Paper 095-29

Dirty Data? dfPower Studio® to the Rescue!

Kimberly A. Todd, Centers for Disease Control and Prevention, Atlanta, GA
Catherine A. Lindsey, Northrup Grumman, Atlanta, GA
Kristin M. Debnar, Volt Technical Services, Atlanta, GA

ABSTRACT

The most challenging aspect of migrating legacy data into a new system is cleaning up years of data entry inconsistencies. DataFlux's dfPower Studio® provides the tools to meet this challenge.

The HIV/AIDS Reporting System (HARS) captures data to monitor the HIV/AIDS epidemic, including names and locations of facilities of diagnosis and treatment. Each record may contain up to five different facility entries. The name and location of a given facility varies considerably from record to record. Before migrating to the new system, these data must be cleaned.

In order to standardize across all five sets of facility fields in a single pass, a SAS® data set is created containing one record per unique facility entry. dfPower Studio is then used to create and apply a standardization scheme to each facility entry record. After the scheme is applied to the SAS data set, the original HARS database is updated.

State and local health departments have utilized this process and have found it to be invaluable. In the absence of an automated data cleaning process, cleaning data manually would require days or even weeks. Using dfPower Studio and this process, years of data are uniformly standardized in a fraction of the time.

INTRODUCTION

The Centers for Disease Control and Prevention is developing a document-based application, e-HARS, to support HIV/AIDS surveillance activities and research projects. HIV/AIDS surveillance activities are currently supported by the existing HIV/AIDS Reporting System (HARS). There are many steps involved in getting the existing HARS data ready to export and load into the new e-HARS system. One such step involves the standardization of the following HARS fields used to store facility information: *bhosp* – facility at birth; *chosp* – facility at child's birth; *hhosp_dx* – facility at HIV diagnosis; *hosp_dx* – facility at AIDS diagnosis; and *phosp_dx* – facility at perinatal HIV exposure. At each site running HARS, there may be multiple people responsible for data entry. As a result, there may be multiple spellings or representations of the name of a single facility. For example, "Saint Joseph's Hospital" could also be entered as "St. Joseph's Hospital". In addition, each facility has an associated city and state. Viewing the facility name, city, and state together may aid in accurately distinguishing correct facility names from possible misspellings. For example, "Allandale Clinic" may appear to be a misspelling of "Allendale Clinic" when looking only at facility name. However, when the facility names are viewed with their city and state, it can be determined that "Allandale Clinic, Atlanta, GA" is a unique facility from "Allendale Clinic, New York, NY". Conversely, entries of "Allandale Clinic, Atlanta, GA" and "Allendale Clinic, Atlanta, GA" likely represent a misspelling. Ideally, these data should be standardized so that there is one correct representation for each unique facility. Once the data are standardized, each unique combination of facility name, city, and state can be used to create a site-specific valid value list of facilities to be used as part of the e-HARS and HARS systems to maintain the standardizations.

Together, SAS and dfPower Studio provide a solution for cleaning the HARS facility data. Using SAS, a facility key is constructed containing facility name, city, and state. With dfPower Studio, it is possible to view each existing permutation of the facility key, associate each permutation with a data standard, and apply that standard to the data. This document describes the steps involved in cleaning the HARS data.

PRE-PROCESSING: SAS

PREPARING THE DATA

1. Make a current back-up of the HARS database.

2. The HARS database must first be converted into SAS format. Variables that require cleansing are extracted and written to a SAS data set to be standardized using dfPower Studio.

3. In this example, the HARS database is assumed to be named $aids$.dbs and stored in the C:\AIDS\INPUT folder. It is expected that a SAS format library exists in the C:\AIDS\SAS\FORMATS folder. The SAS data set used by dfPower Studio is stored in the C:\AIDS folder. If these locations do not match your configuration, edit hars_pre_process_key.sas, changing the directory locations specified in the SAS libname statements.

