DAF User's Code Library

April 2024

This page has been left blank for double-sided copying.

Contents

Tables

Glossary

AB	Accelerated Benefits Demonstration
ADM	Awardee Data Mart
AIME	Average Indexed Monthly Earnings
BEST	Benefits Entitlement Services Team
BFW	Benefits forgone due to work
BIC	Beneficiary Identification Code
BMF	Budget Month Factor
BOAN	Beneficiary's Own Account Number
BOND	Benefit Offset National Demonstration
BOPD	Benefit Offset Pilot Demonstration
CAN	Claim Account Number
CDR	Continuing Disability Review
CER	Characteristics Extract Record 100% Field File
COLA	Cost-of-Living Adjustment
COSSN	Claimants Own Social Security Number
DAC	Disabled Adult Child
DAF	Disability Analysis File (previously known as TRF)
DBAD	Disabled Beneficiary and Dependents Extract
DCF	Disability Control File
DDS	Disability Determination Services
DER	Detailed Earnings Record
DI	Disability Insurance, also referred to as SSDI
DMG	Demographic component of the DAF
DRF	Disability Research File
DWB	Disabled Widow Beneficiaries
EDW	Enterprise Data Warehouse
EN	Employment Network (also called a TTW provider)
EPE	Extended Period of Eligibility
EVS	Enumeration Verification System
EXR	Expedited Reinstatement
FBR	Federal Benefit Rate
FCI	Federal Countable Income
FIPS	Federal Information Processing Standards (in reference to U.S. Census standardized codes for uniform identification of geographic entities)

FRA	Full Retirement Age
HI	Hospital Insurance (Medicare Part A)
HOPE	Homeless Outreach Projects and Evaluation Demonstration
HUN	Housed Under Number
ICD-9	International Classification of Diseases Coding Scheme
IPE	Individualized Plan for Employment, developed by State VR Agency
IRS	Internal Revenue Service
IRWE	Impairment-Related Work Expense
LAF	Ledger Account File
LAUS	Local Area Unemployment Statistics
MBR	Master Beneficiary Record
MEF	Master Earnings File
MHTS	Mental Health Treatment Study
MIE	Medical Improvement Expected
MO	Milestone + Outcomes payment system
MPR-EVS	Mathematica's EVS
NBS	National Beneficiary Survey
NSCF	National Survey of SSI Children and Families
NUMIDENT	Numerical Identification File
OIM	Office of Information Management
OO	Outcomes-Only payment system
PAN	Person's Account Number
PASS	Program to Achieve Self-Support
PHUS	Payment History Update System
PIA	Primary Insurance Amount
PIN	Personal Identification Number
POD	Promoting Opportunity Demonstration
POMS	SSA's Program Operations Manual System
PROMISE	Promoting Readiness of Minors in SSI
Provider	Service provider under TTW (also called an EN)
PUF	Public Use File
RECS	Race and Ethnicity Classification System
REMICS	Revised Management Information Counts System
RIB	Retirement Insurance Benefits
RMA	Retrospective Monthly Accounting

RSA	Rehabilitation Services Administration
RSA-911	RSA Case Service Report
SAIPE	Small Area Income and Poverty Estimates
SAS	Statistical Analysis Software, used to produce the DAF
SCWF	Standalone Companion Work File
SED	Supported Employment Demonstration
SER	Summary Earnings Record
SGA	Substantial Gainful Activity
SMI	Supplemental Medical Insurance (Medicare Part B)
SNAP	Supplemental Nutrition Assistance Program
SSN	Social Security Number
SSA	Social Security Administration
SSDI	Social Security Disability Insurance (also referred to as DI)
SSI	Supplemental Security Income
SSI-LF	SSI - Longitudinal File
SSR	Supplemental Security Record
STW	Suspension or termination of cash benefits for work
T2	Title II, the SSDI Program
T16	Title XVI, the SSI Program
TANF	Temporary Assistance for Needy Families
TCNEI	Total countable non-earned income
TKT	DAF component containing data related to TTW participation
TRF	Ticket Research File, now called the DAF
TTW	Ticket to Work
TWP	Trial Work Period
VR	Federal/State Vocational Rehabilitation program/agency
VRRMS	Vocational Rehabilitation Reimbursement Management System; data from this system is contained in the Payments component
YTD	Youth Transition Demonstration

Overview of DAF Documentation

The documentation for the DAF consists of the eleven volumes described below. Questions about these documents should be directed to ORDES.DAF@ssa.gov. All of these documents are available at https://www.ssa.gov/disabilityresearch/daf.html.

- **Volume 1: Getting Started with the DAF22**. Provides an overview of the structure and contents of the DAF and related linkable files.

- **Volume 2: Working with the DAF22**. Contains practical suggestions such as how to extract data and interpret blank or missing variables as well as more detailed information on DAF data marts and linkable files.

- **Volume 3: Tips for Conducting Analysis with the DAF22**. Contains suggestions for working with common research concepts in the DAF such as program participation, benefits paid versus benefits due, and constructed measures related to beneficiary work activity resulting in the loss of cash benefits.

- **Volume 4: Lists of DAF22 Variables**. Contains lists of new, changed, and deleted variables, as well as lists of variables by DAF component and analytic category.

- **Volume 5: DAF Variable Detail Pages**. Contains specifications for each DAF variable, including name, definition, data format, identification of the DAF component to which it belongs, data source, availability, and (where applicable) SAS code used to construct the variable.

- **Volume 6: Validating the DAF22 Against Other Sources**. Provides an explanation of validation methods and summary of validation results.

- **Volume 7: DAF22 Development History and Construction Methods**. Describes key changes in DAF construction methodology over time as well as a description of each step in the current year DAF construction process.

- **Volume 8: DAF22 Construction Workflow Charts and Task Tables**. Provides detailed information in both chart and table format on each step in the current year DAF construction process.

- **Volume 9: DAF22 Source File Descriptions**. Describes the administrative source files used to construct the DAF.

- **Volume 10: DAF22 Administrative Source File Documentation**. Contains documentation from SSA or other agencies on the administrative source files described in Volume 9.

- **Volume 11: DAF22 Construction Code**. Contains all SAS code used to construct the DAF.

- **Volume 12: DAF22 RSA Administrative Source File Documentation**. Contains a description of the processing of Rehabilitation Services Administration (RSA) data for linkage to the DAF, along with documentation from RSA on the RSA-911 files.

The following table provides specific locations for common research-related questions and issues.

In order to …	Refer to …
Get started with a research task	Volume 2, "Working with the DAF22," for information about selecting beneficiaries using finder files versus selection criteria
Identify what's changed in the latest version of the DAF	Volume 1, "Getting Started with the DAF22"
View lists of DAF variables	Volume 4, "Lists of DAF22 Variables"
Understand individual variable definitions, specifications, and value ranges	Volume 5, "DAF Variable Detail Pages"
Understand the structure of the DAF data files at a high level	Volume 1, "Getting Started with the DAF22"
Identify variables for a specific research task	Volume 4, "Lists of DAF22 Variables," for a list of variables contained within each DAF file and by analytic category
Understand the beneficiaries for which the DAF does and does not contain data	Volume 1, "Getting Started with the DAF22"
Identify administrative data sources for the DAF	Volume 9, "DAF22 Source File Descriptions"
Understand the linkage of the DAF to RSA-911 data and contents of the RSA files	Volume 12, "DAF22 RSA Administrative Source File Documentation"
Generate ideas for using the DAF more efficiently	Volume 1, "Getting Started with the DAF22" and Volume 2, "Working with the DAF22"
Find suggested ways to identify common research concepts in the DAF, such as calculating age of retirement, or disability title	Volume 3, "Tips for Conducting Analysis with the DAF22"
Understand what variables have changed in the most recent DAF	Volume 4, "Lists of DAF22 Variables"
Read about how information in the DAF is validated against other sources	Volume 6, "Validating the DAF22 Against Other Sources"

Introduction

To make the DAF more efficient and easier to use, we have developed SAS code for common analytical tasks run on DAF files. Researchers can use and modify this code as needed. In writing this code and putting together this code library, we aimed to accomplish three goals:

- select commonly used analysis tasks that may pose difficulty for many DAF users developing code on their own;

- write the code in such a way that it can be used as a base for accomplishing other similar tasks; and

- provide previously debugged and tested code to spare DAF researchers the need to develop, test, debug, and revise new code.

