FNS-905 Summer Meals Report Standard Operating Procedure

Table of Contents

[1.0 – Set Up 2](#_Toc109027033)

[2.0 – Library 2](#_Toc109027034)

[3.0 – Data Import and Prep 2](#_Toc109027035)

[3.1 – Merge Lat & Long Coordinates 2](#_Toc109027036)

[3.2 – Combine Data1026 & Data1027 3](#_Toc109027037)

[3.3 – Format Dates 3](#_Toc109027038)

[3.4 – Address Corrections 4](#_Toc109027039)

[4.0 – Filter Out Inactive Sites By Date 6](#_Toc109027040)

[5.0 – Edit Checks 6](#_Toc109027041)

[5.1 – Check for Duplicate Rows 6](#_Toc109027042)

[5.2 – Save Data Rows with NAs in Required Fields (except lat & long) 7](#_Toc109027043)

[5.3 – Save Data Rows With Missing Long and Lat Coordinates & Save 7](#_Toc109027044)

[6.0 – Identify Sites Added & Sites Removed 7](#_Toc109027045)

[6.1 – Combine current and previous submissions and look for unique values 7](#_Toc109027046)

[6.2 – Filter Unique Values by Submission Date 8](#_Toc109027047)

[6.2.1 – Sites added are indicated by unique values in the current submission. 8](#_Toc109027048)

[6.2.2 – Sites removed are indicated by unique values in the previous submission 8](#_Toc109027049)

[7.0 – Edit Check 8](#_Toc109027050)

[8.0 – Prepare to Save 9](#_Toc109027051)

[8.1 – Add spaces to site numbers to prevent data from being converted to dates in Excel. 9](#_Toc109027052)

[8.2 – Reformat Column Headers For Viewing 9](#_Toc109027053)

[9.0 – Create the FNS-905 Report 9](#_Toc109027054)

[9.1 – Create Workbook 9](#_Toc109027055)

[9.2 – Add Worksheets to Workbook and Assign Names 9](#_Toc109027056)

[9.3 – Write Data to Worksheets 10](#_Toc109027057)

[9.4 – Save All Four Worksheets to One Workbook 10](#_Toc109027058)

[10.0 – Save operational Changes (last seven days) 10](#_Toc109027059)

# 1.0 – Set Up

#rm(list = ls(all.names = T)) #clear all objects from environment.  
options(max.print=100) #limit output.  
start.time <- Sys.time() # Log start time.   
options(scipen=999) #prohibit use of scientific notation  
gc() #free up memory and report the memory usage.  
curr\_date <- as.numeric(format(Sys.Date(), "%Y%m%d")) # Format date as numeric  
#date <- as.Date(date) # Use this code if date format is to be kept  
prior\_date <- (curr\_date - 9)  
prior\_date

# 2.0 – Library

library(janitor)  
library(tidyverse)  
library(openxlsx)  
library(readr)  
library(readxl)

# 3.0 – Data Import and Prep

## 3.1 – Merge Lat & Long Coordinates

Note: When transitioning between school years, it may be necessary download both the new SY & the previous SY of Data1027 & Data1026.

# Import Previous submission, clean names, remove extraneous columns  
previous\_submission <- read\_excel(paste0("P:/1\_Internal/Reports/Summer/USDA 543a\_FNS 905/2022\_post COVID waivers/USDA FNS 905 Submission 20220708.xlsx")) %>%   
 clean\_names %>%  
 select(-c(so\_o:map\_y))  
  
# Import both sheets of the Geocode Master List, clean names, rename long and lat to  
# match FNS-905 Template, remove columns.  
sso <- read\_excel(  
 "P:/1\_Internal/Reports/Summer/USDA 543a\_FNS 905/Geocode Master List\_CURRENT.xlsx",   
 sheet = 1) %>%   
 clean\_names %>%  
 rename(soo\_x = latitude,  
 soo\_y = longitude) %>%  
 rename(state\_site\_id\_number = site\_number) %>% # Rename unique ID in order to merge()  
 select(state\_site\_id\_number, soo\_x, soo\_y)  
  
sfsp <- read\_excel(  
 "P:/1\_Internal/Reports/Summer/USDA 543a\_FNS 905/Geocode Master List\_CURRENT.xlsx",   
 sheet = 2) %>%   
 clean\_names %>%  
 rename(soo\_x = latitude,  
 soo\_y = longitude) %>%  
 rename(state\_site\_id\_number = site\_number) %>% # Rename unique ID in order to merge()  
 select(state\_site\_id\_number, soo\_x, soo\_y)  
  
