COMMUNITY AND
TECHNICAL COLLEGES
Washington State Board

DATA DICTONARY

FVT/GE FINANCIAL AID DATA MERGE (R)

Revisions

Change reference	Date	Version
Initial documentation	05/27/2025	1.0

Contacts

Data Services
dataservices@sbctc.edu

Table of Contents

Financial Value Transparency/Gainful Employment (FVT/GE) (R)

```
######################
#
#  GE FVT Query and National Student Clearinghouse file combination script
#
#    original by Anna Fenner (Anna.Fenner@lwtech.edu) for 2024 GE FVT cycle
#    last update 05/08/2025
#
#     updates:   ----
#       05/01/2025 - generalization attempt - Kelsey Anderson (kanderson@highline.edu)
#       05/08/2025 - adjusted for query changes (residency added to COA) - ka
#
#      ----
#
#------------------
# Instructions:
#------------------
#    Requires running 7 queries in PS Query in xls format for one award year,
#    plus student level report from the NSC as a csv.
#    Column names are often hard-coded, and changes to file specifications will require script updates.
#
#    Queries:
#    DEV_FA_GE_FVT_GRANTS_UPDATE (once for Award Year = Y, once N)
#      cols:    EMPLID          (expandable with arrow)  ----
#              NID
#              Institution
#              Career
#              Acad Prog
#              Prog Descr
#              Acad Plan
#              Plan Descr
#              Plan Status
#              CIP Code
#              Degree
#              Title IV Aid?
#              Total Grants and Scholarships
#              Institutional Grants and Schol
#              Other, State or Private Grants
#              Private Loans
#    ----
#    DEV_FA_GE_FVT_COA_UPDATE (once for Award Year = Y, once N)
#      cols:    EMPLID          (expandable with arrow) ----
#              NID
#              Institution
#              Career
#              Acad Prog
#              Prog Descr
#           Acad Plan
#           Plan Descr
#           Status
#           CIP Code
#           Degree
#           Res Date
#           Residency
#           COA Books, Supplies, Equipment
#           COA Food and Housing
#           COA Total
#    ----
#    DEV_FA_GE_FVT_DEBT_UPDATE (once for Award Year = Y, once N)
#      cols:    EMPLID  (expandable with arrow)  ----
#           NID
#           Institution
#           Career
#           Acad Prog
#           Prog Descr
#           Acad Plan
#           Plan Descr
#           Plan Status
#           CIP Code
#           Degree
#           Charges
#           Credits
#           Total Institutional Debt
#           Total Tuition and Fees
#    ----
#    National Student Clearinghouse file specs: https://theclearinghouse.download/fvtge-
studentcohortspecs
#
######################

# Packages:
# ------------------
#Install required packages if needed:
#install.packages(c("readxl", "dplyr","openxlsx"))
#version
#Load libraries
library(readxl)
library(dplyr)
library(openxlsx)

# ------------------
# User Inputs:
#-------------------

# Enter Academic Year (YYYYYYYY format):
ayr <- '20232024'

# Enter the start date of the first term for which your campus has
# ctcLink FinAid data (YYYYMMDD format):
convert_date <- '20210111'
```

```
# Read in files - replace the string file path below with your local path and names:
# - run from PS query into xls format.
# - assumes file from the clearinghouse is in csv format.

## Set Working Directory if needed:
setwd(choose.dir())

## QCS_FE_GE_FVT_GRANTS (Academic Year Only = N)
grants_all <- read_excel('Downloaded_PS_Queries/DEV_FA_GE_FVT_GRANTS_UPDATE-2024_ALL.xlsx') # <---
UPDATE PATH HERE!
## QCS_FE_GE_FVT_GRANTS (Academic Year Only = Y)
grants_award <- read_excel('Downloaded_PS_Queries/DEV_FA_GE_FVT_GRANTS_UPDATE-2024_YO.xlsx') #
<--- UPDATE PATH HERE!