4. Start SAS and run hars_pre_process_key.sas. A data set is created containing a new variable called fac_key. The fac_key variable is comprised of facility name, city, and state delimited by "^^^". If any of the fields that make up the key are blank, they will be represented with "..". For example, if a record contains facility name (Grady) and state (GA), but city is blank, the key would look as follows: "GRADY^^^..^^^GA". Changes may be made to any or all parts of the key, but do NOT delete or change the delimiter.

```
/*********************************************************************/
/*  Program Name:  hars_pre_process_key.sas                          */
/*  Date:          08/11/2003                                        */
/*  Author:        KAT1                                              */
/*  Purpose:       1) Converts HARS PRODAS data set into SAS data set  */
/*                    format                                         */
/*                 2) Creates aggregated data sets with specific     */
/*                    variables to be cleansed.  These variables are  */
/*                    to be used for scheme creation and             */
/*                    standardization with DataFlux dfPower Studio   */
/*                 3) Creates a "key" for each record by combining   */
/*                    facility attributes, such as name, city, state  */
/*********************************************************************/

libname prodata prodas 'c:\aids\input' status=1000 dollar=_ ;
libname sasdata v8     'c:\aids';
libname library        'c:\aids\sas\formats';

options nofmterr yearcutoff=1910;

/*********************************************************************/
/*  Create two permanent SAS data sets from the HARS PRODAS data set  */
/*  Build a facility key to identify unique facilities               */
/*  The standardization scheme will be applied to fac                */
/*  The fac_base data set will retain the original facility names and  */
/*  will be compared to fac to identify changed records              */
/*  Drop the original HARS facility name, city, and state variables  */
/*  because the standardization scheme will be applied to the key    */
/*  variable only                                                    */
/*********************************************************************/

data sasdata.fac      (drop=bhosp    chosp    hosp_dx  hhosp_dx phosp_dx
                            bhcity   chcity   hosp_cty hhsp_cty phsp_cty
                            bhosp_st chosp_st hosp_st  hhosp_st phosp_st
                            tmpfac   tmpcity  tmpst    i)
     sasdata.fac_base (drop=bhosp    chosp    hosp_dx  hhosp_dx phosp_dx
                            bhcity   chcity   hosp_cty hhsp_cty phsp_cty
                            bhosp_st chosp_st hosp_st  hhosp_st phosp_st
                            tmpfac   tmpcity  tmpst    i
                       rename=(bhosp_key = bhosp_key_base
                               chosp_key = chosp_key_base
                               hosp_key  = hosp_key_base
                               hhosp_key = hhosp_key_base
                               phosp_key = phosp_key_base));
  set prodata._aids_  (keep=stateno
                            bhosp    bhcity   bhosp_st
                            chosp    chcity   chosp_st
                            hhosp_dx hhsp_cty hhosp_st
                            hosp_dx  hosp_cty hosp_st
                            phosp_dx phsp_cty phosp_st);

  array hars_key{5}  $66 bhosp_key chosp_key hosp_key hhosp_key phosp_key;
  array hars_fac{5}  $27 bhosp     chosp     hosp_dx  hhosp_dx  phosp_dx;
  array hars_city{5} $27 bhcity    chcity    hosp_cty hhsp_cty  phsp_cty;
  array hars_st{5}   $2  bhosp_st  chosp_st  hosp_st  hhosp_st  phosp_st;

/*  Create temp fields to input missing values '..' as part of the key */
```

```
  length tmpfac  $27;
  length tmpcity $27;
  length tmpst   $2;

  do i = 1 to 5;
    if hars_fac{i}  = ' ' then
      tmpfac  = '..';     else tmpfac  = hars_fac{i};
    if hars_city{i} = ' ' then
      tmpcity = '..';     else tmpcity = hars_city{i};
    if hars_st{i}   = ' ' then
      tmpst   = '..';     else tmpst   = hars_st{i};

    hars_key{i} = (trim(tmpfac))||'^^^'||(trim(tmpcity))||'^^^'||tmpst;

/*********************************************************************/
/*  Set key to blank if there is no key data - dfPower Studio was    */
/*  standardizing the ^^^ ^^^                                        */
/*********************************************************************/

    if hars_key{i} = '..^^^..^^^..' then hars_key{i} = ' ';
  end;

run;

/*  Create a temporary SAS data set of facility keys                 */

data fac_keys (keep=fac_key);
  set sasdata.fac;

  array hars_key{5} $66 bhosp_key chosp_key hosp_key hhosp_key phosp_key;

  length fac_key $66;

  fac_key = ' ';

  do i = 1 to 5;
    if hars_key{i} ne ' ' then do;
      fac_key = hars_key{i};
      output;
    end;
  end;

run;

/*********************************************************************/
/*  Create a permanent SAS data set to be used by dfPower Studio in  */
/*  creating a standardization scheme                                */
/*********************************************************************/

proc sort data=fac_keys out=sasdata.facility;
  by fac_key;
run;
```

