PSID A national study of socioeconomics and health over lifetimes and across generations

Technical Series Paper #12-03

Programming and Use of Blaise to SAS Transformation Tool for Streamlining Processing in the Panel Study of Income Dynamics

(Presented at the International Blaise Users Group Conference, April 25, 2012, London, UK)

April Beaule and Mohammad Mushtaq
Survey Research Center - Institute for Social Research
University of Michigan

This project was supported by funding from the National Science Foundation (SES 0518943).

Programming and Use of Blaise to SAS Transformation Tool for Streamlining Processing in the Panel Study of Income Dynamics

April Beaulé and Mohammad Mushtaq, University of Michigan

1. Introduction

The Panel Study of Income Dynamics (PSID) is a nationally representative longitudinal study of approximately 9000 U.S. families. Since 2001, the PSID has used Blaise as its main software for the data collection instrument. Due to size and complexity of the instrument, the PSID staff developed a tool to assist with extraction and transformation of initial files. This tool, referred to as Data Extraction, Transformation and Loading (DETL), streamlines first step processing of Blaise files.

This paper will discuss the development and use of the DETL tool in the processing cycle, specifically providing an overview of the development and how we improve efficiency and save time setting up first step files for further processing.

2. Background

In the early years of the study, the data collection tool was a paper and pencil questionnaire. In 1992, the PSID automated the survey instrument using SurveyCraft software. By 2001, the Survey Research Center at the University of Michigan began using Blaise and the PSID was re-programmed at that time in the new software. Just as the sample continues to grow, so has the instrument. In 1968, the first year of the PSID the interview took approximately 20 minutes to administer and the staff released 447 variables. By 2009, the interview length increased to an average of 89 minutes and we have released over 5012 variables in this last wave.

In order to make the processing of first step files more efficient, the DETL tool was developed. This tool allows the user to take advantage of the Blaise meta information including question text, code frames, loop levels and frequencies in order to bundle sets of variables together in a logical way.

The end product from DETL are SAS programs that the user produces by grouping together sets of variables from the basic output tables (extracted at the block level), and sub-setting for certain types of individuals when necessary.

3. Display of Meta Information

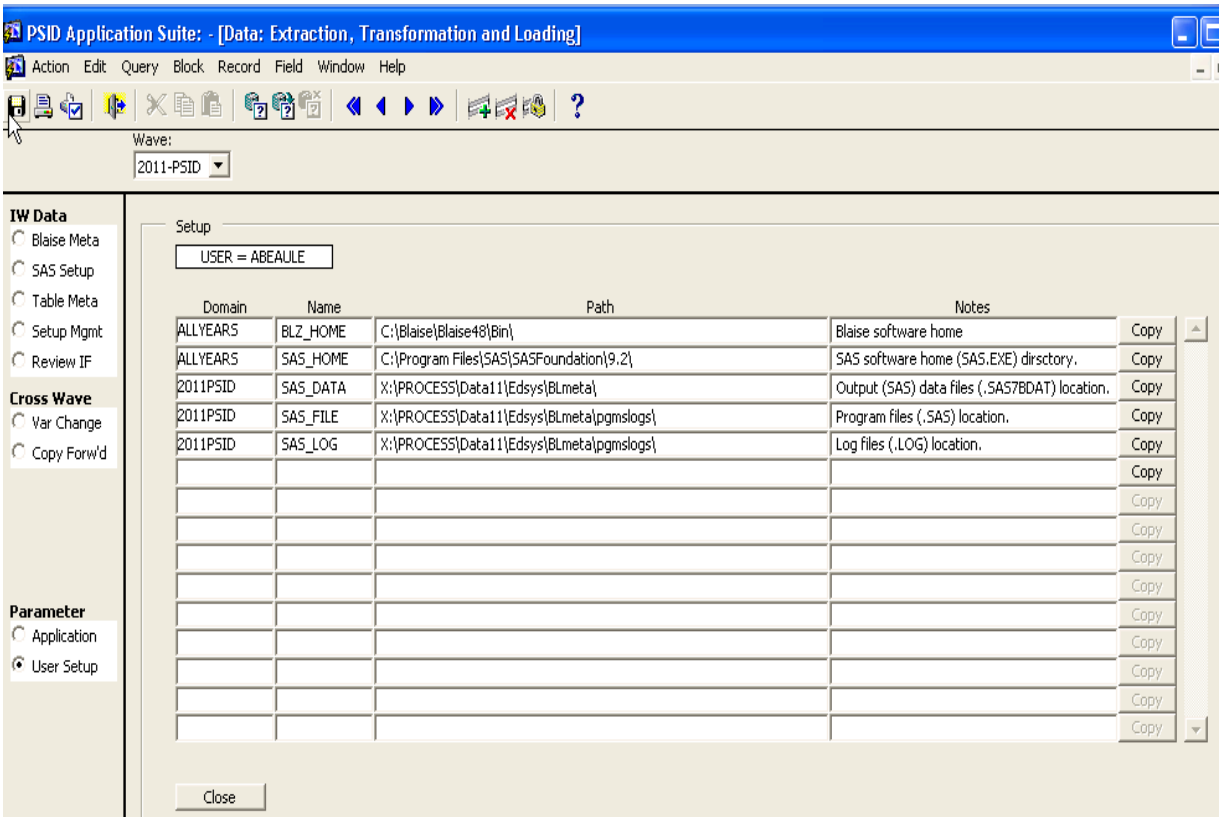
The main purpose of first step files is to aggregate variables together at the content level and at the person level. The PSID sample is at the family level but within the instrument we ask more questions about certain people in the family, we label those ‘special’ people Head and Wife (if there is one). We ask fewer sets of questions about the rest of the family members and we label those individuals Other Family Unit Members (OFUM’s). For Blaise programming those questions which are the same for Head and Wife and