## QCS_FE_GE_FVT_COA (Academic Year Only = N)
coa_all <- read_excel('Downloaded_PS_Queries/DEV_FA_GE_FVT_COA_UPDATE-2024_ALL.xlsx') # <---
UPDATE PATH HERE!
## QCS_FE_GE_FVT_COA (Academic Year Only = Y)
coa_award <- read_excel('Downloaded_PS_Queries/DEV_FA_GE_FVT_COA_UPDATE-2024_YO.xlsx') # <---
UPDATE PATH HERE!

## QCS_FE_GE_FVT_DEBT (Academic Year Only = N)
debt_all <- read_excel('Downloaded_PS_Queries/DEV_FA_GE_FVT_DEBT_UPDATE-2024_ALL.xlsx') # <---
UPDATE PATH HERE!
## QCS_FE_GE_FVT_DEBT (Academic Year Only = Y)
debt_award <- read_excel('Downloaded_PS_Queries/DEV_FA_GE_FVT_DEBT_UPDATE-2024_YO.xlsx') # <---
UPDATE PATH HERE!


## Import National Student Clearinghouse TA/AA list
nch_all <- read.csv('NSC_Rosters/003781_AY20232024T_20241221.csv',  # <--- UPDATE PATH HERE!
                    check.names = FALSE,
                    colClasses = c("Institution Code (OPEID)"="character",
                                   "Credential Level" = "character",
                                   "Published Length of Program" = "character",
                                   "Weeks in Title IV Academic Year" = "character"))




# -------------------
# Functions:
# -------------------

#Clean function (Searches for row containing header value of Institution and sets it to header row)
clean_dataframe <- function(df) {

  # Find the first row that contains "ID" (either EMPLID or Student ID) by searching across columns
  headerStart <- min(unlist(apply(df[,colnames(df)], 2,
                              function(x)
                                which(unlist(lapply(gregexpr(pattern = 'ID$', x),
                                                    min)) >= 1))))
  colnames(df) <- df[headerStart,] # Set the second row as column names
  df <- df[-c(1:headerStart),] # Drop the first row rownames(df) <- NULL # Reset row names
  return(df)
}

#Function to remove decimal and convert CIP Code column to integer like NCH
remove_decimal_from_cip <- function(df, column_name = "CIP Code") {
  if(column_name %in% colnames(df)) {
    df[[column_name]] <- gsub("\\.", "", as.character(df[[column_name]]))
    df[[column_name]] <- as.integer(df[[column_name]])
  } else {
    print(paste("Column", column_name, "not found"))
  }
  return(df)
}

# Function to do basic file cleaning and data type changes for GE_FVT PS Query outputs
clean_values_in_GEFVT_dfs <- function(df) {

  df_name <- deparse(substitute(df))

  df <- remove_decimal_from_cip(df)
  df$`Award Year` <- ayr

  # add TA / AA Record type
  if(substr(df_name, nchar(df_name)-2, nchar(df_name)) == 'all') {
    df$'Record Type' <- 'TA'
    } else {
    df$'Record Type' <- 'AA'
    }

  # Converting SSN and Money columns to numeric
  # Note: NA coersion error on 'XXXXXXXXX' SSNs is fine
  df$NID <- as.integer(coalesce(as.numeric(df$NID), -1))
  colnames(df)[colnames(df) == "NID"] <- "Student Social Security Number"

  if(substr(df_name, 1, 3) == "coa") {
    df$'COA Books, Supplies, Equipment' <-
      round(as.numeric(df$'COA Books, Supplies, Equipment'), 0)

    df$'COA Food and Housing' <-
       round(as.numeric(df$'COA Food and Housing'), 0)

    df$'COA Total' <-
      round(as.numeric(df$'COA Total'), 0)
  }

  if(substr(df_name, 1, 4) == "debt") {

    df$'Total Institutional Debt' <-
    round(as.numeric(df$'Total Institutional Debt'), 0)

  df$'Total Tuition and Fees' <-
```

```
    round(as.numeric(df$'Total Tuition and Fees'), 0)
  }

  if(substr(df_name,1,5) == 'grant') {

    df$'Total Grants and Scholarships' <-
      round(as.numeric(df$'Total Grants and Scholarships'), 0)

    df$'Institutional Grants and Schol' <-
      round(as.numeric(df$'Institutional Grants and Schol'), 0)

    df$'Other, State or Private Grants' <-
      round(as.numeric(df$'Other, State or Private Grants'), 0)