STANDARDIZATION AND VERIFICATION: DFPOWER STUDIO

CREATING THE STANDARDIZATION SCHEME

1. Start dfPower Studio from the desktop icon and select Analysis from the Profiling Tools toolbar to begin creating the standardization scheme.

2. Select the appropriate data source. In this example, the data source is HARS_SAS. The first time a data source is selected, dfPower Studio creates an ODBC server SAS session that will run for the duration of the dfPower Studio session. A window will appear indicating a specific amount of time needed to start the session. Do not use this SAS session for any SAS programming and do not close it until after exiting dfPower Studio. Once the data source has been selected, the names of the available data sets will appear under the data source name.

3. Select the data set created in hars_pre_process_key.sas (sasdata.facility).

4. Select the variable name (*fac_key*), to be standardized with the following specifications:

    a. Match: None
    b. Sensitivity: leave blank
    c. Kind: Phrase Analysis
    d. Type: Alphabetical

5. Begin processing by selecting the right green arrow button. The Analysis Editor window will appear.

6. From the Analysis menu, select Compare Analysis to Scheme. Select either Hide Existing Permutations or Highlight Unaccounted Permutations. Hide will remove from display all processed permutations. Highlight will display in red the permutations that have not yet been processed.

7. Build the automated scheme by selecting the correct record from the Permutation panel. The "correct" record is the actual name and spelling that will be used to standardize other records with different permutations of the same facility. When the correct record is selected, it will appear in the "with standard" box at the bottom of the Permutation panel. Edit the entry as needed in the "with standard" box. Click the "Add To Scheme" button. Select each permutation to be standardized using the record currently in the "with standard" box, and single-click the "Add To Scheme" button. Repeat the process to select each successive correct record and its corresponding permutations.

    As data is added to the scheme, it will appear in the standardization panel on the right side of the window. The data column represents each unique permutation added to the scheme. The standard column represents the corresponding standard that the permutation will be updated to reflect.

    If needed, changes may be made to the permutation-standard combinations within the standardization panel. However, if a change is made to the spelling of a standard, it will change the spelling of the standard for every permutation associated with that standard. If a permutation-standard combination is completely wrong, it can be deleted from the scheme by highlighting that combination within the standardization panel, right-clicking, and selecting delete. The selected permutation will then reappear in the permutation panel.

8. Save the scheme. From the Schemes menu, select Save Scheme as HARS facility. Save the analysis report. From the Analysis menu, select Save Analysis Report as HARS facility. Exit the Analysis Editor. From the File menu, select Exit. Save the job. From the File menu, select Save as Job as HARS facility.

9. Exit Analysis. From the File menu, select Exit.

APPLYING THE STADARDIZAION SCHEME

1. Select Standardize from the Quality Tools toolbar to apply the standardization scheme.

2. Select the appropriate data source. In this example, the data source is HARS_SAS. Once the data source is selected, the names of the available data sets will appear under the data source name.

3. Select the data set that needs to be standardized (sasdata.fac).

4. Select the variables that need to be standardized. Under the Scheme drop-down box, associate the scheme created for the facility key (HARS facility) with all of the HARS variables that store facility information: *bhosp_key, chosp_key, hhosp_key, hosp_key,* and *phosp_key*.

5. Set *stateno* as the primary key by selecting [PRIMARY KEY] under the Scheme drop-down box. To generate the list of changes by stateno, from the Control menu, select Log Updates. This step is not required, but will speed up processing and provide a detailed list of changes by *stateno* that may be used to verify standardizations.