sometimes OFUM's are looped. However, for processing we have to break them apart. Using DETL we can subset the types of records we want to focus on and label the SAS setups and output resulting from the sub-setting using 'Head' or 'Wife' labels.

The basic extraction from Blaise produces tables at the block level. For Blaise programming purposes often to maximize run time speed, questions are grouped together in blocks. For processing and editing however, we need to group variables in alternative ways.

The first step for the user is to define the directories for a given year. The user defines the location where he or she wants to copy the output data as well as where the .SAS programs and .LOG files are to be written (Figure 1).

Figure 1: User Definition for DETL



Once the user has defined the parameters, they begin by reviewing the meta information about the variables and begin selecting variables for setup. The user clicks the radio button to the left under IW Data, 'Blaise Meta' (Figure 2.1). Displayed under Blaise Meta is a complete list of all the SAS files extracted by block for this BMI. The name of the output file is the name of the block as defined by the Blaise programmer. For PSID in 2009 for the main BMI, there were one hundred and twenty-three tables extracted. Variables from all these tables need to be merged together and sub-set in a coherent format for processing.

To assist the user with choosing the correct variables for each setup, the Blaise Meta screen allows the user to click on a block in the upper portion of the window (Figure 2.1). Down in the bottom pane, all the

variables in this block are displayed along with their meta information. Helpful information such as the type of variable (character or numeric), the width, the number of loops and whether 'Don't Know' and 'Refused' were allowed for this variable (Figure 2.2). In the check boxes in the center of the lower pane, there are options to view the question text (Figure 2.1), the code frame, frequencies and the universe for every variable in the block selected.