# Import Data1026 & Data1027, latin encoding, clean names, merge long and lat coordinates  
data1026\_sfsp <- read\_csv(  
 "H:/Downloads/Data1026 (70).csv", locale = locale(encoding = "Latin1")) %>%  
 clean\_names %>%  
 select(site\_name:state\_sponsor\_id\_number, last\_modified\_date) %>%  
 merge(sfsp, by = "state\_site\_id\_number", all.x = T)  
  
data1027\_sso <- read\_csv(  
 "H:/Downloads/Data1027 - 2022-07-15T091136.119.csv", locale = locale(encoding = "Latin1")) %>%  
 clean\_names %>%  
 select(site\_name:state\_sponsor\_id\_number, last\_modified\_date) %>%  
 relocate(last\_modified\_date) %>%  
 merge(sso, by = "state\_site\_id\_number", all.x = T)  
  
# Note: When transitioning between school years, it may be necessary download both  
# the new SY & the previous SY of Data1027 & Data1026. If the previous SY has sites  
# with end dates that are in the future, check to see if those sites are in the current  
# year's data. If not, both schools years should be imported. After importing both  
# years, combine the data.  
  
# For Example:  
#data1026 <- rbind(data1026 (1), data1026 (2))  
#data1027 <- rbind(data1027 (1), data1027 (2))

## 3.2 – Combine Data1026 & Data1027

# Combine Data1026 & Data1027  
all<-rbind(data1027\_sso, data1026\_sfsp)

## 3.3 – Format Dates

# Format Dates  
all$end\_date\_mm\_dd\_yy <- format(as.Date(all$end\_date\_mm\_dd\_yy, "%m/%d/%Y"))  
all$start\_date\_mm\_dd\_yy <- format(as.Date(all$start\_date\_mm\_dd\_yy, "%m/%d/%Y"))  
all$last\_modified\_date <- format(as.Date(all$last\_modified\_date, "%m/%d/%Y"))

## 3.4 – Address Corrections

# RICH JR HIGH/NRES   
  
all[which(all$state\_site\_id\_number == "24-304"), grep("site\_name", colnames(all))]  
all[which(all$site\_name == "RICH JR HIGH/NRES"), grep("state\_site\_id\_number", colnames(all))]  
all[which(all$state\_site\_id\_number == "24-304"), grep("site\_address1", colnames(all))]  
all[which(all$state\_site\_id\_number =="24-304"), grep("site\_address1", colnames(all))] <- "54 E 100 S"  
all[which(all$state\_site\_id\_number == "24-304"), grep("site\_address1", colnames(all))]  
  
# NORTH SUMMIT HIGH  
all[which(all$state\_site\_id\_number == "21-704"), grep("site\_name", colnames(all))]  
all[which(all$site\_name == "NORTH SUMMIT HIGH"), grep("state\_site\_id\_number", colnames(all))]  
all[which(all$state\_site\_id\_number == "21-704"), grep("site\_address1", colnames(all))]  
all[which(all$state\_site\_id\_number =="21-704"), grep("site\_address1", colnames(all))] <- "53 S 100 E"  
all[which(all$state\_site\_id\_number == "21-704"), grep("site\_address1", colnames(all))]  
  
# NORTH SUMMIT ELEMENTARY  
all[which(all$state\_site\_id\_number == "21-110"), grep("site\_name", colnames(all))]  
all[which(all$site\_name == "NORTH SUMMIT ELEMENTARY"), grep("state\_site\_id\_number", colnames(all))]  
all[which(all$state\_site\_id\_number == "21-110"), grep("site\_address1", colnames(all))]  
all[which(all$state\_site\_id\_number =="21-110"), grep("site\_address1", colnames(all))] <- "240 S BEACON DR"  
all[which(all$state\_site\_id\_number == "21-110"), grep("site\_address1", colnames(all))]  
  