At this point, the DAF Users' Code Library includes code to complete the following tasks:

- determine whether a beneficiary is in current pay for either SSDI or SSI within a user-specified time period;

- categorize impairment codes into the groupings used in SSA's published statistics;

- determine whether a beneficiary has been suspended or terminated due to work within a user-specified time period; and

- Reorder N suffixed variables to be in a chronological order.

In addition to providing code, we specify the DAF components necessary to run the code, an example data step, the variables used in the program, and output files and variables created by the program. We expect the DAF Users' Code Library to grow over time, so please check back periodically.

I.   How to Determine the Number of Months a Beneficiary Was Eligible for SSI or SSDI Benefits Within a Specific Time Period

A.   Description

Every beneficiary in the DAF was eligible for either SSDI or SSI (or both) in at least one month in or after 1996. Determining whether a beneficiary was eligible for SSDI or SSI benefits in a given month or series of months is a central research task and precursor to many DAF analyses. The associated SAS code offers users an example of how to determine the number of months of eligibility during an inclusive time period using the LAF variable (LAFyymm) for SSDI and the payment status code variable (PSTAyymm) for SSI. A LAF value beginning with C or E indicates SSDI eligibility in that month and a PSTA value of C01, M01, or M02 indicates SSI eligibility in that month, as described in Volume 3 of the DAF documentation.

The code provides a method for determining the number of months of eligibility between a set of dates, for SSDI and SSI separately. A value of 0 for either of the measures indicates no months of eligibility for the relevant program during the period. This code can be tailored to the user's period of interest or could be modified by the user to account for combined SSDI and SSI eligibility rather than considering each program separately. For example, if a user is interested in obtaining a count of concurrent SSDI/SSI eligibility months, the code should be modified to identify months in which LAF=C or E and PSTA=C01, M01, or M02.

Note that this code determines eligibility, not payment. For more information on the difference between benefit eligibility, benefit payment, and the variables involved in determining relevant eligibility and payment status for SSDI and SSI, see Volume 3 of the DAF documentation. There are many reasons why a beneficiary may be eligible for benefits in a given month but not have received a benefit payment. For example, beneficiaries can wait months or years after application to receive a decision as to whether they meet SSA's definition of disability. In cases where an award is ultimately made, eligibility is usually retroactively determined, subject to program rules concerning onset date and waiting periods. In the DAF data then, there could be many months of eligibility without a payment.

B.   DAF files and SAS code details

The following information presents the DAF components, variables, and SAS code necessary to identify the number of months a beneficiary was eligible for SSI or SSDI benefits within a specific time period. To run this code, users should access the files OPDR.TG.PRD.ETTW.FINAL.DAF22P.YyyyyE, where yyyy ranges from 1994 through the DAF year. You can use more or fewer years to cover the timeframe for your analysis.

1. Example data step

```
* Combine Annual Files;
DATA OUT.ELIG_BETW_DATES;
MERGE
ANN94.Y1994
ANN95.Y1995
ANN96.Y1996
ANN97.Y1997
ANN98.Y1998
ANN99.Y1999
ANN00.Y2000
ANN01.Y2001
ANN02.Y2002
ANN03.Y2003
ANN04.Y2004
ANN05.Y2005
ANN06.Y2006
ANN07.Y2007
ANN08.Y2008
ANN09.Y2009
ANN10.Y2010
ANN11.Y2011
ANN12.Y2012
ANN13.Y2013
ANN14.Y2014
ANN15.Y2015
ANN16.Y2016
ANN17.Y2017
ANN18.Y2018
ANN19.Y2019
ANN20.Y2020
ANN21.Y2021
ANN22.Y2022
;
```

BY SSN;

RUN;

2. *Variables used in example*

PSTAyymm

LAFyymm

Please note: The code below uses only the variables listed above so that data elements can be limited using a SAS KEEP statement

3. *Output file created by the program*

This program creates the temporary output datasets SSIELIG and DIELIG.

4. *Variables created by the program*

This program creates the following variables:

SSI_ELIG_MOS_CNT

SSDI_ELIG_MOS_CNT

5. *SAS code*

```
%MACRO SSIELIG;
* THESE ARE THE ONLY CHANGES YOU NEED TO MAKE TO THIS CODE;

* ENTER THE 4 DIGIT YEAR AND 2 DIGIT MONTH OF THE FIRST MONTH;
%LET BEGYRMO=199401;

* ENTER THE 4 DIGIT YEAR AND 2 DIGIT MONTH OF THE LAST MONTH;
%LET ENDYRMO=202212;

DATA SSIELIG;

        SET OUT.ELIG_BETW_DATES;

   * COUNT NUMBER OF MONTHS BENEFICIARY WAS ELIGIBLE FOR SSI BENEFITS;
```

```
     ARRAY PSTAYYMM (*) $ %DO YEAR=%SUBSTR(&BEGYRMO.,1,4) %TO
     %SUBSTR(&ENDYRMO.,1,4);
                              %LET YR=%SUBSTR(&YEAR.,3,2);
                              %IF &YEAR.=%SUBSTR(&BEGYRMO.,1,4)
                              %THEN %LET STARTMO=%SUBSTR(&BEGYRMO.,5,2);
                              %ELSE %LET STARTMO=1;

                              %IF &YEAR.=%SUBSTR(&ENDYRMO.,1,4)
                              %THEN %LET STOPMO=%SUBSTR(&ENDYRMO.,5,2);
                              %ELSE %LET STOPMO=12;

                              %DO MO=&STARTMO. %TO &STOPMO.;
                                 %IF &MO.<10 %THEN PSTA&YR.0&MO.;
                              %ELSE PSTA&YR.&MO.;
                              %END;

                       %END;;


   SSI_ELIG_MOS_CNT=0;
   DO I=1 TO DIM(PSTAYYMM);
       IF PSTAYYMM(I) IN ("C01","M01","M02") THEN
   SSI_ELIG_MOS_CNT=SUM(SSI_ELIG_MOS_CNT,1);
     END;


   LABEL SSI_ELIG_MOS_CNT = "# OF MONTHS ELIGIBLE FOR SSI BETWEEN
   &BEGYRMO. AND &ENDYRMO.";


   DROP I;
RUN;
%MEND SSIELIG;
%SSIELIG;


%MACRO SSIDIELIG;
* THESE ARE THE ONLY CHANGES YOU NEED TO MAKE TO THIS CODE;
```

```
* ENTER THE 4 DIGIT YEAR AND 2 DIGIT MONTH OF THE FIRST MONTH;
%LET BEGYRMO=199401;


* ENTER THE 4 DIGIT YEAR AND 2 DIGIT MONTH OF THE LAST MONTH;
%LET ENDYRMO=202212;


DATA SSDIELIG;


SET OUT.ELIG_BETW_DATES;


  * COUNT NUMBER OF MONTHS BENEFICIARY WAS ELIGIBLE FOR SSDI
  BENEFITS;
  ARRAY LAFYYMM (*) $ %DO YEAR=%SUBSTR(&BEGYRMO.,1,4) %TO
  %SUBSTR(&ENDYRMO.,1,4);
                    %LET YR=%SUBSTR(&YEAR.,3,2);
                    %IF &YEAR.=%SUBSTR(&BEGYRMO.,1,4)
                    %THEN %LET STARTMO=%SUBSTR(&BEGYRMO.,5,2);
                    %ELSE %LET STARTMO=1;


                    %IF &YEAR.=%SUBSTR(&ENDYRMO.,1,4)
                    %THEN %LET STOPMO=%SUBSTR(&ENDYRMO.,5,2);
                    %ELSE %LET STOPMO=12;


                    %DO MO=&STARTMO. %TO &STOPMO.;
                        %IF &MO.<10 %THEN LAF&YR.0&MO.;
                    %ELSE LAF&YR.&MO.;
                    %END;
                  %END;;


    SSDI_ELIG_MOS_CNT=0;
   DO I=1 TO DIM(LAFYYMM);
      IF LAFYYMM(YR,MO) IN ("C","E") THEN
  SSDI_ELIG_MOS_CNT=SUM(SSDI_ELIG_MOS_CNT,1);
   END;
```

