SHEEO

STATE HIGHER EDUCATION EXECUTIVE OFFICERS ASSOCIATION

POSTSECONDARY DATA PARTNERSHIP (PDP) TECHNICAL BRIEF:

USING SAS SOFTWARE TO PREPARE PDP FILE UPLOADS

MARCH 2020

ACKNOWLEDGMENTS

This technical brief was developed in partnership with Achieving the Dream. SHEEO is especially grateful to Bobbie Frye from Achieving the Dream for her technical expertise in creating this document.

This paper is based on research funded in part by the Bill & Melinda Gates Foundation. The findings and conclusions contained within are those of the author(s) and do not necessarily reflect positions or policies of the Bill & Melinda Gates Foundation.

TABLE OF CONTENTS

INTRODUCTION

This document is part of SHEEO's efforts to support the higher education community with technical assistance[1] related to the National Student Clearinghouse (Clearinghouse) Postsecondary Data Partnership (PDP).[2] The PDP is a national partnership to support institution- and system-leveldata use and reduce reporting burden. By providing additional data files to the Clearinghouse, institutions and systems have access to Key Performance Indicator (KPI) Tableau dashboards with a specific emphasis on early momentum metrics, an analysis-ready file with PDP submitted and Clearinghouse calculated fields that include Clearinghouse enrollment and degree data, as well as decreased data reporting requirements for institutions and systems participating in initiatives that are also PDP partners.

The objective of this technical brief is to discuss an approach to preparing PDP data collections that consists of formatting both term cohort files and course files using SAS macros. The process used SAS software and macro variables to identify the reporting institution, create additional required identifying fields, and export the files for uploading to the Clearinghouse, using the appropriate file naming conventions. The data preparation approach was employed by a Central Piedmont Community College's Center for Applied Research (CFAR) analyst during the initial PDP data collection. The approach has been replicated at additional Achieving the Dream Network colleges and provides an efficient method for file preparation.

[1] https://sheeo.org/pdp

[2] https://www.studentclearinghouse.org/colleges/pdp

USING SAS SOFTWARE TO PREPARE PDP FILE UPLOADS

The Clearinghouse Postsecondary Data Partnership data collections require two types of files. The first type of file is a single-record student cohort file that consists of all new students at the institution for the term being reported. The second type of file is a multiple-record course file that consists of students' course records for the term being reported. All cohort students to date should be captured in the course files. To facilitate the data collections, the analyst used macro variables in the SAS environment to create required fields and to provide identifying information in the file uploads. The macros were loaded during the file creation process to 1) provide proper formatting and complete field names, and 2) to accomplish the creation of files for the PDP with headers and footers.

DATA FILES PREPARATION

To facilitate the preparation of cohort and course files, the Clearinghouse provides documentation for file formatting and field length requirements.[3] Additionally, the Clearinghouse provides sample templates that can be downloaded and used as references in the preparation of the data files. CFAR's analyst utilized the file documentation and sample templates to create files for both the cohort and course file extracts for each term reported. The files were formatted with appropriate header/footer rows and prepared for submission to the Clearinghouse.

The use of macro variables in the SAS environment added efficiency to the file creation process, and macro variables were utilized throughout file creation to aid in accomplishing file and field formatting requirements. The flexibility of macros equipped the analyst with a file creation process that could be replicated across multiple years or terms. Additionally, to harness the power of SAS macros in file creation processing, macros were combined with LIBNAME statements to point to file locations, used to add additional required fields, and used in file naming conventions.

In this brief, several pieces of SAS code were extracted and isolated for illustration purposes. The pieces of code can be located in the complete set of cohort and course SAS software programs by consulting the appendices (see *Appendix A: Entire Cohort Program for Sample Institution* and *Appendix B: Entire Course Program for Sample Institution*).

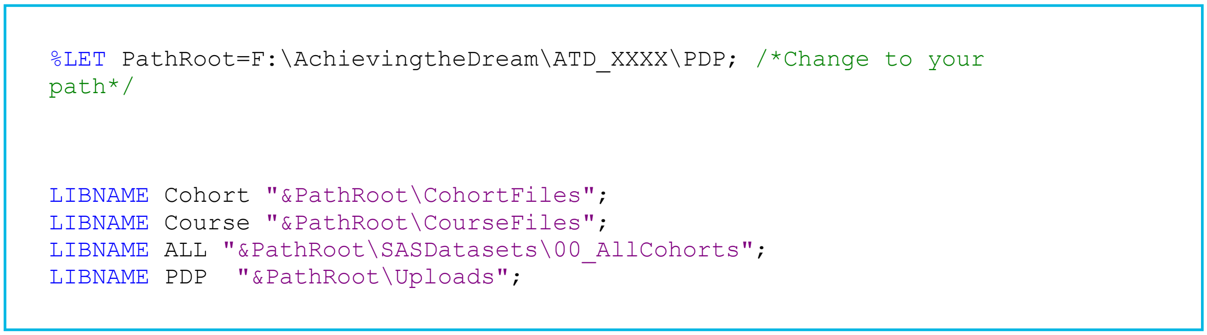
MACROS IN THE PATH LOCATION AND LIBNAME STATEMENTS

LIBNAME statements provide file addresses and name the folder locations that will be used during file processing. Macro variables facilitate the preparation of the files for the PDP data collection. During file preparation, the analyst provides information in the SAS environment regarding what files will be accessed during processing and where the files are located. In this example, LIBNAME statements identify the location of SAS files to be accessed and can also be combined with macro variables.

