

4090-2020

Dynamic Programming: Using Files to Change the Behavior of Your Program

Bryan K. Merrell, **Merrick Bank**

ABSTRACT

Dilemma:

- A company much larger than mine was providing information on how data was to be received. Their documentation was over a thousand pages of data transforms and usage and could change at any time.
- You have very little influence on this 3rd party

Solution

- Become more pro-active.
- Create a file that lists out all field transforms to be able to change those values as they change from the provider.

INTRODUCTION

A 3rd party that works with over almost a hundred thousand businesses has an automated data reporting tool that you need to use. They provide a total of almost ten thousand values in the data they return. Just one value in this tool has over thirty thousand different transforms. The file they return is a random length per row with descriptors that say what length each section is:

- **AU23awe146orgin11asof1987igINA25orihaadvalue27houichhsx45....**
 - AU is a type of data coming in. 23 is the length of that type
 - INA is a second type of data coming in. 25 is the length of that type

Each type is then divided by further descriptors to get data.

Getting any specific value requires a look up of the 3rd **party's** documentation. This could be a many to one relationship between the data we have and the data being looked for from the 3rd party. The overall file has over thirty thousand combinations that can be derived into one value.

- These values can change at any time from the 3rd party.

We saw this and changed our code to be more dynamic to minimize our cost of our changes when there were 3rd party changes.

FIRST STEPS IN DYNAMIC PROGRAMMING

Planning and preparation.

- Get the data needed from the 3rd party
- Create a program to import and clean the 3rd party data

- This program will be used each time you get new data from the 3rd party
- Import program will need to clean the data for use (Shown later)
- Modify your data handling program (PRGM2) to use the cleaned 3rd party data
- Now changes to the data from the 3rd party are easily (Dynamically) changing the behavior of your data handling program.

```
libname trsfrm XLSX "X:\FileLoc\Input to Output Processing.xlsx";
libname trsfrmh "X:\Fileloc";

data stats;
  format Process_code INPUT_VALUE $17. G date_closed $40.;
  set trsfrm.X05 trsfrm.X11 trsfrm.X13 trsfrm.X61 trsfrm.X62 trsfrm.X63 trsfrm.X64
  trsfrm.X65 trsfrm.X01_02_03_05_08_12
  trsfrm.X08 trsfrm.X09 trsfrm.X31 trsfrm.X41 trsfrm.X51 trsfrm.X61 trsfrm.X71
  trsfrm.XGA;
run;
```

First step in transforming following data:

process code	payment sec	credit code	credit code 2	consumer id	date open	date closed	high	highest	bal	amt pd	comment code	CComCod	Pattern
01,02,03,05,08,12	n/a				val rep	val rep	val rep	val rep	val rep	val rep		52,53,55,57,78,88,91	val rep
01,02,03,05,08,12	n/a	A			val rep	val rep	val rep	val rep	val rep	val rep	DFG	52,53,55,57,78,88,91	val rep
01,02,03,05,08,12	n/a	C			val rep	val rep	val rep	val rep	val rep	val rep	EFG	52,53,55,57,78,88,91	val rep
01,02,03,05,08,12	n/a	E			val rep	val rep	val rep	val rep	val rep	val rep	MFG	52,53,55,57,78,88,91	val rep
01,02,03,05,08,12	n/a	G			val rep	val rep	val rep	val rep	val rep	val rep	EFG	52,53,55,57,78,88,91	val rep
01,02,03,05,08,12	n/a	FI			val rep	val rep	val rep	val rep	val rep	val rep	DFG	52,53,55,57,78,88,91	val rep
01,02,03,05,08,12	n/a	FO			val rep	val rep	us FZ if pres	val rep	0	0	TRF	52,53,55,57,78,88,91	val rep
01,02,03,05,08,12	n/a	FM			val rep	val rep	val rep	val rep	val rep	val rep	DFG	52,53,55,57,78,88,91	val rep
01,02,03,05,08,12	n/a	FT			val rep	val rep	val rep	val rep	val rep	val rep	EFG	52,53,55,57,78,88,91	val rep
01,02,03,05,08,12	n/a	FZ			val rep	val rep	val rep	val rep	val rep	val rep	MFG	52,53,55,57,78,88,91	val rep
01,02,03,05,08,12	n/a	GG			val rep	val rep	val rep	val rep	val rep	val rep	PRB	52,53,55,57,78,88,91	val rep
01,02,03,05,08,12	n/a	GZ			val rep	val rep	val rep	val rep	val rep	val rep	AFG	52,53,55,57,78,88,91	val rep

All data will need to be transformed to the way your organization needs it.

DATA CLEANING

```
data stats2;
  set stats;
  format status $2.;
  i=1;
  do until(status='');
    status = scan(process_code,i,',');
    if status ~= '' then output;
    i+1;
  end;
  drop i process_code;
run;
```

The above code will break out the multiple process codes in on field into on process code per row. Rinse and repeat for each field. Cleaning the data into a format that your organization needs. This is just an example of one type of transformation. Data transformation would need its own presentation but simply put, divide the transformation into very small steps and then code each step.

NOW THAT THE TRANSFORMATION CODE IS WRITTEN

You can simply insert the transforms **XLSX file, validate, and you're done**, unless the input data format changes. Even if they do change their formats, all I have to do is get the data in a comparable format. Much less work than changing things manually

So far, for the last 2 years, I have been able to just upload their updates into this program and I am good. The program is dynamic because it uses a file to change the needed behavior due to a 3rd party change.

Usage of the transformed data is easy. Here is an example of one of my Dynamic programs:

```
proc sql;
  create table with_trsfm as
  select a.*, coalesce(b.credit_code,'Not Found') as Credit_code
  from comparisons3 a
  left join trsfm.transform b on strip(a.comment_code) = strip(b.comment_code)
  and strip(a.CCom) = strip(b.CComCod)
  and strip(a.ACCT_Process_CD) = strip(b.status)
  and strip(a.KeyVal) = strip(b.KeyValue)
  and strip(a.CC2) = strip(b.Credit_Code2);
quit;
```

In above code:

Trsfm.transform is the file generated in prior pages

Comparisons3 is my data pull from my data

Now I can test my data up against what the 3rd party has to find any discrepancies or exceptions on either side.

RESULTS OF DYNAMIC PROGRAMMING

- I have one program has 32 behavior changing files
- Changes to the files then changes the end results of the program
- Each of these programs has reduced my maintenance needs
- I can do more with less.

This makes my boss happy!!



RESOURCES

The resources that I use most are:

SAS® 9.4 FUNCTIONS AND CALL ROUTINES: REFERENCE, FIFTH EDITION

<https://documentation.sas.com/?docsetId=lefunctionsref&docsetTarget=titlepage.htm&docsetVersion=9.4&locale=en>

SAS(R) 9.4 Formats and Informats: Reference

<http://support.sas.com/documentation/cdl/en/leforinforref/64790/HTML/default/viewer.htm#p0z62k899n6a7wn1r5in6q5253v1.htm>

Carpenter's Complet Guide to the SAS Macro Lanuage

Third Edition, copyright 2016

by Art Carpenter

And above all: The people I work with at Merrick Bank.