# NORTH SUMMIT MIDDLE  
all[which(all$state\_site\_id\_number == "21-304"), grep("site\_name", colnames(all))]  
all[which(all$site\_name == "NORTH SUMMIT MIDDLE"), grep("state\_site\_id\_number", colnames(all))]  
all[which(all$state\_site\_id\_number == "21-304"), grep("site\_address1", colnames(all))]  
all[which(all$state\_site\_id\_number =="21-304"), grep("site\_address1", colnames(all))] <- "76 S 100 E"  
all[which(all$state\_site\_id\_number == "21-304"), grep("site\_address1", colnames(all))]  
  
# BLUE PEAK HIGH  
all[which(all$state\_site\_id\_number == "30-740"), grep("site\_name", colnames(all))]  
all[which(all$site\_name == "BLUE PEAK HIGH"), grep("state\_site\_id\_number", colnames(all))]  
all[which(all$state\_site\_id\_number == "30-740"), grep("site\_address1", colnames(all))]  
all[which(all$state\_site\_id\_number =="30-740"), grep("site\_address1", colnames(all))] <- "211 South Tooele Blvd."  
all[which(all$state\_site\_id\_number == "30-740"), grep("site\_address1", colnames(all))]  
  
# NAA TSIS' AAN COMMUNITY SCHOOL  
all[which(all$state\_site\_id\_number == "Q8-1"), grep("site\_name", colnames(all))]  
all[which(all$state\_site\_id\_number == "Q8-1"), grep("site\_address1", colnames(all))]  
all[which(all$state\_site\_id\_number == "Q8-1"), grep("site\_city", colnames(all))]  
all[which(all$state\_site\_id\_number == "Q8-1"), grep("site\_city", colnames(all))] <- "Tonalea"  
all[which(all$state\_site\_id\_number == "Q8-1"), grep("site\_zip", colnames(all))]  
all[which(all$state\_site\_id\_number == "Q8-1"), grep("site\_zip", colnames(all))] <- 86044  
all[which(all$state\_site\_id\_number == "Q8-1"), grep("site\_address1", colnames(all))] <- "24 Highway 98"  
all[which(all$state\_site\_id\_number == "Q8-1"), grep("site\_address1", colnames(all))]  
  
# GROUSE CREEK SCHOOL  
all[which(all$site\_name == "GROUSE CREEK SCHOOL"), grep("state\_site\_id\_number", colnames(all))]  
all[which(all$state\_site\_id\_number == "03-140"), grep("site\_name", colnames(all))]  
all[which(all$state\_site\_id\_number == "03-140"), grep("site\_address1", colnames(all))]  
all[which(all$state\_site\_id\_number == "03-140"), grep("site\_address1", colnames(all))] <- "76785 W 11900 N"  
all[which(all$state\_site\_id\_number == "03-140"), grep("site\_address1", colnames(all))]  
  
# BELKNAP ELEMENTARY  
all[which(all$state\_site\_id\_number == "02-104"), grep("site\_name", colnames(all))]  
all[which(all$state\_site\_id\_number == "02-104"), grep("site\_address1", colnames(all))]  
all[which(all$state\_site\_id\_number == "02-104"), grep("site\_address1", colnames(all))] <- "510 North 650 East"  
all[which(all$state\_site\_id\_number == "02-104"), grep("site\_address1", colnames(all))]  
  
#BEAVER HIGH  
all[which(all$state\_site\_id\_number == "02-704"), grep("site\_name", colnames(all))]  
all[which(all$state\_site\_id\_number == "02-704"), grep("site\_address1", colnames(all))]  
all[which(all$state\_site\_id\_number == "02-704"), grep("site\_address1", colnames(all))] <- "195 E CENTER"  
all[which(all$state\_site\_id\_number == "02-704"), grep("site\_address1", colnames(all))]  
  
# WELLINGTON ELEMENTARY  
all[which(all$state\_site\_id\_number == "05-154"), grep("site\_name", colnames(all))]  
all[which(all$state\_site\_id\_number == "05-154"), grep("site\_address1", colnames(all))]  
all[which(all$state\_site\_id\_number == "05-154"), grep("site\_address1", colnames(all))] <- "250 W 200 N"  
all[which(all$state\_site\_id\_number == "05-154"), grep("site\_address1", colnames(all))]  
  