[3] National Student Clearinghouse. *Postsecondary Data Partnership data submission guide (2019).* https://studentclearinghouse.info/onestop/wp-content/uploads/PDP-Submission-Guide.pdf

The first macro listed below (see *Figure 1*) assigns a path location for your PDP files.[4] The path is a main folder that resides on your local computer or network. The macro variable name in this example is assigned by the statement: *%LET VarName=EntirePathLocation;* and is subsequently employed in LIBNAME statements to add additional information about where the files are located. Notice the use of double quotes to embed the macro PathRoot in LIBNAME statements. Also, in this example, the macro name is PathRoot, and the path location is provided so that the program can locate the PDP files to be utilized. The LIBNAME statement can then be used to provide additional path information and point to the sub-folder locations where the specific files are located in the folder named PathRoot on physical drive F.

FIGURE 1:
USING MACROS TO PROVIDE PATH AND FOLDER LOCATIONS



```
%LET PathRoot=F:\AchievingtheDream\ATD_XXXX\PDP; /*Change to your
path*/



LIBNAME Cohort "&PathRoot\CohortFiles";
LIBNAME Course "&PathRoot\CourseFiles";
LIBNAME ALL "&PathRoot\SASDatasets\00_AllCohorts";
LIBNAME PDP  "&PathRoot\Uploads";
```

MACROS IN THE FILE CREATION PROCESS

The cohort files included new students in each term being reported. While the institution's existing SAS data sets included a data field that identified the students as new students in a particular term, not all of the information was included in the data sets. CFAR's analyst completed the process with the required data fields by utilizing macro variables in SAS software. Notice the use of macros in *Figure 2* to name the academic term using the institution's term naming convention for ease of reference, the academic year for the respective cohort, and the name of the term.

Additionally, in September 2019, the Clearinghouse added a requirement that term begin and term end dates be added to the cohort files. The additional information was accomplished by consulting the academic calendar and using macro variables to accommodate the changes. Information that could not be located in the student information system and was added to the files is shown below, respectively. If the information is required by the Clearinghouse, missing information will create an error when files are uploaded to the Clearinghouse. File preparers will need to adhere to field requirements and ensure that required fields are provided for each student in the file.

[4] SAS Institute Inc. The code for this paper was generated using SAS software. Copyright, SAS Institute Inc. SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc., Cary, NC, USA. SAS support website: https://support.sas.com/en/technical-support.html

FIGURE 2:
USING MACROS TO PROVIDE TERM INFORMATION

```
%LET yt01=2019SU; /*Academic term using term naming convention*/
%LET ch01=2018-19; /*Academic year*/
%LET chtm01=Summer; /*PDP term - Fall, Winter Spring or Summer*/
%LET TermBeginDate=20190521; /*Term Begin Date in this format, yyyymmdd*/
%LET TermEndDate=20190720;   /*Term End   Date in this format, yyyymmdd*/
```

To add the required fields to the cohort and course files, the analyst added data fields to the files by associating the field name with the macro field in "quotes," as shown below in *Figure 3.* Associating a field name with a SAS macro requires the following naming convention: *FieldName="&macroname"*. Data fields that were not populated in the institution's files were subsequently added using SAS coding. The process required collaboration with the institution's registrar to ensure that the fields were accurately coded. The files were processed in the SAS environment using appropriate field formatting and prepared for uploading to the Clearinghouse.

FIGURE 3:
USING MACROS TO CREATE FIELDS

```
CohortTerm="&chtm01";

TermBegin="&TermBeginDate";

TermEnd="&TermEndDate";

Cohort="&ch01";
```

MACROS IN THE FILE FORMATTING PROCESS

The Clearinghouse process requires file naming conventions and files uploaded in a specific format. In each file, two header rows are required: the first row identifies the institution reporting files, and the second row contains the field names associated with the files. The footer row requires T1 in the first position of the row followed by the total number of rows reported, including the footer row. Not adhering to the file structure will cause a structural file error, and Clearinghouse files are not accepted for processing. The specific file structure can be accomplished by using SAS macros to identify the reporting institution's PDP account information and corresponding PUT statements for each segment of the Clearinghouse file (see *Figure 4*).

FIGURE 4:
USING MACROS TO ENTER STUDENT CLEARINGHOUSE INFORMATION

```
/*Enter Student Clearinghouse Information*/

%LET SBD=20191112;      /* NSC File Completion Date */
%LET SC=XXXXXX;       /* OPEID# */
%LET SA=XXXXXXXX;   /*Service Account # for Header Row*/
%LET BC=00;    /*Branch Code # for Header Row*/
%LET ID=XXXX;  /*Institutional Short Name for Header Row*/
%LET SFTPMailbox=XXXXXXXpdp; /*PDP Mailbox Address OPEID, BC, & pdp*/
```

The SAS code in *Table 5* provides the location to export the final file and names the file using the macro variables as defined. The set statement reads the SAS datafile—final&yt01—that was created during file processing. The file statement names the file with the appropriate txt extension and establishes the location for the course file, i.e., the uploads folder appropriately named PDP in the LIBNAME statement (see *Figure 1*).

FIGURE 5:
USING MACROS TO ENTER STUDENT CLEARINGHOUSE TEXT FILE INFORMATION

```
/*Export to Text File*/

data test2;
set  /*input sas file*/
 Final&yt01 end=IsLast ;
 file  "&pathroot\Uploads\'&SFTPMailbox'_course&yt01&SC .txt" DSD dlm=",";

.
```

The first PUT statement creates the first of two header rows on the first line of text (_N_=1). Notice the use of macro variables to accomplish the Clearinghouse requirements for the first header row, as shown below in *Figure 6*. Commas are used to indicate blank spaces in the header row and may be used by Clearinghouse in the future. The entire row of text is listed in double quotes.