    df$'Private Loans' <-
      round(as.numeric(df$'Private Loans'), 0)
  }

   return(df)
}


# ---------------------
# Cleaning Process:
# ---------------------

# Adjust header rows and drop extra lines at the beginning of the PS Query files
grants_all  <- clean_dataframe(grants_all)
grants_award <- clean_dataframe(grants_award)
coa_all <- clean_dataframe(coa_all)
coa_award <- clean_dataframe(coa_award)
debt_all <- clean_dataframe(debt_all)
debt_award <- clean_dataframe(debt_award)


# GE_FVT File Cleaning:
grants_all  <- clean_values_in_GEFVT_dfs(grants_all)
grants_award <- clean_values_in_GEFVT_dfs(grants_award)
coa_all <- clean_values_in_GEFVT_dfs(coa_all)
coa_award <- clean_values_in_GEFVT_dfs(coa_award)
debt_all <- clean_values_in_GEFVT_dfs(debt_all)
debt_award <- clean_values_in_GEFVT_dfs(debt_award)


# ---------------------
# Merging Datasets:
# ---------------------

# TA dataset  – note: Tf Title IV only was set to "Y" on PS Query,
#                     the number of records will be reduced to only
#                     those in the Grants file
merged_TA <- merge(grants_all, debt_all, all.x = TRUE ) %>%
            merge(., coa_all, all.x = TRUE )
```

```
# AA dataset -- note: Tf Title IV only was set to "Y" on PS Query,
#                       the number of records will be reduced to only
#                       those in the Grants file
merged_AA <- merge(grants_award, debt_award, all.x = TRUE ) %>%
  merge(., coa_award, all.x = TRUE )

# TA + AA together
merged_TAAA <- bind_rows(merged_TA, merged_AA)
rm(merged_AA, merged_TA)

# Merge into NSC
merged_data <- merge(nch_all, merged_TAAA,
                     by.x = c("Student Social Security Number", "CIP Code", "Record Type", "Award Year"),
                     by.y = c("Student Social Security Number", "CIP Code", "Record Type", "Award Year"),
                     all.x = TRUE)


#Finalizing report: copying columns from query data into the NCH columns

#Add residency data
merged_data <- merged_data %>%
  mutate(`Residency Tuition Status by State or District` =
           case_when(`Title IV Aid?` == 'N' ~ NA,
                     is.na(`Title IV Aid?`) ~ NA,
                     Residency == 'IS' ~ 'IS',
                     TRUE ~ 'OS'))

# Update monetary columns for 'TA' record type
merged_data <- merged_data %>%
  mutate(
        # Update Private Loans
        `Total Amount Student Received in Private Education Loans During Student's Entire Enrollment in the
Program` =
           case_when(`Record Type` == 'TA' & `Title IV Aid?` == 'Y' ~ `Private Loans`, TRUE ~ NA_real_ ),
        # Update Institutional Debt  -- NOTE: Negative Debt amounts are being set to zero here!
        `Total Amount of Institutional Debt During Student's Entire Enrollment in the Program` =
          case_when(`Record Type` == 'TA' & `Title IV Aid?` == 'Y' & `Total Institutional Debt` >0 ~ `Total
Institutional Debt`,
                    `Record Type` == 'TA' & `Title IV Aid?` == 'Y' & `Total Institutional Debt` <=0 ~ 0.0,
                    TRUE ~ NA_real_ ),
        # Update Tuition & Fees
        `Total Amount of Tuition & Fees Assessed During Student's Entire Enrollment in the Program` =
           case_when(`Record Type` == 'TA' & `Title IV Aid?` == 'Y' ~ `Total Tuition and Fees`, TRUE ~ NA_real_
),
        # Update 'COA Books, Supplies, Equipment'
        `Total Amount of Allowance for books, supplies, and equipment included in the student's title IV, HEA
COA During Student's Entire Enrollment in the Program` =
          case_when(`Record Type` == 'TA' & `Title IV Aid?` == 'Y' ~ `COA Books, Supplies, Equipment`, TRUE ~
NA_real_ ),
        # Update 'Institutional Grants and Scholarships'
        `Total Amount of Grants and Scholarships the student received During Student's Entire Enrollment in