LABEL SSDI_ELIG_MOS_CNT= "# OF MONTHS ELIGIBLE FOR SSDI BETWEEN &BEGYRMO. AND &ENDYRMO.";

    DROP I;

RUN;

%MEND SSDIELIG;

%SSDIELIG;

II. How to Categorize Impairment Codes into the Aggregated Impairment Families

A. Description

Disability beneficiaries have a range of disabling conditions but are often categorized by SSA based on the primary impairment that makes them eligible for benefits. For example, in the SSI Annual Statistical Report, a number of tables provide information about SSI recipients by their diagnostic group.[1] The DAF contains several variables related to diagnosis code variables, but for research purposes we recommend using the constructed DAF variable DXPRIBEST. DXPRIBEST selects the best diagnosis code from the various administrative source files, such as the MBR, SSR, and 831 & 832/833 disability files, according to an algorithm developed in concert with SSA.

In Chapter V of Volume 3 we discuss two options for grouping individual impairment codes into categories: one that was developed for research purposes, and one that aligns with the SSI Annual Statistical Report. Below we present code corresponding to each of these categorization approaches. Depending on their needs, users may want to reassign certain disabling conditions to alternate groups; this can be done easily in the associated code by reassigning a given DXPRIBEST to another group.

B. DAF files and SAS code details

The following information presents the DAF components, variables, and SAS code necessary to categorize impairment codes.

Table II.1. DAF components necessary to run the code

File name	SAS name
OPDR.TG.PRD.ETTW.FINAL.DAF22P.DMG	DMG

1. Example data step

```
DATA OUT. DXGROUPS;
   SET DMGLIB.DMG;
RUN;
```

Please Note: DMGLIB references the SAS library above.

---

[1] See, for example, the section on Recipients Under Age 65 in the 2020 SSI Annual Statistical Report, available at https://www.ssa.gov/policy/docs/statcomps/ssi_asr/2020/

2. *Variable used in example*

DXPRIBEST

3. *Output file created by the program*

This program creates the temporary output dataset DXGROUPS.

4. *Variables created by the program*

The program creates the formatted variable GROUP.

5. *SAS code for SSI Annual Statistical Report categorization*

```
PROC FORMAT;
  VALUE DIAG
    1=CONGEN_ANOM
    2=ENDOCRINE
    3=INFECTIOUS
    4=INJURIES
    5=AUTISTIC
    6=DEVELOPMENT
    7=CHILDHOOD
    8=INT_DIS
    9=DEPRESSIVE_BIPOLAR
    10=NEUROCOGNITIVE
    11=SCHIZOPHREN
    12=MENTAL_OTHER
    13=NEOPLASMS
    14=BLOOD
    15=CIRCULATORY
    16=DIGESTIVE
    17=GENITO
    18=MUSCULO
    19=NERVOUS
```

```
        20=RESPIRATORY
        21=SKIN
        22=OTHER
        23=MISSING;
    RUN;

    DATA DXGROUPS;

      SET OUT.DXGROUPS;

    *** INFECTIOUS/PARASITIC DISEASES ***;
        IF DXPRIBEST>='0020' AND DXPRIBEST<='0189' THEN GROUP=3;
        ELSE IF DXPRIBEST>='0200' AND DXPRIBEST<='0279' THEN GROUP=3;
        ELSE IF DXPRIBEST>='0300' AND DXPRIBEST<='0419' THEN GROUP=3;
        ELSE IF DXPRIBEST>='0430' AND DXPRIBEST<='0579' THEN GROUP=3;
        ELSE IF DXPRIBEST>='0600' AND DXPRIBEST<='0669' THEN GROUP=3;
        ELSE IF DXPRIBEST>='0700' AND DXPRIBEST<='0889' THEN GROUP=3;
        ELSE IF DXPRIBEST>='0900' AND DXPRIBEST<='1049' THEN GROUP=3;
        ELSE IF DXPRIBEST>='1100' AND DXPRIBEST<='1189' THEN GROUP=3;
        ELSE IF DXPRIBEST>='1200' AND DXPRIBEST<='1359' THEN GROUP=3;
        ELSE IF DXPRIBEST>='1370' AND DXPRIBEST<='1399' THEN GROUP=3;
        ELSE IF DXPRIBEST>='7710' AND DXPRIBEST<='7719' THEN GROUP=3;

        *** NEOPLASMS ***;
        ELSE IF DXPRIBEST>='0420' AND DXPRIBEST<='0429' THEN GROUP=13;
        ELSE IF DXPRIBEST>='1400' AND DXPRIBEST<='1659' THEN GROUP=13;
        ELSE IF DXPRIBEST>='1700' AND DXPRIBEST<='1769' THEN GROUP=13;
        ELSE IF DXPRIBEST>='1780' AND DXPRIBEST<='2089' THEN GROUP=13;
        ELSE IF DXPRIBEST>='2100' AND DXPRIBEST<='2399' THEN GROUP=13;

        *** ENDOCRINE ***;
        ELSE IF DXPRIBEST>='2400' AND DXPRIBEST<='2469' THEN GROUP=2;
        ELSE IF DXPRIBEST>='2500' AND DXPRIBEST<='2539' THEN GROUP=2;
```

ELSE IF DXPRIBEST>='2550' AND DXPRIBEST<='2559' THEN GROUP=2;
ELSE IF DXPRIBEST>='2600' AND DXPRIBEST<='2799' THEN GROUP=2;

*** DISEASES OF THE BLOOD ***;
ELSE IF DXPRIBEST>='2800' AND DXPRIBEST<='2899' THEN GROUP=14;
ELSE IF DXPRIBEST>='7720' AND DXPRIBEST<='7739' THEN GROUP=14;
ELSE IF DXPRIBEST>='7760' AND DXPRIBEST<='7769' THEN GROUP=14;