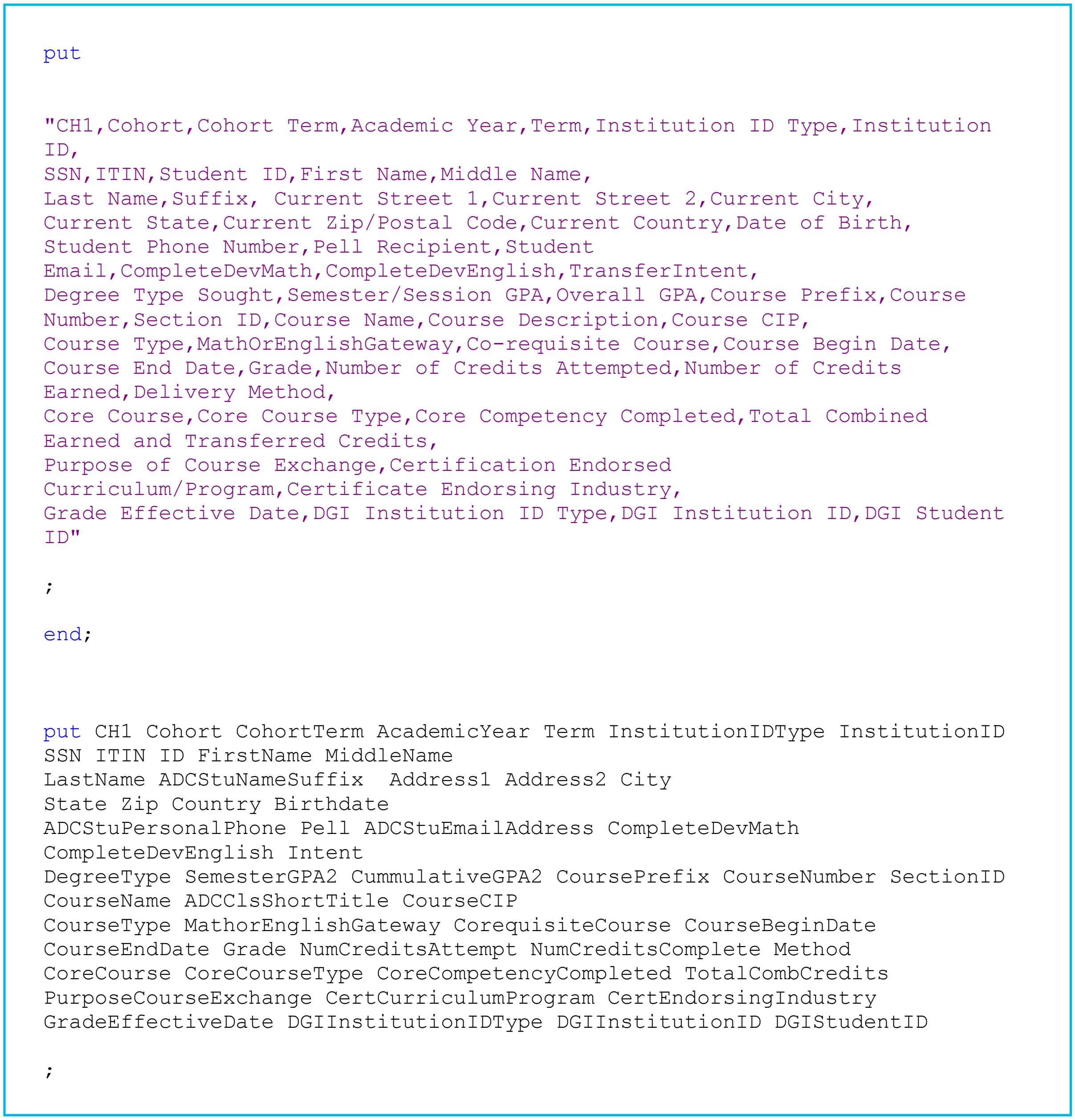
FIGURE 6:
USING MACROS TO CREATE FIRST HEADER ROW

```
If _N_=1 then do;
put
 @1 "DCE02,&SA,&SC,&BC,,,,,&SBD,,&ID,,,,"
```

The second PUT statement creates the second header row, as outlined in the Clearinghouse documentation. A third PUT statement is used to reference the actual field names in the data file. The advantage of using the two PUT statements here is that there is no need to rename your field names in your files, which can vary across data systems. The fields are associated with the specific Clearinghouse documentation requirements using the PUT statement, as shown below. The first entire set of text is enclosed in double quotes. The next PUT statement references the actual field names in the data file, and this text is not enclosed in double quotes.

FIGURE 7:

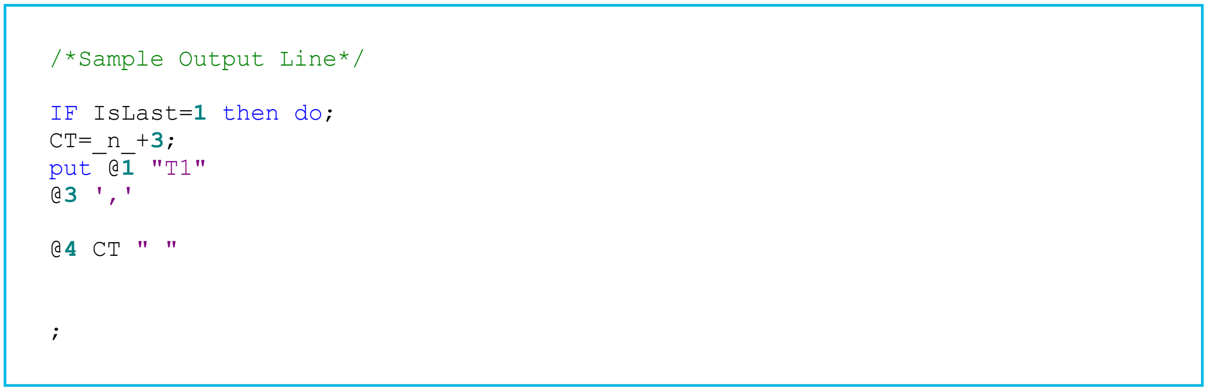
USING MACROS TO CREATE SECOND HEADER ROW AND DETAIL ROWS



```
put


"CH1,Cohort,Cohort Term,Academic Year,Term,Institution ID Type,Institution
ID,
SSN,ITIN,Student ID,First Name,Middle Name,
Last Name,Suffix, Current Street 1,Current Street 2,Current City,
Current State,Current Zip/Postal Code,Current Country,Date of Birth,
Student Phone Number,Pell Recipient,Student
Email,CompleteDevMath,CompleteDevEnglish,TransferIntent,
Degree Type Sought,Semester/Session GPA,Overall GPA,Course Prefix,Course
Number,Section ID,Course Name,Course Description,Course CIP,
Course Type,MathOrEnglishGateway,Co-requisite Course,Course Begin Date,
Course End Date,Grade,Number of Credits Attempted,Number of Credits
Earned,Delivery Method,
Core Course,Core Course Type,Core Competency Completed,Total Combined
Earned and Transferred Credits,
Purpose of Course Exchange,Certification Endorsed
Curriculum/Program,Certificate Endorsing Industry,
Grade Effective Date,DGI Institution ID Type,DGI Institution ID,DGI Student
ID"

;

end;



put CH1 Cohort CohortTerm AcademicYear Term InstitutionIDType InstitutionID
SSN ITIN ID FirstName MiddleName
LastName ADCStuNameSuffix  Address1 Address2 City
State Zip Country Birthdate
ADCStuPersonalPhone Pell ADCStuEmailAddress CompleteDevMath
CompleteDevEnglish Intent
DegreeType SemesterGPA2 CummulativeGPA2 CoursePrefix CourseNumber SectionID
CourseName ADCClsShortTitle CourseCIP
CourseType MathorEnglishGateway CorequisiteCourse CourseBeginDate
CourseEndDate Grade NumCreditsAttempt NumCreditsComplete Method
CoreCourse CoreCourseType CoreCompetencyCompleted TotalCombCredits
PurposeCourseExchange CertCurriculumProgram CertEndorsingIndustry
GradeEffectiveDate DGIInstitutionIDType DGIInstitutionID DGIStudentID

;
```

Finally, the footer row is accomplished by the final PUT statement, and proper formatting is applied to the last row of the file. Using the sample below in *Figure 8*, the footer row is formatted correctly, listed on the last row of the text file, and ready for Clearinghouse submission without manual computations.

FIGURE 8:
USING MACROS TO CREATE FOOTER ROW



```
/*Sample Output Line*/

IF IsLast=1 then do;
CT=_n_+3;
put @1 "T1"
@3 ','

@4 CT " "


;
```

CHALLENGES DURING FILE CREATIONS

Using the SAS code as displayed in *Figure 3*, the resulting 'SFTPMailbox' will be enclosed with ' ' and the set of single quotes will need to be removed in the file names before uploading to the Clearinghouse. The removal of single quotes is accomplished by simply renaming the file. Note the file naming convention in the Clearinghouse PDP submission guide.

During the second PUT statement that creates the second header row, the following note will be generated in SAS: "NOTE: The quoted string currently being processed has become more than 262 characters long. You might have unbalanced quotation marks." The note can be ignored in this context.

RESULTS

The results of utilizing SAS macros in the file creation process demonstrated an efficient process for handling the PDP data collections. The use of macro variables in the SAS environment added efficiency to the file creation process, and macro variables were utilized throughout file creation to aid in accomplishing file and field formatting requirements. The flexibility of macros equipped the analyst with a file creation process that could be replicated across multiple years or terms. Additionally, to harness the power of SAS macros in file creation processing: macros were combined with LIBNAME statements to point to file locations; used to add additional required fields, which add file flexibility; and used in file naming conventions.

Not adhering to the Clearinghouse file structure will cause a structural file error, and Clearinghouse files are not accepted for processing. The specific file structure can be accomplished by using SAS macros to identify the reporting institution's PDP account information and corresponding PUT statements for each segment of the Clearinghouse file.

Not adhering to the Clearinghouse field requirements will cause field errors, and Clearinghouse files are not accepted for processing. As additional fields are added to PDP data collection requirements, SAS macros can be associated with the additional field names.

RECOMMENDATIONS

- Macro variables require specific coding, as detailed in this document. Macro statements that do not adhere to SAS criteria will lead to errors in file processing. Pay careful attention to the macro SAS software coding and carefully read the specific coding requirements to conceptualize the field specifications needed.
- The initial data collection is the most time consuming, including up to five years of data, but the process outlined in this document can be replicated for future data collections. Macro variables are flexible and can accommodate changes over time.
- While the process can be replicated for future data collections, coding changes in the student information system will need to be addressed in the ongoing data extracts. Missing data fields can indicate a change in the location of data stored in the student information system and should be investigated.
- Using macro variables to create fields not found in the student information system should be done conservatively. The analyst used the option to add term start and end dates to the files. To document this option for record keeping and validation purposes, a spreadsheet of term start and end dates should be compiled and maintained with the data files.