the Program` =
          case_when(`Record Type` == 'TA' & `Title IV Aid?` == 'Y' ~ `Institutional Grants and Schol `, TRUE ~
NA_real_ )
        )

# Update monetary columns for 'AA' record type
merged_data <- merged_data %>%
  mutate(
    # Update Private Loans
    `Private Loans Amount`=
      case_when(`Record Type` == 'AA' & `Title IV Aid?` == 'Y' ~ `Private Loans`, TRUE ~ NA_real_ ),
    # Update 'COA Books, Supplies, Equipment'
    `Allowance for Books, Supplies, and Equipment` =
      case_when(`Record Type` == 'AA' & `Title IV Aid?` == 'Y' ~ `COA Books, Supplies, Equipment`, TRUE ~
NA_real_ ),
    # Update 'COA Housing and Food'
    `Allowance for Housing and Food` =
      case_when(`Record Type` == 'AA' & `Title IV Aid?` == 'Y' ~ `COA Food and Housing`, TRUE ~ NA_real_ ),
    # Update Total COA
    `Annual Cost of Attendance (COA)` =
      case_when(`Record Type` == 'AA' & `Title IV Aid?` == 'Y' ~ `COA Total`, TRUE ~ NA_real_ ),
    # Update Tuition & Fees
    `Tuition and Fees Amount for Award Year being Reported` =
      case_when(`Record Type` == 'AA' & `Title IV Aid?` == 'Y' ~ `Total Tuition and Fees`, TRUE ~ NA_real_ ),
    # Update 'Institutional Grants and Scholarships'
    `Institutional Grants and Scholarships` =
      case_when(`Record Type` == 'AA' & `Title IV Aid?` == 'Y' ~ `Institutional Grants and Schol`, TRUE ~
NA_real_ ),
    # Update Other state or private grants
    `Other State, Tribal, or Private Grants` =
      case_when(`Record Type` == 'AA' & `Title IV Aid?` == 'Y' ~ `Other, State or Private Grants`, TRUE ~
NA_real_ ),
  )

# Update Invalid flag for non-Title IV rows and rearrange a bit
# and updating data types for NSC upload specs
merged_data <- merged_data %>%
  mutate(`Invalid Flag` = case_when(`Title IV Aid?`== 'Y' ~ ' ',
                                    TRUE ~ 'T')) %>%
  select(EMPLID, `Student Social Security Number`, colnames(nch_all), everything()) %>%
  mutate(`Student Social Security Number` =  formatC(`Student Social Security Number`, width = 9, format
= "d", flag = "0"),
         `CIP Code` = formatC(`CIP Code`, width = 6, format = "d", flag = "0"),
         `College Student ID` = case_when(is.na(`College Student ID`) ~ NA_character_,
                                         TRUE ~ as.character(`College Student ID`))
         )

# Drop unneeded columns
merged_output <- merged_data %>%
  select(colnames(nch_all))


#Sorting values by 'Record Type'
merged_output <- merged_output %>% arrange(`Record Type`)
```

```
# ---------------------
# Pull records that might have legacy data:
# ---------------------
check_legacy <- merged_data %>%
  filter(`Program Enrollment Begin Date` < convert_date) %>%
  select(EMPLID, `Student Social Security Number`, colnames(nch_all), everything()) %>%
  mutate(Action = if_else(`Record Type` == 'AA',
                          'Check legacy for Aid, use annual values from queries, remove invalid flag, add residency',
                          'Check legacy for Aid, sum legacy values with totals from queries, remove invalid flag, add
residency'),
         .before = EMPLID)



# ---------------------
# Output Results Files:
# ---------------------

# Save the final merged data to Excel files
write.xlsx(merged_output, paste0("GE_FVT_", ayr, "_NCH_Output.xlsx"), rownames = FALSE)

# Output the final file with ALL columns (not just the NSC required ones)
write.xlsx(merged_data, paste0("GE_FVT_", ayr, "_ALL_COLUMNS_Output.xlsx"), rownames = FALSE)

# Save list of students to check for in Title IV Aid in Legacy
write.xlsx(check_legacy, paste0("GE_FVT_", ayr, "_Records_for_Legacy_Check.xlsx"), rownames = FALSE)


# Script Ends #
```

Washington State Board for Community and Technical Colleges