*** AUTISM ***;
ELSE IF DXPRIBEST>='2990' AND DXPRIBEST<='2999' THEN GROUP=5;

*** DEVELOPMENTAL DISORDERS ***;
ELSE IF DXPRIBEST>='3150' AND DXPRIBEST<='3159' THEN GROUP=6;

*** CHILDHOOD/ADOLESCENT DISORDERS ***;
ELSE IF DXPRIBEST>='3120' AND DXPRIBEST<='3139' THEN GROUP=7;

*** INTELLECTUAL DISABILITY ***;
ELSE IF DXPRIBEST>='3170' AND DXPRIBEST<='3194' THEN GROUP=8;
ELSE IF DXPRIBEST>='3196' AND DXPRIBEST<='3199' THEN GROUP=8;

*** DEPRESSIVE/BIPOLAR DISORDERS ***;
ELSE IF DXPRIBEST>='2960' AND DXPRIBEST<='2969' THEN GROUP=9;
ELSE IF DXPRIBEST>='3110' AND DXPRIBEST<='3119' THEN GROUP=9;

*** NEUROCOGNITIVE MENTAL DISORDERS ***;
ELSE IF DXPRIBEST>='2900' AND DXPRIBEST<='2909' THEN GROUP=10;
ELSE IF DXPRIBEST>='2940' AND DXPRIBEST<='2949' THEN GROUP=10;
ELSE IF DXPRIBEST>='3100' AND DXPRIBEST<='3109' THEN GROUP=10;

*** SCHIZOPHRENIC ***;
ELSE IF DXPRIBEST>='2950' AND DXPRIBEST<='2959' THEN GROUP=11;
ELSE IF DXPRIBEST>='2970' AND DXPRIBEST<='2989' THEN GROUP=11;

*** OTHER MENTAL DISORDERS ***;

ELSE IF DXPRIBEST>='2910' AND DXPRIBEST<='2939' THEN GROUP=12;

ELSE IF DXPRIBEST>='3000' AND DXPRIBEST<='3099' THEN GROUP=12;

ELSE IF DXPRIBEST>='3140' AND DXPRIBEST<='3149' THEN GROUP=12;

ELSE IF DXPRIBEST>='3160' AND DXPRIBEST<='3169' THEN GROUP=12;

ELSE IF DXPRIBEST='3195' THEN GROUP=12;

*** NERVOUS SYSTEM ***;

ELSE IF DXPRIBEST>='3200' AND DXPRIBEST<='3269' THEN GROUP=19;

ELSE IF DXPRIBEST>='3290' AND DXPRIBEST<='3379' THEN GROUP=19;

ELSE IF DXPRIBEST>='3400' AND DXPRIBEST<='3749' THEN GROUP=19;

ELSE IF DXPRIBEST>='3760' AND DXPRIBEST<='3899' THEN GROUP=19;

*** CIRCULATORY ***;

ELSE IF DXPRIBEST>='3750' AND DXPRIBEST<='3759' THEN GROUP=15;

ELSE IF DXPRIBEST>='3900' AND DXPRIBEST<='3989' THEN GROUP=15;

ELSE IF DXPRIBEST>='4010' AND DXPRIBEST<='4059' THEN GROUP=15;

ELSE IF DXPRIBEST>='4100' AND DXPRIBEST<='4179' THEN GROUP=15;

ELSE IF DXPRIBEST>='4200' AND DXPRIBEST<='4389' THEN GROUP=15;

ELSE IF DXPRIBEST>='4400' AND DXPRIBEST<='4449' THEN GROUP=15;

ELSE IF DXPRIBEST>='4460' AND DXPRIBEST<='4489' THEN GROUP=15;

ELSE IF DXPRIBEST>='4510' AND DXPRIBEST<='4599' THEN GROUP=15;

*** RESPIRATORY ***;

ELSE IF DXPRIBEST>='4600' AND DXPRIBEST<='4669' THEN GROUP=20;

ELSE IF DXPRIBEST>='4700' AND DXPRIBEST<='4789' THEN GROUP=20;

ELSE IF DXPRIBEST>='4800' AND DXPRIBEST<='4879' THEN GROUP=20;

ELSE IF DXPRIBEST>='4900' AND DXPRIBEST<='4969' THEN GROUP=20;

ELSE IF DXPRIBEST>='5000' AND DXPRIBEST<='5089' THEN GROUP=20;

ELSE IF DXPRIBEST>='5100' AND DXPRIBEST<='5199' THEN GROUP=20;

ELSE IF DXPRIBEST>='7680' AND DXPRIBEST<='7709' THEN GROUP=20;

*** DIGESTIVE ***;

ELSE IF DXPRIBEST>='5200' AND DXPRIBEST<='5379' THEN GROUP=16;

ELSE IF DXPRIBEST>='5400' AND DXPRIBEST<='5439' THEN GROUP=16;

ELSE IF DXPRIBEST>='5500' AND DXPRIBEST<='5539' THEN GROUP=16;

ELSE IF DXPRIBEST>='5550' AND DXPRIBEST<='5589' THEN GROUP=16;

ELSE IF DXPRIBEST>='5600' AND DXPRIBEST<='5609' THEN GROUP=16;

ELSE IF DXPRIBEST>='5620' AND DXPRIBEST<='5629' THEN GROUP=16;

ELSE IF DXPRIBEST>='5640' AND DXPRIBEST<='5799' THEN GROUP=16;

ELSE IF DXPRIBEST>='7770' AND DXPRIBEST<='7779' THEN GROUP=16;

*** GENITOURINARY ***;

ELSE IF DXPRIBEST>='5800' AND DXPRIBEST<='6089' THEN GROUP=17;

ELSE IF DXPRIBEST>='6100' AND DXPRIBEST<='6119' THEN GROUP=17;

ELSE IF DXPRIBEST>='6140' AND DXPRIBEST<='6299' THEN GROUP=17;

*** SKIN ***;

ELSE IF DXPRIBEST>='6800' AND DXPRIBEST<='6869' THEN GROUP=21;

ELSE IF DXPRIBEST>='6900' AND DXPRIBEST<='6989' THEN GROUP=21;

ELSE IF DXPRIBEST>='7000' AND DXPRIBEST<='7099' THEN GROUP=21;

ELSE IF DXPRIBEST>='7780' AND DXPRIBEST<='7789' THEN GROUP=21;

*** MUSCULOSKELETAL ***;

ELSE IF DXPRIBEST>='7100' AND DXPRIBEST<='7399' THEN GROUP=18;

*** CONGENITAL ***;

ELSE IF DXPRIBEST>='7400' AND DXPRIBEST<='7599' THEN GROUP=1;

*** INJURIES ***;

ELSE IF DXPRIBEST>='8000' AND DXPRIBEST<='8489' THEN GROUP=4;

ELSE IF DXPRIBEST>='8500' AND DXPRIBEST<='8549' THEN GROUP=4;

ELSE IF DXPRIBEST>='8600' AND DXPRIBEST<='8879' THEN GROUP=4;

ELSE IF DXPRIBEST>='8900' AND DXPRIBEST<='8979' THEN GROUP=4;

ELSE IF DXPRIBEST>='9000' AND DXPRIBEST<='9059' THEN GROUP=4;

ELSE IF DXPRIBEST>='9070' AND DXPRIBEST<='9099' THEN GROUP=4;
ELSE IF DXPRIBEST>='9200' AND DXPRIBEST<='9299' THEN GROUP=4;
ELSE IF DXPRIBEST>='9400' AND DXPRIBEST<='9599' THEN GROUP=4;