APPENDIX A: ENTIRE COHORT PROGRAM FOR SAMPLE INSTITUTION

```
OPTIONS PS=5000 LS=256 NONUMBER NOCENTER;


%LET PathRoot=F:\AchievingtheDream\ATD_XXXX\PDP; /*Change to your path*/



LIBNAME Cohort "&PathRoot\CohortFiles";
LIBNAME Course "&PathRoot\CourseFiles";
LIBNAME ALL "&PathRoot\SASDatasets\00_AllCohorts";
LIBNAME PDP  "&PathRoot\Uploads";

/*Enter Student Clearinghouse Information*/


%LET SBD=20191112;      /* NSC File Completion Date */
%LET SC=XXXXXX;       /* OPEID# */
%LET SA=XXXXXXXX;   /*Service Account # for Header Row*/
%LET BC=00;    /*Branch Code # for Header Row*/
%LET ID=XXXX;  /*Institutional Short Name for Header Row*/
%LET SFTPMailbox=XXXXXXXpdp; /*PDP Mailbox Address (OPEID, BC,) & pdp*/

%LET yt01=2019SU; /*Academic term using term naming convention*/
%LET ch01=2018-19; /*Academic year*/
%LET chtm01=Summer; /*PDP term - Fall, Winter Spring or Summer*/
%LET TermBeginDate=20190521; /*Term Begin Date in this format, yyyymmdd*/
%LET TermEndDate=20190720;   /*Term End   Date in this format, yyyymmdd*/


 data WORK. test    ;
      %let _EFIERR_ = 0; /* set the ERROR detection macro variable */
      infile "&Pathroot\CohortFiles\&yt01..csv" delimiter = ','
        MISSOVER DSD lrecl=32767 firstobs=2 ;
         informat CH1 $2. ;
         informat Cohort $10. ;
         informat CohortTerm $10. ;
         informat SSN 9. ;
         informat ITIN 9. ;
         informat ID 9. ;
         informat FirstName $60. ;
         informat MiddleName $60. ;
         informat LastName $60. ;
         informat Address1 $30. ;
         informat Address2 $30. ;
         informat City $30. ;
         informat State $2. ;
         informat Zip $10. ;
         informat Country $2. ;
         informat Birthdate $8. ;
         informat Ethnicity $2. ;
         informat Race $2. ;
         informat InstitutionIDType $5. ;
         informat InstitutionID $8. ;
         informat HSDip $1. ;
         informat HSDipYr $4. ;
         informat HSGPA 4.2 ;
         informat HSWeightedGPA 4.2 ;
         informat FirstGen $1. ;
         informat Dual $2. ;
         informat EnrollmentType $1. ;
         informat NumberofCreditsAttemptedTR 6.;
         informat NumberofCreditsAcceptedTR 6.;
         informat MathPlacement $2. ;
         informat EngPlacement $2. ;
         informat GatewayMath $2. ;
         informat GatewayEnglish $2. ;
      input

              CH1 $
          Cohort $
          CohortTerm $
          SSN
          ITIN
          ID
          FirstName $
          MiddleName $
          LastName $
          Address1 $
          Address2 $
          City $
          State $
          Zip $
          Country $
          Birthdate $
          Ethnicity $
          Race $
          InstitutionIDType $
          InstitutionID $
          HSDip $
          HSDipYr $
          HSGPA
          HSWeightedGPA
          FirstGen $
          Dual $
          EnrollmentType $
          NumberofCreditsAttemptedTR
          NumberofCreditsAcceptedTR
          MathPlacement $
          EngPlacement $
          GatewayMath $
          GatewayEnglish $


      ;
      if _ERROR_ then call symputx('_EFIERR_',1);  /* set ERROR detection
macro variable */

CohortTerm="&chtm01";
TermBegin="&TermBeginDate";
TermEnd="&TermEndDate";
```

```
Cohort="&ch01";
if Address1=' ' then Address1='UK';
if Address2=' ' then Address2=' ';
if City=' ' then City='UK';
if State=' ' then State='UK';
if FirstGen=' ' then FirstGen='N';
IF Ethnicity = ('  ') THEN Ethnicity='UK';
If Ethnicity = 'H' then Race='UK';
InstitutionIDType='OPEID';
format InstitutionID $8.;
InstitutionID=cats("&SC","&BC");
CH1='D1';


     run;

/*Final Processing*/

Data Final&yt01 all .cohort_&yt01;
set test;
ITIN=' ';
SSN=' ';
run;




data test;
set  /*input sas file*/
 final&yt01 end=IsLast  ;
  file "&pathroot\Uploads\'&SFTPMailbox'_cohort&yt01&SC .txt" dsd dlm=",";
If _N_=1 then do;
put
 @1 "DCE01,&SA,&SC,&BC,,,,,&SBD,,&ID,,,,"


;



put


"CH1,Cohort,Cohort Term, Cohort Term Begin Date, Cohort Term End Date,
SSN,ITIN,Student ID,First Name,Middle Name,
Last Name,Street Line 1,Street Line 2,City,
State,Zip/Postal Code,Country,Date of Birth,Ethnicity,Race,Institution ID
Type,Institution ID,HS Completion Status,
HS Completion Year,HS Unweighted GPA,
HS Weighted GPA,First Gen,Dual and Summer Enrollment,Enrollment Type,
Number of College Credits Attempted to Transfer,Number of College Transfer
Credits Accepted,
Math Placement,English Placement,Gateway Math Status,Gateway English Status"
;

end;
```

```
put CH1 Cohort CohortTerm TermBegin TermEnd SSN ITIN ID FirstName MiddleName
LastName
Address1 Address2
City State Zip
Country Birthdate Ethnicity Race
InstitutionIDType InstitutionID HSDip HSDipYr
HSGPA HSWeightedGPA FirstGen Dual EnrollmentType
NumberofCreditsAttemptedTR NumberofCreditsAcceptedTR
MathPlacement EngPlacement
GatewayMath GatewayEnglish


;

/*Sample Output Line*/

IF IsLast=1 then do;
CT=_n_+3;
put @1 "T1"
@3 ','

@4 CT " "


;


end;

run;
```