6. Specify output requirements by selecting from the Control menu, Standardization Target Options. Select Current Source Table. This will apply the changes to sasdata.fac.

7. Begin processing by selecting the right green arrow button.

VERIFYING STANDARDIZATIONS

1. From the File menu, select View Statistics. If *stateno* was set as the primary key and the Control menu Log

Updates option is on, a detailed list of changes by *stateno* may be viewed. Print the statistics. From the File menu in the Statistics window, select Print.

2. Save the job, if desired. From the File menu, select Save As Job.

3. Exit dfPower Studio.

POST-PROCESSING: SAS

UPDATING THE DATA

1. The cleansed data must be converted to ASCII format and loaded into HARS.

2. Make a backup of the HARS database before beginning the ASCII load.

3. If changes were made to the directory locations specified in the SAS libname statements in hars_pre_process_key.sas, edit hars_post_process_key.sas to reflect those same changes.

4. Start SAS and run hars_post_process_key.sas to write the newly cleansed SAS data to an ASCII file (sasdata.fac_out.txt) for importation into HARS.

```
/*********************************************************************/
/*  Program Name:  hars_post_process_key.sas                         */
/*  Date:          08/12/2003                                        */
/*  Author:        KAT1                                              */
/*  Purpose:       Converts standardized facility SAS data set into  */
/*                 ASCII file in preparation for importation into HARS */
/*********************************************************************/

libname sasdata v8 'c:\aids';
libname library    'c:\aids\sas\formats';

filename fac_out   'c:\aids\fac_out.txt';

options nofmterr yearcutoff=1910;

/*********************************************************************/
/*  Sort the dataset to which the dfPower Studio standardization     */
/*  scheme was applied                                               */
/*********************************************************************/

proc sort data=sasdata.fac;
  by stateno;
run;

/*********************************************************************/
/*  Sort the dataset containing the original facility values found in  */
/*  the HARS record                                                  */
/*********************************************************************/

proc sort data=sasdata.fac_base;
  by stateno;
run;

/*********************************************************************/
/*  Merge the standardized data set with the original data set to    */
/*  identify and write records where facility was changed            */
/*********************************************************************/

data sasdata.fac_changed_recs;
  merge sasdata.fac
        sasdata.fac_base;
  by stateno;

/*********************************************************************/
```

```
/*  Recreate HARS facility name, city, and state variables from the    */
/*  key variables so any changed values can be loaded into HARS        */
/*  Recreate key from extracted parts to make sure formatting matches  */
/*  base key                                                           */
/*********************************************************************/

  array hars_key{5}  $66 bhosp_key chosp_key hosp_key hhosp_key phosp_key;
  array hars_fac{5}  $27 bhosp     chosp     hosp_dx  hhosp_dx  phosp_dx;
  array hars_city{5} $27 bhcity    chcity    hosp_cty hhsp_cty  phsp_cty;
  array hars_st{5}   $2  bhosp_st  chosp_st  hosp_st  hhosp_st  phosp_st;

  do i = 1 to 5;

    hars_fac{i}  = ' ';
    hars_city{i} = ' ';
    hars_st{i}   = ' ';

    if hars_key{i} ne ' ' then do;
      hars_fac{i}  =      scan(hars_key{i}, 1, '^^^');
      hars_city{i} = left(scan(hars_key{i}, 2, '^^^'));
      hars_st{i}   = left(scan(hars_key{i}, 3, '^^^'));
      hars_key{i}  = (trim(hars_fac{i})) || '^^^' ||
                     (trim(hars_city{i}))|| '^^^' || hars_st{i};
    end;

    if hars_fac{i}  = '..' then hars_fac{i}  = ' ';
    if hars_city{i} = '..' then hars_city{i} = ' ';
    if hars_st{i}   = '..' then hars_st{i}   = ' ';

  end;

       if bhosp_key ne bhosp_key_base then output sasdata.fac_changed_recs;
  else if chosp_key ne chosp_key_base then output sasdata.fac_changed_recs;
  else if hhosp_key ne hhosp_key_base then output sasdata.fac_changed_recs;
  else if hosp_key  ne hosp_key_base  then output sasdata.fac_changed_recs;
  else if phosp_key ne phosp_key_base then output sasdata.fac_changed_recs;

run;

/* Write changed records to an ASCII file to be loaded into HARS       */

data _null_;
  set sasdata.fac_changed_recs;
  file fac_out lrecl=290;

  if _n_ = 1 then do;
    put "stateno,1,10";
    put "bhosp,11,27";
    put "bhcity,38,27";
    put "bhosp_st,65,2";
    put "chosp,67,27";
    put "chcity,94,27";
    put "chosp_st,121,2";
    put "hosp_dx,123,27";
    put "hosp_cty,150,27";
    put "hosp_st,177,2";
    put "hhosp_dx,179,27";
    put "hhsp_cty,206,27";
    put "hhosp_st,233,2";
    put "phosp_dx,235,27";
    put "phsp_cty,262,27";
    put "phosp_st,289,2";
    put "*";
  end;
```

```
  put @1   stateno  $char10.
      @11  bhosp    $char27.
      @38  bhcity   $char27.
      @65  bhosp_st $char2.
      @67  chosp    $char27.
      @94  chcity   $char27.
      @121 chosp_st $char2.
      @123 hosp_dx  $char27.
      @150 hosp_cty $char27.
      @177 hosp_st  $char2.
      @179 hhosp_dx $char27.
      @206 hhsp_cty $char27.
      @233 hhosp_st $char2.
      @235 phosp_dx $char27.
      @262 phsp_cty $char27.
      @289 phosp_st $char2.;
run;
```