*** OTHER ***;
ELSE IF DXPRIBEST>='7600' AND DXPRIBEST<='7609' THEN GROUP=22;
ELSE IF DXPRIBEST>='7640' AND DXPRIBEST<='7669' THEN GROUP=22;
ELSE IF DXPRIBEST>='7800' AND DXPRIBEST<='7809' THEN GROUP=22;
ELSE IF DXPRIBEST>='7830' AND DXPRIBEST<='7849' THEN GROUP=22;
ELSE IF DXPRIBEST='9330' THEN GROUP=22;

*** UNKNOWN ***;
ELSE IF DXPRIBEST='' THEN GROUP=23;
ELSE IF DXPRIBEST>='0000' AND DXPRIBEST<='0019' THEN GROUP=23;
ELSE IF DXPRIBEST>='0190' AND DXPRIBEST<='0199' THEN GROUP=23;
ELSE IF DXPRIBEST>='0280' AND DXPRIBEST<='0299' THEN GROUP=23;
ELSE IF DXPRIBEST>='0580' AND DXPRIBEST<='0599' THEN GROUP=23;
ELSE IF DXPRIBEST>='0670' AND DXPRIBEST<='0699' THEN GROUP=23;
ELSE IF DXPRIBEST>='0890' AND DXPRIBEST<='0899' THEN GROUP=23;
ELSE IF DXPRIBEST>='1050' AND DXPRIBEST<='1099' THEN GROUP=23;
ELSE IF DXPRIBEST>='1190' AND DXPRIBEST<='1199' THEN GROUP=23;
ELSE IF DXPRIBEST>='1360' AND DXPRIBEST<='1369' THEN GROUP=23;
ELSE IF DXPRIBEST>='1660' AND DXPRIBEST<='1699' THEN GROUP=23;
ELSE IF DXPRIBEST>='1770' AND DXPRIBEST<='1779' THEN GROUP=23;
ELSE IF DXPRIBEST>='2090' AND DXPRIBEST<='2099' THEN GROUP=23;
ELSE IF DXPRIBEST>='2470' AND DXPRIBEST<='2499' THEN GROUP=23;
ELSE IF DXPRIBEST>='2540' AND DXPRIBEST<='2549' THEN GROUP=23;
ELSE IF DXPRIBEST>='2560' AND DXPRIBEST<='2599' THEN GROUP=23;
ELSE IF DXPRIBEST>='3270' AND DXPRIBEST<='3289' THEN GROUP=23;
ELSE IF DXPRIBEST>='3380' AND DXPRIBEST<='3399' THEN GROUP=23;
ELSE IF DXPRIBEST>='3990' AND DXPRIBEST<='4009' THEN GROUP=23;
ELSE IF DXPRIBEST>='4060' AND DXPRIBEST<='4099' THEN GROUP=23;
ELSE IF DXPRIBEST>='4180' AND DXPRIBEST<='4199' THEN GROUP=23;

ELSE IF DXPRIBEST>='4390' AND DXPRIBEST<='4399' THEN GROUP=23;

ELSE IF DXPRIBEST>='4450' AND DXPRIBEST<='4459' THEN GROUP=23;

ELSE IF DXPRIBEST>='4490' AND DXPRIBEST<='4509' THEN GROUP=23;

ELSE IF DXPRIBEST>='4670' AND DXPRIBEST<='4699' THEN GROUP=23;

ELSE IF DXPRIBEST>='4790' AND DXPRIBEST<='4799' THEN GROUP=23;

ELSE IF DXPRIBEST>='4880' AND DXPRIBEST<='4899' THEN GROUP=23;

ELSE IF DXPRIBEST>='4970' AND DXPRIBEST<='4999' THEN GROUP=23;

ELSE IF DXPRIBEST>='5090' AND DXPRIBEST<='5099' THEN GROUP=23;

ELSE IF DXPRIBEST>='5380' AND DXPRIBEST<='5399' THEN GROUP=23;

ELSE IF DXPRIBEST>='5440' AND DXPRIBEST<='5499' THEN GROUP=23;

ELSE IF DXPRIBEST>='5540' AND DXPRIBEST<='5549' THEN GROUP=23;

ELSE IF DXPRIBEST>='5590' AND DXPRIBEST<='5599' THEN GROUP=23;

ELSE IF DXPRIBEST>='5610' AND DXPRIBEST<='5619' THEN GROUP=23;

ELSE IF DXPRIBEST>='5630' AND DXPRIBEST<='5639' THEN GROUP=23;

ELSE IF DXPRIBEST>='6090' AND DXPRIBEST<='6099' THEN GROUP=23;

ELSE IF DXPRIBEST>='6120' AND DXPRIBEST<='6139' THEN GROUP=23;

ELSE IF DXPRIBEST>='6300' AND DXPRIBEST<='6799' THEN GROUP=23;

ELSE IF DXPRIBEST>='6870' AND DXPRIBEST<='6899' THEN GROUP=23;

ELSE IF DXPRIBEST>='6990' AND DXPRIBEST<='6999' THEN GROUP=23;

ELSE IF DXPRIBEST>='7610' AND DXPRIBEST<='7639' THEN GROUP=23;

ELSE IF DXPRIBEST>='7670' AND DXPRIBEST<='7679' THEN GROUP=23;

ELSE IF DXPRIBEST>='7790' AND DXPRIBEST<='7799' THEN GROUP=23;

ELSE IF DXPRIBEST>='7810' AND DXPRIBEST<='7829' THEN GROUP=23;

ELSE IF DXPRIBEST>='7850' AND DXPRIBEST<='7999' THEN GROUP=23;

ELSE IF DXPRIBEST>='8490' AND DXPRIBEST<='8499' THEN GROUP=23;

ELSE IF DXPRIBEST>='8550' AND DXPRIBEST<='8599' THEN GROUP=23;

ELSE IF DXPRIBEST>='8880' AND DXPRIBEST<='8899' THEN GROUP=23;

ELSE IF DXPRIBEST>='8980' AND DXPRIBEST<='8999' THEN GROUP=23;

ELSE IF DXPRIBEST>='9060' AND DXPRIBEST<='9069' THEN GROUP=23;

ELSE IF DXPRIBEST>='9100' AND DXPRIBEST<='9199' THEN GROUP=23;

ELSE IF DXPRIBEST>='9300' AND DXPRIBEST<='9329' THEN GROUP=23;

ELSE IF DXPRIBEST>='9331' AND DXPRIBEST<='9399' THEN GROUP=23;

ELSE IF DXPRIBEST>='9600' AND DXPRIBEST<='9999' THEN GROUP=23;

```
      ELSE GROUP=23;
      FORMAT GROUP DIAG.;
RUN;
```