APPENDIX B: ENTIRE COURSE PROGRAM FOR SAMPLE INSTITUTION

```
OPTIONS PS=5000 LS=256 NONUMBER NOCENTER;

%LET PathRoot=F:\AchievingtheDream\ATD_XXXX\PDP; /*Change to your path*/

LIBNAME Cohort "&PathRoot\CohortFiles";
LIBNAME Course "&PathRoot\CourseFiles";
LIBNAME ALL "&PathRoot\SASDatasets\00_AllCohorts";
LIBNAME PDP  "&PathRoot\Uploads";

/*Enter Student Clearinghouse Information*/

%LET SBD=20191112;      /* NSC File Completion Date */
%LET SC=XXXXXX;       /* OPEID# */
%LET SA=XXXXXXXX;   /*Service Account # for Header Row*/
%LET BC=00;    /*Branch Code # for Header Row*/
%LET ID=XXXX;  /*Institutional Short Name for Header Row*/
%LET SFTPMailbox=XXXXXXXpdp; /*PDP Mailbox Address OPEID, BC, & pdp*/

%LET yt01=2019SU; /*Academic term using term naming convention*/
%LET ch01=2018-19; /*Academic year*/
%LET chtm01=Summer; /*PDP term - Fall, Winter Spring or Summer*/

/*Cohort Terms Reported*/

%LET yt1=2013FA;
%LET yt2=2014SP;
%LET yt3=2014SU;
%LET yt4=2014FA;
%LET yt5=2015SP;
%LET yt6=2015SU;
%LET yt7=2015FA;
%LET yt8=2016SP;
%LET yt9=2016SU;
%LET yt10=2016FA;
%LET yt11=2017SP;
%LET yt12=2017SU;
%LET yt13=2017FA;
%LET yt14=2018SP;
%LET yt15=2018SU;
%LET yt16=2018FA;
%LET yt17=2019SP;
%LET yt18=2019SU;
%LET yt19=2019FA;

     data WORK.Course    ;
      %let _EFIERR_ = 0; /* set the ERROR detection macro variable */
      infile "&Pathroot\CourseFiles\&yt01 course.csv" delimiter = ','
MISSOVER DSD lrecl=32767 firstobs=2 ;
         informat CH1 $2. ;
         informat Cohort $7. ;
         informat Cohort_Term $10. ;
         informat Academic_Year $7. ;
         informat Term $10. ;
```

```
   informat Institution_ID_Type $5. ;
   informat Institution_ID best32. ;
   informat SSN best32. ;
   informat ITIN $9. ;
   informat ID 20. ;
   informat FirstName $60. ;
   informat MiddleName $60. ;
   informat LastName $60. ;
   informat ADCStuNameSuffix $10. ;
   informat Address1 $30. ;
   informat Address2 $30. ;
   informat City $20. ;
   informat State $2. ;
   informat Zip $10. ;
   informat Country $2. ;
   informat Birthdate $20. ;
   informat ADCStuPersonalPhone $20. ;
   informat Pell $2. ;
   informat ADCStuEmailAddress $255. ;
   informat CompleteDevMath $4. ;
   informat CompleteDevEnglish $4. ;
   informat Intent $4. ;
   informat DegreeType $2. ;
   informat SemesterGPA best32. ;
   informat CummulativeGPA best32. ;
   informat CoursePrefix $50. ;
   informat CourseNumber $20. ;
   informat CourseSection $20. ;
   informat CourseName $50. ;
   informat ADCclsShortTitle $255. ;
   informat CourseCIP $10. ;
   informat CourseType $2. ;
   informat MathOrEnglishGateway $10. ;
   informat CorequisiteCourse $1. ;
   informat CourseBeginDate $20. ;
   informat CourseEndDate $20. ;
   informat LetterGrade $20. ;
   informat NumCreditsAttempt best32. ;
   informat NumCreditsComplete best32. ;
   informat Method $1. ;
   informat Core_Course $1. ;
   informat Core_Course_Type $1. ;
   informat Core_Competency_Completed $1. ;
   informat Total_Combined_Earned___Transf $1. ;
   informat Purpose_of_Course_Exchange best32. ;
   informat Certification_Endorsed_Curricu $1. ;
   informat Certificate_Endorsing_Industry $1. ;
   informat Grade_Effective_Date $1. ;
   informat DGI_Institutional_ID_Type best32. ;
   informat DGI_Institutional_ID best32. ;
   informat DGI_Student_ID $1. ;

input
            CH1  $
            Cohort  $
            Cohort_Term $
            Academic_Year  $
```

```
                  Term $
                  Institution_ID_Type  $
                  Institution_ID
                  SSN
                  ITIN  $
                  ID
                  FirstName  $
                  MiddleName  $
                  LastName  $
                  ADCStuNameSuffix  $
                  Address1  $
                  Address2  $
                  City  $
                  State  $
                  Zip
                  Country  $
                  Birthdate
                  ADCStuPersonalPhone  $
                  Pell  $
                  ADCStuEmailAddress  $
                  CompleteDevMath  $
                  CompleteDevEnglish  $
                  Intent  $
                  DegreeType  $
                  SemesterGPA
                  CummulativeGPA
                  CoursePrefix  $
                  CourseNumber
                  CourseSection  $
                  CourseName  $
                  ADCClsShortTitle  $