# BLUFF ELEMENTARY   
all[which(all$site\_name == "BLUFF ELEMENTARY"), grep("state\_site\_id\_number", colnames(all))]  
all[which(all$state\_site\_id\_number == "25-108"), grep("site\_name", colnames(all))]  
all[which(all$state\_site\_id\_number =="25-108"), grep("site\_address1", colnames(all))]  
all[which(all$state\_site\_id\_number == "25-108"), grep("site\_address1", colnames(all))] <- "Old Main Highway 191"  
all[which(all$state\_site\_id\_number == "25-108"), grep("site\_address1", colnames(all))]

# 4.0 – Filter Out Inactive Sites By Date

active <- all %>%   
 relocate(state\_site\_id\_number, .before = state\_sponsor\_id\_number) %>%  
 filter(end\_date\_mm\_dd\_yy >= Sys.Date()+1) %>%   
 filter(start\_date\_mm\_dd\_yy <= Sys.Date()+30)  
  
refresh\_active <- active  
refresh\_all <- all  
  
nrow(all)  
nrow(active)

# 5.0 – Edit Checks

## 5.1 – Check for Duplicate Rows

active$b<-duplicated(active$state\_site\_id\_number, fromLast = F)  
active$a<-duplicated(active$state\_site\_id\_number, fromLast = T)  
active$ab<-paste0(active$a, active$b)  
  
all\_dups<-active[active$ab != "FALSEFALSE", ]  
active<-active[active$ab == "FALSEFALSE", ]  
  
active <- active %>%  
 select(-c(a, b, ab))  
  
nrow(all\_dups) # Number of duplicates

## 5.2 – Save Data Rows with NAs in Required Fields (except lat & long)

# Identify NAs in the following columns  
  
NAs1 <- active %>%  
 filter(is.na(site\_address1) | is.na(site\_zip) | is.na(site\_city)  
 | is.na(site\_state) | is.na(sponsoring\_organization)   
 | is.na(state\_site\_id\_number) | is.na(state\_sponsor\_id\_number)) %>%  
 write.xlsx("H:/Documents/NAs1.xlsx", overwrite=T)  
  
nrow(NAs1)  
  
# NA Locations (used for correcting NAs with code)  
which(is.na(active$site\_address1))  
which(is.na(active$site\_zip))  
which(is.na(active$site\_state))  
which(is.na(active$sponsoring\_organization))  
which(is.na(active$state\_site\_id\_number))  
which(is.na(active$state\_sponsor\_id\_number))  
which(is.na(active$soo\_x))  
which(is.na(active$soo\_y))

## 5.3 – Save Data Rows With Missing Long and Lat Coordinates & Save

# If NAs are present, save to file to add to Geocode Master List  
NAs2<-active[is.na(active$soo\_x),]  
  
NAs2<-NAs2 %>%  
 select(site\_program, sponsoring\_organization, state\_sponsor\_id\_number,   
 site\_name, state\_site\_id\_number, site\_address1, site\_city,   
 site\_state, site\_zip) %>%  
 write.xlsx("H:/Documents/NAs2.xlsx", overwrite=T)  
  
nrow(NAs2)

# 6.0 – Identify Sites Added & Sites Removed

## 6.1 – Combine current and previous submissions and look for unique values

current\_submission <- active %>%  
 select(-last\_modified\_date) %>%  
 mutate(submission = "current")  
  
previous\_submission <- active %>%  
 select(-last\_modified\_date) %>%  
 mutate(submission = "previous")  
  
# Combine  
added\_removed<-rbind(previous\_submission,current\_submission)  
  
# Identify unique values  
added\_removed$b<-duplicated(added\_removed$state\_site\_id\_number, fromLast = F)  
added\_removed$a<-duplicated(added\_removed$state\_site\_id\_number, fromLast = T)  
added\_removed$ab<-paste0(added\_removed$a, added\_removed$b)  
  
unique<-added\_removed[added\_removed$ab == "FALSEFALSE", ]  
  
nrow(unique)