MAINTAINING THE NEW STANDARDS

1. The HARS system supports the use of a list of valid values for the facility fields: *bhosp, chosp, hhosp_dx, hosp_dx,* and *phosp_dx*. When a value is entered into one of those fields, it is checked against the valid values for that field. A warning is displayed if the value entered does not match one of the valid values. This feature will help maintain the standards implemented for these fields and keep the data clean.

2. The valid value lists are simple ASCII files containing each valid permutation for the field. Once all of the above steps have been performed, and the data for the facility fields are clean, an ASCII file of valid values can be created by starting and running hars_create_lis_key.sas.

```
/*********************************************************************/
/*  Program Name:  hars_create_lis_key.sas                           */
/*  Date:          08/06/2003                                        */
/*  Author:        KAT1                                              */
/*  Purpose:       Creates HARS PRODAS .lis files for facility       */
/*********************************************************************/

libname sasdata v8 'c:\aids';
libname library    'c:\aids\sas\formats';

filename fac_lis   'c:\aids\fac_lis.txt';

options nofmterr yearcutoff=1910;

/*********************************************************************/
/*  Create a data set of updated facilities from standardized data set */
/*  Output only non-blank records                                    */
/*********************************************************************/

data fac_names (keep=fac_name);
  set sasdata.fac;

  array hars_key{5} $66 bhosp_key chosp_key hosp_key hhosp_key phosp_key;

  length fac_name $27;

  fac_name = ' ';

  do i = 1 to 5;
    if hars_key{i} ne ' ' then do;
      fac_name = scan(hars_key{i}, 1, '^^^');
      if fac_name = '..' then fac_name = ' ';
      if fac_name ne ' ' then output fac_names;
    end;
  end;
```

```
run;

proc sort data=fac_names nodupkey;
  by fac_name;
run;

/*  Create an ASCII file of unique facility names                    */

data _null_;
  set fac_names;

  file fac_lis;

  put @1 fac_name $char27.;

run;
```

CONCLUSION

dfPower Studio has proved to be invaluable in preparing the HARS legacy data for migration to a new system. dfPower Studio provides the tools to efficiently standardize large amounts of data. Following the procedures outlined in this document, SAS and dfPower Studio are used to create and implement data standards as well as maintain those standards. These processes eliminate data inconsistencies and ensure the correct representation of each unique facility name.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Kimberly A. Todd
Centers for Disease Control and Prevention
1600 Clifton Road, N.E. (Mailstop E-48)
Atlanta, GA 30333
Work Phone: (404) 639-2033
Fax: (404) 639-8642
Email: kat1@cdc.gov