                  CourseCIP
                  CourseType  $
                  MathOrEnglishGateway  $
                  CorequisiteCourse  $
                  CourseBeginDate
                  CourseEndDate
                  LetterGrade  $
                  NumCreditsAttempt
                  NumCreditsComplete
                  Method  $
                  Core_Course  $
                  Core_Course_Type  $
                  Core_Competency_Completed  $
                  Total_Combined_Earned___Transf  $
                  Purpose_of_Course_Exchange
                  Certification_Endorsed_Curricu  $
                  Certificate_Endorsing_Industry  $
                  Grade_Effective_Date  $
                  DGI_Institutional_ID_Type
                  DGI_Institutional_ID
                  DGI_Student_ID  $
      ;
      if _ERROR_ then call symputx('_EFIERR_',1);  /* set ERROR detection
macro variable */
```

```
BeginYear=substr(CourseBeginDate,7,2);
BeginMonth=substr(CourseBeginDate,1,2);
BeginDay=substr(CourseBeginDate,4,2);
CourseBeginDate=cats('20',beginyear,beginmonth,beginday);

EndYear=substr(CourseEndDate,7,2);
EndMonth=substr(CourseEndDate,1,2);
EndDay=substr(CourseEndDate,4,2);
CourseEndDate=cats('20',Endyear,Endmonth,Endday);

BirthYear=substr(BirthDate,7,2);
BirthMonth=substr(BirthDate,1,2);
BirthDay=substr(BirthDate,4,2);
if BirthYear <= substr("&yt01",3,2) then
BirthDate=cats ('20',birthyear,birthmonth,birthday);
else BirthDate=cats ('19',birthyear,birthmonth,birthday);


Course=cats(CoursePrefix, CourseNumber);
if substr(Course,1,4) in ('MTH1') or substr(course,1,5) in ('MATH1')
then MathorEnglishGateway='M';
else if substr(Course,1,4) in ('ENG1') or substr(course,1,5)in ('ENGL1')
then
MathorEnglishGateway='E';
else MathorEnglishGateway='NA';

If NumCreditsAttempt < NumCreditsComplete then do;
NumCreditsAttempt=NumCreditsComplete;
end;




If LetterGrade='A' then grade='4.00';
else if LetterGrade='A-' then grade='3.67';
else if LetterGrade='B+' then grade='3.33';
else if LetterGrade='B' then grade='3.00';
else if LetterGrade='B-' then grade='2.67';
else if LetterGrade='C+' then grade='2.33';
else if LetterGrade='C' then grade='2.00';
else if LetterGrade='C-' then grade='1.67';
else if LetterGrade='D' then grade='1.00';
else if LetterGrade='F' then grade='F';
else if LetterGrade in ('P')then Grade='P';
else if LetterGrade in ('R')then Grade='O';
else if LetterGrade='Audit' then Grade='A';
else if LetterGrade in ('W','UW') then Grade='W';
else if LetterGrade='UF' then Grade='F';
else if substr(LetterGrade,1,1)='I' then Grade='I';
else if LetterGrade=' ' then Grade='M';
else Grade='O';


If Country='UN' then do;
Country='US';
end;
run;
```

```
proc sort; by id;
run;


/*Combine Cohorts Reported to Date*/
%macro cohort;
proc sort data=All .Cohort_&yt out=cohort&yt (keep=id cohort cohortterm) ;
by id;
run;
%mend;
%LET yt=&yt1; %cohort;
%LET yt=&yt2; %cohort;
%LET yt=&yt1; %cohort;
%LET yt=&yt2; %cohort;
%LET yt=&yt3; %cohort;
%LET yt=&yt4; %cohort;
%LET yt=&yt5; %cohort;
%LET yt=&yt6; %cohort;
%LET yt=&yt7; %cohort;
%LET yt=&yt8; %cohort;
%LET yt=&yt9; %cohort;
%LET yt=&yt10; %cohort;
%LET yt=&yt11; %cohort;
%LET yt=&yt12; %cohort;
%LET yt=&yt13; %cohort;
%LET yt=&yt14; %cohort;
%LET yt=&yt15; %cohort;
%LET yt=&yt16; %cohort;
%LET yt=&yt17; %cohort;
%LET yt=&yt18; %cohort;
/*%LET yt=&yt19; %cohort;
%LET yt=&yt20; %cohort;
%LET yt=&yt21; %cohort;
%LET yt=&yt22; %cohort;
%LET yt=&yt23; %cohort;
%LET yt=&yt24; %cohort;*/

/*Facilitates the Coding Change, to Cohort Term from Numeric to Fall,
Winter, Spring, Summer, in September 2019*/

Data allcohorts&yt01;
format CohortTerm $10.;
set cohort&yt1
cohort&yt2
cohort&yt3
cohort&yt4
cohort&yt5
cohort&yt6
cohort&yt7
cohort&yt8
cohort&yt9
cohort&yt10
cohort&yt11
cohort&yt12
cohort&yt13
cohort&yt14
```

```
cohort&yt15
cohort&yt16
cohort&yt17
cohort&yt18
/*cohort&yt19
cohort&yt20
cohort&yt21
cohort&yt22
cohort&yt23
cohort&yt24*/


;

if CohortTerm='1' then CohortTerm='Fall';
if CohortTerm='3' then CohortTerm='Spring';
if CohortTerm='6' then CohortTerm='Summer';
;

proc sort; by id;
run;



/*Code Developmental Requirements*/


Data DEVGroup&yt01;
set All .Cohort_&yt1
All .Cohort_&yt2
All .Cohort_&yt3
All .Cohort_&yt4
All .Cohort_&yt5
All .Cohort_&yt6
All .Cohort_&yt7
All .Cohort_&yt8
All .Cohort_&yt9
All .Cohort_&yt10
All .Cohort_&yt11
All .Cohort_&yt12
All .Cohort_&yt13
All .Cohort_&yt14
All .Cohort_&yt15