## 6.2 – Filter Unique Values by Submission Date

### 6.2.1 – Sites added are indicated by unique values in the current submission.

### 6.2.2 – Sites removed are indicated by unique values in the previous submission

added <- unique %>%   
 select(site\_program, sponsoring\_organization, site\_name,   
 start\_date\_mm\_dd\_yy, end\_date\_mm\_dd\_yy, submission) %>%   
 filter(submission == "current") %>%  
 select(-submission)  
  
removed <- unique %>%   
 select(site\_program, sponsoring\_organization, site\_name,   
 start\_date\_mm\_dd\_yy, end\_date\_mm\_dd\_yy, submission) %>%   
 filter(submission == "previous")%>%  
 select(-submission)

# 7.0 – Edit Check

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  | + - = | *the # of sites in the previous submission* *the # of sites added* *the # of sites removed* *the # of sites in the current submission* |

# Make sure the number of sites corresponds with the number of sites added and removed  
nrow(added)  
nrow(removed)  
  
nrow(previous\_submission)+nrow(added)-nrow(removed)  
nrow(current\_submission)

# 8.0 – Prepare to Save

## 8.1 – Add spaces to site numbers to prevent data from being converted to dates in Excel.

active <- refresh\_active  
all <-refresh\_all  
  
# Add spaces to site numbers to prevent them from turning into dates  
all <- all %>%  
 mutate(state\_site\_id\_number = paste0(" ", state\_site\_id\_number, " ")) %>%  
 select(-last\_modified\_date)  
   
active <- active %>%  
 mutate(state\_site\_id\_number = paste0(" ", state\_site\_id\_number, " ")) %>%  
 select(-last\_modified\_date)

## 8.2 – Reformat Column Headers For Viewing

# Column Names for Added / Removed Sheets  
colnames(added) <- stringr::str\_to\_title(gsub("\_", " ",colnames(added)))  
colnames(removed) <- stringr::str\_to\_title(gsub("\_", " ",colnames(removed)))  
  
# Get the number of rows of each data frame and save it as a vector.  
nrow\_all<-as.vector(nrow(all))  
nrow\_active<-as.vector(nrow(active))  
nrow\_added<-as.vector(nrow(added))  
nrow\_removed<-as.vector(nrow(removed))

# 9.0 – Create the FNS-905 Report

## 9.1 – Create Workbook

# Create workbook  
wb <- createWorkbook()

## 9.2 – Add Worksheets to Workbook and Assign Names

# Add Worksheets to Workbook and Name  
addWorksheet(wb, paste0("Active Sites, (",nrow\_active,")"))  
addWorksheet(wb, paste0("All Sites, (",nrow\_all,")"))  
addWorksheet(wb, paste0("Added, (",nrow\_added,")"))  
addWorksheet(wb, paste0("Removed, (",nrow\_removed,")"))

## 9.3 – Write Data to Worksheets

# Write Data to Worksheets  
writeData(wb, sheet = paste0("Active Sites, (",nrow\_active,")"), x = active)  
writeData(wb, sheet = paste0("All Sites, (",nrow\_all,")"), x = all)  
writeData(wb, sheet = paste0("Added, (",nrow\_added,")"), x = added)  
writeData(wb, sheet = paste0("Removed, (",nrow\_removed,")"), x = removed)

## 9.4 – Save All Four Worksheets to One Workbook

# Save  
saveWorkbook(wb, paste0("H:/Documents/USDA FNS 905 Submission ", curr\_date, ".xlsx"), overwrite = T)

# 10.0 – Save operational Changes (last seven days)

active <- refresh\_active  
  
# Subset data  
modifications <- active %>%   
 filter(last\_modified\_date >= Sys.Date()-7) %>%  
 mutate(state\_site\_id\_number = paste0(" ", state\_site\_id\_number, " "))  
  
# Save to P Drive  
write.csv(modifications, paste0("P:/1\_Internal/Contracts, MOUs, State Agreements, Purchase Requests (non-conference)/DOH SFSP Health Inspections/2022/FNS 905 Operational Changes ", curr\_date, ".csv"),row.names=F, na="")  
  
end.time <- Sys.time() # Stop Timer   
duration <- end.time - start.time  
duration