
 unique()

### Locations

The ‘Locations’ csv output is optional, but all the information is present in the analysis and previous ATTAINS records. All columns must be provided if this csv is to be uploaded to ATTAINS including Assessment Unit ID, Location Type Context, Location Type, and Location Value.

locations <- data\_all\_AUs %>%  
 select(AUID\_ATTNS, locationTypeCode, locationText, locationDescription) %>%  
 rename(ASSESSMENT\_UNIT\_ID = AUID\_ATTNS,  
 LOCATION\_TYPE\_CODE = locationTypeCode,  
 LOCATION\_TYPE\_CONTEXT = locationDescription,  
 LOCATION\_TEXT = locationText) %>%  
 unique()

### Monitoring Stations

The ‘Monitoring Stations’ csv output is optional, but all the information is present in the analysis and previous ATTAINS records. Three columns must be provided if this csv is to be uploaded to ATTAINS including Assessment Unit ID, Organization Identifier, and Monitoring Station ID. The Data Link column is optional and is not included in the output.

monitoring\_stations <- samples %>%  
 select(AUID\_ATTNS, ActivityConductingOrganizationText, MonitoringLocationIdentifier) %>%  
 unique() %>%  
 filter(!is.na(AUID\_ATTNS)) %>%  
 rename(ASSESSMENT\_UNIT\_ID = AUID\_ATTNS,  
 MS\_ORG\_ID = ActivityConductingOrganizationText,  
 MS\_LOCATION\_ID = MonitoringLocationIdentifier)