All .Cohort_&yt16
All .Cohort_&yt17
All .Cohort_&yt18


;
proc sort data=DEVGroup&yt01; by id;
run;


/*Begin Processing Developmental Completions*/

Data Course2;
```

```
merge  course (in=a)   devgroup&yt01(in=b keep=id MathPlacement EngPlacement
); by id;;
format completedevmath $2. completedevenglish $2.;
completedevmath=' ';
completedevenglish=' ';

if a;
if mathplacement='N' then do;
CompleteDevMath='D';

If course in ('MTH99','MATH904') and grade in ('P')
then CompleteDevMath='C';
end;
if mathplacement='C' then do;
CompleteDevMath='NA';
end;


if engplacement='N' then do;
CompleteDevEnglish='D';

If course in ('ENG99', 'ENGL900') and grade in ('P')
then CompleteDevEnglish='C';
end;
if engplacement='C' then do;
CompleteDevEnglish='NA';
end;

run;


Data devmath (keep=id completedevmath) deveng (keep=id completedevenglish);
set course2;
run;

proc sort data=devmath; by completedevmath;
proc sort data=devmath nodupkey; by id;
proc sort data=deveng; by completedevenglish;
proc sort data=deveng nodupkey; by id;
run;

Data CombDev;
merge deveng devmath; by id;
run;

/*Finish Creating Student Completion Attribute*/

Data course3;
merge course2 (in=a)  combdev ; by id;
if a;
if CompleteDevEnglish=' ' then CompleteDevEnglish='NA';
if CompleteDevMath=' ' then CompleteDevMath='NA';
run;


       /*Final Merge*/
```

```
Data final&yt01;
format DGIInstitutionID $8. InstitutionID $8. TotalCombCredits 10.
CoreCourseType $20. GradeEffectiveDate $8. DGIStudentID $20.;
merge allcohorts&yt01 (in=a)  course3 (in=b drop=Cohort Cohort_term term);
by id;
if a and b;
AcademicYear="&ch01";
Term="&chtm01";
SectionID='Missing';
SSN=' ';
ITIN=' ';
CH1='D1';
CoreCourse=' ';
CoreCourseType=' ';
CoreCompetencyCompleted=' ';
PurposeCourseExchange=2;
CertCurriculumProgram=' ';
CertEndorsingIndustry=' ';
GradeEffectiveDate=' ';
CourseDescription=' ';
TotalCombCredits=.;
InstitutionIDType='OPEID';
InstitutionID=cats("&SC","&BC");

DGIInstitutionIDType='OPEID';
DGIInstitutionID=cats("&SC","&BC");
DGIStudentID=ID;
Semestergpa3=put(semestergpa,4.2);
if SemesterGPA3=. then SemesterGPA2='Missing';
else SemesterGPA2=SemesterGPA3;
Cummulativegpa3=put(cummulativegpa,4.2);
if CummulativeGPA3=. then CummulativeGPA2='Missing';
else CummulativeGPA2=CummulativeGPA3;
RUN;


/*Export to Text File*/

data test2;
set  /*input sas file*/
 Final&yt01 end=IsLast ;
 file  "&pathroot\Uploads\'&SFTPMailbox'_course&yt01&SC .txt" DSD dlm=",";
If _N_=1 then do;
put
 @1 "DCE02,&SA,&SC,&BC,,,,,&SBD,,&ID,,,,"


;


put


"CH1,Cohort,Cohort Term,Academic Year,Term,Institution ID Type,Institution
ID,
SSN,ITIN,Student ID,First Name,Middle Name,
Last Name,Suffix, Current Street 1,Current Street 2,Current City,
Current State,Current Zip/Postal Code,Current Country,Date of Birth,
Student Phone Number,Pell Recipient,Student
Email,CompleteDevMath,CompleteDevEnglish,TransferIntent,
Degree Type Sought,Semester/Session GPA,Overall GPA,Course Prefix,Course
Number,Section ID,Course Name,Course Description,Course CIP,
Course Type,MathOrEnglishGateway,Co-requisite Course,Course Begin Date,
Course End Date,Grade,Number of Credits Attempted,Number of Credits
Earned,Delivery Method,
Core Course,Core Course Type,Core Competency Completed,Total Combined Earned
and Transferred Credits,
Purpose of Course Exchange,Certification Endorsed
Curriculum/Program,Certificate Endorsing Industry,
Grade Effective Date,DGI Institution ID Type,DGI Institution ID,DGI Student
ID"

;

end;


put CH1 Cohort CohortTerm AcademicYear Term InstitutionIDType InstitutionID
SSN ITIN ID FirstName MiddleName
LastName ADCStuNameSuffix  Address1 Address2 City
State Zip Country Birthdate
ADCStuPersonalPhone Pell ADCStuEmailAddress CompleteDevMath
CompleteDevEnglish Intent
DegreeType SemesterGPA2 CummulativeGPA2 CoursePrefix CourseNumber SectionID
CourseName ADCClsShortTitle CourseCIP
CourseType MathorEnglishGateway CorequisiteCourse CourseBeginDate
CourseEndDate Grade NumCreditsAttempt NumCreditsComplete Method
CoreCourse CoreCourseType CoreCompetencyCompleted TotalCombCredits
PurposeCourseExchange CertCurriculumProgram CertEndorsingIndustry
GradeEffectiveDate DGIInstitutionIDType DGIInstitutionID DGIStudentID

;

/*Sample Output Line*/

IF IsLast=1 then do;
CT=_n_+3;
put @1 "T1"
@3 ','

@4 CT " "


;


end;

run;
```

STATE HIGHER EDUCATION EXECUTIVE OFFICERS

3035 CENTER GREEN DRIVE, SUITE 100, BOULDER, COLORADO, 80301
303.541.1600 • **SHEEO.org**