6. *SAS code for research categorization*

```
PROC FORMAT;
  VALUE DIAG
    1=MAJ_AFFECT
    2=SCHIZOPHREN
    3=ANX_NEURO
    4=OTH_MENTAL
    5=INTEL
    6=BACK
    7=MSK
    8=INFECTIOUS
    9=HIV_AIDS
    10=NEOPLASM
    11=ENDOCR_NUTRIT
    12=BLOOD
    13=VISUAL
    14=HEARING
    15=SPEECH
    16=NERV_SYS
    17=CIRC_SYS
    18=RESP_SYS
    19=DIGEST_SYS
    20=GENITOUR_SYS
    21=SKIN
    22=CONGENITAL
    23=INJURIES
RUN;
```

```
DATA DXGROUPS;

  SET OUT.DXGROUPS;
    *** Major affective disorders ***
  IF DXPRIBEST>='2960' AND DXPRIBEST<='2969' THEN GROUP=1;
  ELSE IF DXPRIBEST>='3110' AND DXPRIBEST<='3119' THEN GROUP=1;

    *** Schizophrenia and psychoses ***
  ELSE IF DXPRIBEST>='2950' AND DXPRIBEST<='2959' THEN GROUP=2;
  ELSE IF DXPRIBEST>='2980' AND DXPRIBEST<='2989' THEN GROUP=2;


    *** Anxiety and neurotic disorders ***
  ELSE IF DXPRIBEST>='3000' AND DXPRIBEST<='3019' THEN GROUP=3;
  ELSE IF DXPRIBEST>='3080' AND DXPRIBEST<='3099' THEN GROUP=3;


    *** Other mental disorders ***
  ELSE IF DXPRIBEST>='2900' AND DXPRIBEST<='2949' THEN GROUP=4;
  ELSE IF DXPRIBEST>='2990' AND DXPRIBEST<='2999' THEN GROUP=4;
  ELSE IF DXPRIBEST>='3030' AND DXPRIBEST<='3079' THEN GROUP=4;
  ELSE IF DXPRIBEST>='3100' AND DXPRIBEST<='3109' THEN GROUP=4;
  ELSE IF DXPRIBEST>='3120' AND DXPRIBEST<='3129' THEN GROUP=4;
  ELSE IF DXPRIBEST>='3138' AND DXPRIBEST<='3169' THEN GROUP=4;
  ELSE IF DXPRIBEST>='3195' AND DXPRIBEST<='3195' THEN GROUP=4;

    *** Intellectual disability ***
  ELSE IF DXPRIBEST>='3170' AND DXPRIBEST<='3194' THEN GROUP=5;
  ELSE IF DXPRIBEST>='3196' AND DXPRIBEST<='3199' THEN GROUP=5;

    *** Back disorders ***
  ELSE IF DXPRIBEST>='7221' AND DXPRIBEST<='7249' THEN GROUP=6;

    *** Musculoskeletal system ***
```

ELSE IF DXPRIBEST>='7100' AND DXPRIBEST<='7200' THEN GROUP=7;
ELSE IF DXPRIBEST>='7250' AND DXPRIBEST<='7399' THEN GROUP=7;

   *** Infectious & parasitic diseases ***
ELSE IF DXPRIBEST>='0110' AND DXPRIBEST<='0119' THEN GROUP=8;
ELSE IF DXPRIBEST>='0450' AND DXPRIBEST<='0459' THEN GROUP=8;
ELSE IF DXPRIBEST>='0930' AND DXPRIBEST<='1359' THEN GROUP=8;
ELSE IF DXPRIBEST>='1380' AND DXPRIBEST<='1389' THEN GROUP=8;

   *** HIV/AIDS ***
ELSE IF DXPRIBEST>='0070' AND DXPRIBEST<='0079' THEN GROUP=9;
ELSE IF DXPRIBEST>='0201' AND DXPRIBEST<='0449' THEN GROUP=9;
ELSE IF DXPRIBEST>='0540' AND DXPRIBEST<='0559' THEN GROUP=9;
ELSE IF DXPRIBEST>='0780' AND DXPRIBEST<='0789' THEN GROUP=9;
ELSE IF DXPRIBEST>='1360' AND DXPRIBEST<='1369' THEN GROUP=9;

   *** Neoplasms ***
ELSE IF DXPRIBEST>='1400' AND DXPRIBEST<='2399' THEN GROUP=10;

   *** Endocrine/ nutritional ***
ELSE IF DXPRIBEST>='2400' AND DXPRIBEST<='2479' THEN GROUP=11;
ELSE IF DXPRIBEST>='2500' AND DXPRIBEST<='2559' THEN GROUP=11;
ELSE IF DXPRIBEST>='2630' AND DXPRIBEST<='2799' THEN GROUP=11;

   *** Blood/ blood-forming diseases ***
ELSE IF DXPRIBEST>='2800' AND DXPRIBEST<='2899' THEN GROUP=12;

   *** Severe visual impairment ***
ELSE IF DXPRIBEST>='3610' AND DXPRIBEST<='3699' THEN GROUP=13;
ELSE IF DXPRIBEST>='3780' AND DXPRIBEST<='3789' THEN GROUP=13;

   *** Severe hearing impairment ***
ELSE IF DXPRIBEST>='3890' AND DXPRIBEST<='3899' THEN GROUP=14;

*** Severe speech impairment ***

ELSE IF DXPRIBEST>='7840' AND DXPRIBEST<='7849' THEN GROUP=15;

*** Nervous system ***

ELSE IF DXPRIBEST>='3200' AND DXPRIBEST<='3419' THEN GROUP=16;

ELSE IF DXPRIBEST>='3430' AND DXPRIBEST<='3599' THEN GROUP=16;

ELSE IF DXPRIBEST>='3860' AND DXPRIBEST<='3889' THEN GROUP=16;

*** Circulatory system ***

ELSE IF DXPRIBEST>='3420' AND DXPRIBEST<='3429' THEN GROUP=17;

ELSE IF DXPRIBEST>='3750' AND DXPRIBEST<='3759' THEN GROUP=17;

ELSE IF DXPRIBEST>='3900' AND DXPRIBEST<='4599' THEN GROUP=17;

*** Respiratory system ***

ELSE IF DXPRIBEST>='4600' AND DXPRIBEST<='4869' THEN GROUP=18;

ELSE IF DXPRIBEST>='4910' AND DXPRIBEST<='5199' THEN GROUP=18;

ELSE IF DXPRIBEST>='7690' AND DXPRIBEST<='7699' THEN GROUP=18;

*** Digestive system ***

ELSE IF DXPRIBEST>='5200' AND DXPRIBEST<='5799' THEN GROUP=19;

*** Genitourinary system ***

ELSE IF DXPRIBEST>='5800' AND DXPRIBEST<='6299' THEN GROUP=20;

*** Skin/ subcutaneous tissue ***

ELSE IF DXPRIBEST>='6900' AND DXPRIBEST<='7099' THEN GROUP=21;

*** Congenital anomalies ***

ELSE IF DXPRIBEST>='7400' AND DXPRIBEST<='7599' THEN GROUP=22;

*** Injuries ***

ELSE IF DXPRIBEST>='8000' AND DXPRIBEST<='9599' THEN GROUP=23;

```
        FORMAT GROUP DIAG.;
    RUN;
```

III. How to Determine the Number of Months Benefits Were Suspended or Terminated as a Result of Work Activity Within a Specified Time Period

A. Description

Eligibility for SSA disability benefits is based on an inability to engage in SGA, meaning that beneficiaries who sustain employment above SGA may have their SSDI benefits suspended or terminated because they returned to work. For the SSI program, disability-based benefits may also be suspended or terminated as a result of work activity, but that determination is based purely on the amount of earnings in combination with the beneficiary's other income without regard to SGA status. The DAF contains information on the months in which suspense or termination for work occurs, in variables known as "STW" (suspense or termination for work). As described in Volume 3 of the DAF documentation, there are separate variables for determining STW in SSI and SSDI, as well as a single indicator that combines information across the two programs.

This SAS code provides the user with a way to determine whether a beneficiary spent any months in STW during a given period by counting the number of months in which this was the case. A value of 0 for this measure indicates no months of STW during the period. The user can modify this code to their own time period of interest. Note that this code uses the "combined" STW indicator, STWCM, which considers suspense or termination in either SSDI or SSI. As described in Volume 3 of the DAF documentation, this variable errs toward current pay status, meaning that if a beneficiary is in STW in one program but not the other, the combined indicator does not show STW. In other words, only when a concurrent beneficiary is in STW in both programs does the STWCM indicator show a loss of benefits due to work. This code could easily be modified to consider STW status in SSI or SSDI separately, using STWSSIyymm or STWDIyymm.

The STW variables account for both suspense and termination of cash benefits due to work, using different values. The associated code includes STW values of 1, 2 or 3 as being no longer entitled to cash benefits due to work; this definition includes both suspense or termination. If a user was only interested in the termination of benefits for work, this code could be modified to only consider values of STW=2 or 3.

B. DAF files and SAS code details

The following information presents the DAF components, variables, and SAS code necessary to determine the number of months in which a beneficiary had cash benefits suspended or terminated as result of work activity within a specified time period. To run this code, users should access the files OPDR.TG.PRD.ETTW.FINAL.DAF22P.YyyyyE, where yyyy ranges from 1994 through the DAF year. You can use more or fewer years to cover the timeframe for your analysis.

1. *Example data step*

```
DATA OUT.STW;
MERGE
ANN94.Y1994
ANN95.Y1995
ANN96.Y1996
ANN97.Y1997
ANN98.Y1998
ANN99.Y1999
ANN00.Y2000
ANN01.Y2001
ANN02.Y2002
ANN03.Y2003
ANN04.Y2004
ANN05.Y2005
ANN06.Y2006
ANN07.Y2007
ANN08.Y2008
ANN09.Y2009
ANN10.Y2010
ANN11.Y2011
ANN12.Y2012
ANN13.Y2013
ANN14.Y2014
ANN15.Y2015
ANN16.Y2016
ANN17.Y2017
ANN18.Y2018
ANN19.Y2019
ANN20.Y2020
ANN21.Y2021
ANN22.Y2022
```

```
  ;
BY SSN;
RUN;
```

Note: ANNyy references the SAS libraries above.

2. *Variables used in example*

STWCMyymm

3. *Output file created by the program*

This program creates the temporary output dataset STW.

4. *Variables created by the program*

This program creates the variable STW_MOS_CNT:

5. *SAS Code*

```
%MACRO STW;

* THESE ARE THE ONLY CHANGES YOU NEED TO MAKE TO THIS CODE (THE CODE
  BELOW SHOWS THE EARLIEST AND LATEST POSSIBLE VALUES);

* ENTER THE 4 DIGIT YEAR AND 2 DIGIT MONTH OF THE FIRST MONTH;
%LET BEGYRMO=199401;

* ENTER THE 4 DIGIT YEAR AND 2 DIGIT MONTH OF THE LAST MONTH;
%LET ENDYRMO=202212;

DATA STW;
  SET OUT.STW;
   * COUNT NUMBER OF MONTHS BENEFICIARY WAS IN STW STATUS;
```

```
ARRAY STWCMYYMM
(%SUBSTR(&BEGYRMO.,1,4):%SUBSTR(&ENDYRMO.,1,4),1:12) %DO
YEAR=%SUBSTR(&BEGYRMO.,1,4) %TO %SUBSTR(&ENDYRMO.,1,4);
                              %LET YR=%SUBSTR(&YEAR.,3,2);
                              STWCM_PUF&YR.01-STWCM_PUF&YR.12
                    %END;;


 STW_MOS_CNT=0;
 DO YR=%SUBSTR(&BEGYRMO.,1,4) TO %SUBSTR(&ENDYRMO.,1,4);
     IF YR=%SUBSTR(&BEGYRMO.,1,4) THEN
STARTMO=%SUBSTR(&BEGYRMO.,5,2);
   ELSE STARTMO=1;


   IF YR=%SUBSTR(&ENDYRMO.,1,4) THEN STOPMO=%SUBSTR(&ENDYRMO.,5,2);
   ELSE STOPMO=12;


   DO MO=STARTMO TO STOPMO;
           IF STWCMYYMM(YR,MO) IN (1,2,3) THEN
STW_MOS_CNT=SUM(STW_MOS_CNT,1);
       END;
  END;


 LABEL STW_MOS_CNT = "# OF MONTHS IN STW BETWEEN &BEGYRMO. AND
 &ENDYRMO.";
 DROP I;
RUN;
%MEND STW;
%STW;
```

IV. How to determine whether a beneficiary is no longer entitled to benefits as a result of work activity within a specified time period

A. Description

In the previous section, we highlighted how to count the number of months in a specified period that a beneficiary was not entitled to cash benefits because they were working at a substantial level. This SAS code complements that in the previous section by providing the user with a way to identify beneficiaries who entered suspense or termination for work within a certain period after some number of months in current payment status. In the previous section, we offered a way to simply count the number of months in STW status during a period. In this section, we offer code to identify whether, during a user-specified period, the beneficiary was in current pay status in at least one month, but was in suspended or terminated for work status at the end of the period.

A value of 0 for the WORKING_FLAG measure defined below indicates that the beneficiary did not meet these criteria during the specified period. Combining this measure with the previous one can allow users a way to identify beneficiaries new to STW status during a period. The user can modify this code to their own time period of interest.

Note that this code uses the "combined" STW indicator, STWCM, which considers suspense or termination in either SSDI or SSI. As described in Volume 3 of the DAF documentation, this variable errs toward current pay status, meaning that if a beneficiary is in STW in one program but not the other, the combined indicator does not show STW. In other words, only when a concurrent beneficiary is in STW in both programs does the STWCM indicator show a loss of benefits due to work. This code could easily be modified to consider STW status in SSI or SSDI separately, using STWSSIyymm or STWDIyymm.

The STW variables account for both suspense and termination of cash benefits due to work, using different values. The associated code includes STW values of 1, 2 or 3 as being no longer entitled to cash benefits due to work; this definition includes both suspense or termination. If a user was only interested in the termination of benefits for work, this code could be modified to only consider values of STW=2 or 3.

B. DAF files and SAS code details

The following information presents the DAF components, variables, and SAS code necessary to determine the number of months in which a beneficiary had cash benefits suspended or terminated as result of work activity within a specified time period. To run this code, users should access the files OPDR.TG.PRD.ETTW.FINAL.DAF22P.YyyyyE, where yyyy ranges from 1994 through the DAF year. You can use more or fewer years to cover the timeframe for your analysis.

1. *Example data step*

```
DATA OUT.STW;
MERGE
ANN94.Y1994
ANN95.Y1995
ANN96.Y1996
ANN97.Y1997
ANN98.Y1998
ANN99.Y1999
ANN00.Y2000
ANN01.Y2001
ANN02.Y2002
ANN03.Y2003
ANN04.Y2004
ANN05.Y2005
ANN06.Y2006
ANN07.Y2007
ANN08.Y2008
ANN09.Y2009
ANN10.Y2010
ANN11.Y2011
ANN12.Y2012
ANN13.Y2013
ANN14.Y2014
ANN15.Y2015
ANN16.Y2016
ANN17.Y2017
ANN18.Y2018
ANN19.Y2019
ANN20.Y2020
ANN21.Y2021
ANN22.Y2022
```

```
  ;
BY SSN;
RUN;
```

Note: ANNyy references the SAS libraries above.

2. *Variables used in example*

STWCMyymm

3. *Output file created by the program*

This program creates the temporary output dataset STW.

4. *Variables created by the program*

This program creates the variable STW_MOS_CNT:

5. *SAS Code*

```
%MACRO STW;

* THESE ARE THE ONLY CHANGES YOU NEED TO MAKE TO THIS CODE (THE CODE
  BELOW SHOWS THE EARLIEST AND LATEST POSSIBLE VALUES);

* ENTER THE 4 DIGIT YEAR AND 2 DIGIT MONTH OF THE FIRST MONTH;
%LET BEGYRMO=199401;

* ENTER THE 4 DIGIT YEAR AND 2 DIGIT MONTH OF THE LAST MONTH;
%LET ENDYRMO=202212;

DATA STW;
  SET OUT.STW;
    * DETERMINE IF IN CURRENT PAY DURING THE PERIOD;
```

```
        ARRAY STWCMYYMM
(%SUBSTR(&BEGYRMO.,1,4):%SUBSTR(&ENDYRMO.,1,4),1:12) %DO
YEAR=%SUBSTR(&BEGYRMO.,1,4) %TO %SUBSTR(&ENDYRMO.,1,4);
                                        %LET YR=%SUBSTR(&YEAR.,3,2);
                                        STWCM &YR.01-STWCM &YR.12
                            %END;;

        HAS_CUR_PAY=0;
   DO YR=%SUBSTR(&BEGYRMO.,1,4) TO %SUBSTR(&ENDYRMO.,1,4)
UNTIL(HAS_CUR_PAY=1);
            IF YR=%SUBSTR(&BEGYRMO.,1,4) THEN
STARTMO=%SUBSTR(&BEGYRMO.,5,2);
    ELSE STARTMO=1;

    IF YR=%SUBSTR(&ENDYRMO.,1,4) THEN STOPMO=%SUBSTR(&ENDYRMO.,5,2);
    ELSE STOPMO=12;

            DO MO=STARTMO TO STOPMO UNTIL(HAS_CUR_PAY=1);
                IF STWCMYYMM(YR,MO)=0 THEN HAS_CUR_PAY=1;
            END;
   END;

        WORKING_FLAG=0;
        IF HAS_CUR_PAY>=1 AND
STWCMYYMM(%SUBSTR(&ENDYRMO.,1,4),%SUBSTR(&ENDYRMO.,5,2)) IN (1,2,3)
THEN WORKING_FLAG=1;

        LABEL WORKING_FLAG = "IN CURRENT  PAY BETWEEN &BEGYRMO. AND
&ENDYRMO. AND STW IN &ENDYRMNO.";

   DROP MO YR;RUN;

%MEND STW;

%STW;
```

V. How to Reorder Variables Suffixed 1-N into a Chronological Order

A. Description

The DAF includes many n suffixed variables; they are variables that have multiple occurrence often for different time periods. In the DAF, these variables are generally numbered in the order they appear on the SSA administrative data sources. Under the current DAF structure, some users may incorrectly assume that these variables to be in some sort of chronological order.

The associated SAS code that can be customized by the user to reorder any of the family of variables in the DAF. Since there are multiple SSA administrative data sources with multiple families of n variables it is difficult to provide a one size fits all set of code. So, as a starting point, the code here is specific to one family of MBR variables. In the future, we may make the code more generic so that it can be executed using user specified parameters. The MBR includes a family of data called Beneficiary Claim Data (see the MBR layout in Volume 10, "DAF Administrative Source File Documentation"). The DAF names for the variables from this family are: NDOF, BDOFn, BDOE_STARTn, BDOE_TERMn, HBICn. The n variables should be ordered chronologically based on the date in BDOE_STARTn. This means that if BDOE_START2 has an earlier date than BDOE_START1 then all the 2 suffixed variables in this family should become the 1 suffixed variables in the reordered set n variables. The NDOF indicates the number of occurrences of the variables in this family. So, if NDOF=3 then there are 3 occurrences each of BDOF, BDOE_START, BDOE_TERM, & HBIC. Use the NDOF to limit the number of occurrences you consider in your code.

B. DAF files and SAS code details

The following information presents the DAF components, variables, and SAS code necessary to reorder variables suffixed 1-n into a chronological order.

Table V.1. DAF components necessary to run the code

File name	SAS name
OPDR.TG.PRD.ETTW.FINAL.DAF22P.DMG	DMG

1. Example data step

DATA TEMPDMG;
    SET DMGLIB.DMG;
 RUN;

Note: DMGLIB references the SAS library above.

2. Variables used in example

NDOF

BDOE_STARTn

BDOE_TERMn

BDOFn

HBICn

3. Output file created by the program

This program creates the temporary output dataset REORD.

4. Variables created by the program

This program creates the variable REORD_FLAG.

5. SAS code

```
PROC SQL;
  SELECT MAX(NDOF) INTO :MAX_NDOF SEPARATED BY " "
  FROM TEMPDMG;
QUIT;

%PUT MAX_NDOF=&MAX_NDOF;

%MACRO START;
DATA REORD (DROP=I SORTED TEMP:);
  SET TEMPDMG;

  * INPUT ARRAY *;
  ARRAY BDOE_START(&MAX_NDOF);
  ARRAY BDOE_TERM(&MAX_NDOF);
  ARRAY BDOF(&MAX_NDOF);
  ARRAY HBIC(&MAX_NDOF);
```

/* THE DO UNTIL LOOP ITERATES UNTIL ALL OF THE VARIABLE *//* VALUES
   WITHIN AN OBSERVATION HAVE BEEN SORTED.*/

/* SET SORTED TO 1 AND SORTED WILL BE SET TO 0 EACH TIME*/

/* THE DO GROUP EXECUTES TO REORDER VALUES.*/

/* WHEN THAT CODE DOES NOT EXECUTE, THE ARRAY*/

/* IS ALREADY SORTED, SORTED REMAINS 1 AND PREVENTS*/

/* THE DO UNTIL LOOP FROM EXECUTING AGAIN.*/

```
DO UNTIL (SORTED);
  SORTED=1;
  DO I=1 TO &MAX_NDOF-1;
    IF NOT MISSING(BDOE_START(I+1)) AND
      BDOE_START(I) > BDOE_START(I+1) THEN DO;
      TEMP  = BDOE_START(I+1);
      TEMP1 = BDOF(I+1);
      TEMP2 = BDOE_TERM(I+1);
      TEMP3 = HBIC(I+1);

      BDOE_START(I+1) = BDOE_START(I);
      BDOF(I+1)      = BDOF(I);
      BDOE_TERM(I+1)  = BDOE_TERM(I);
      HBIC(I+1)      = HBIC(I);

      BDOE_START(I) = TEMP;
      BDOF(I)      = TEMP1;
      BDOE_TERM(I)  = TEMP2;
      HBIC(I)      = TEMP3;

      SORTED = 0;
    END;
  END;
END;
```

```
     * CREATE FLAG VARIABLE TO INDICATE REORDER *;
   IF MISSING(TEMP) THEN REORD_FLAG=1;
      ELSE REORD_FLAG=0;
RUN;
%MEND START;
%START;
```

Mathematica Inc.

Our employee-owners work nationwide and around the world.

Find us at **mathematica.org** and **edi-global.com**.