Figure 2.1: Blaise Meta Display – Question Text

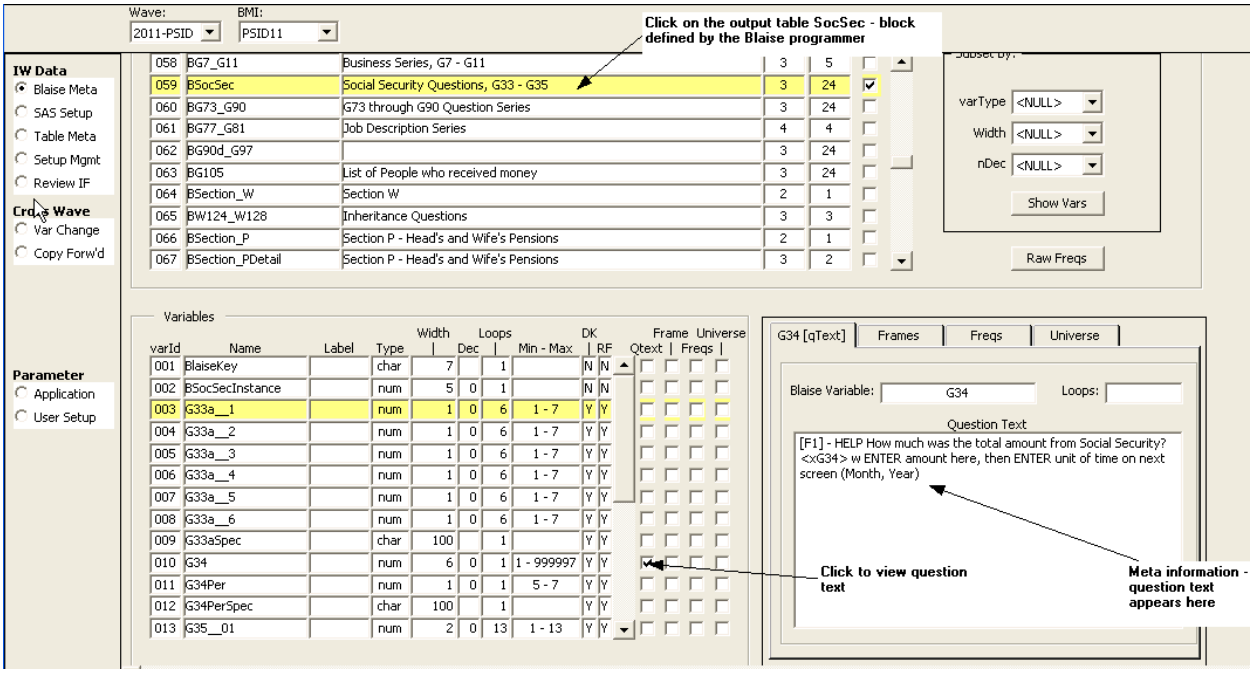
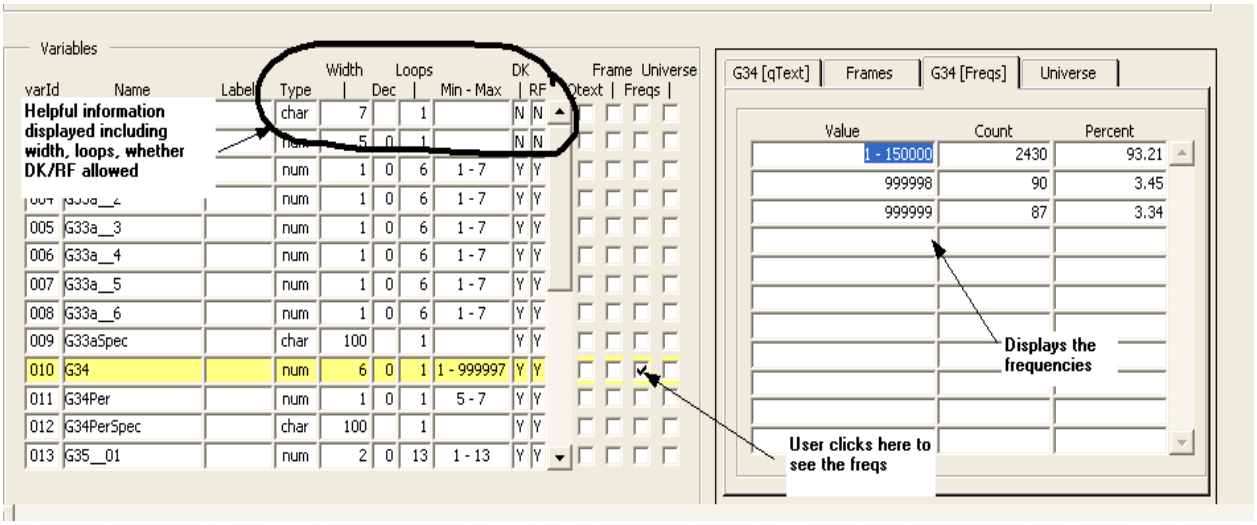


Figure 2.2: Blaise Meta Display – Frequencies



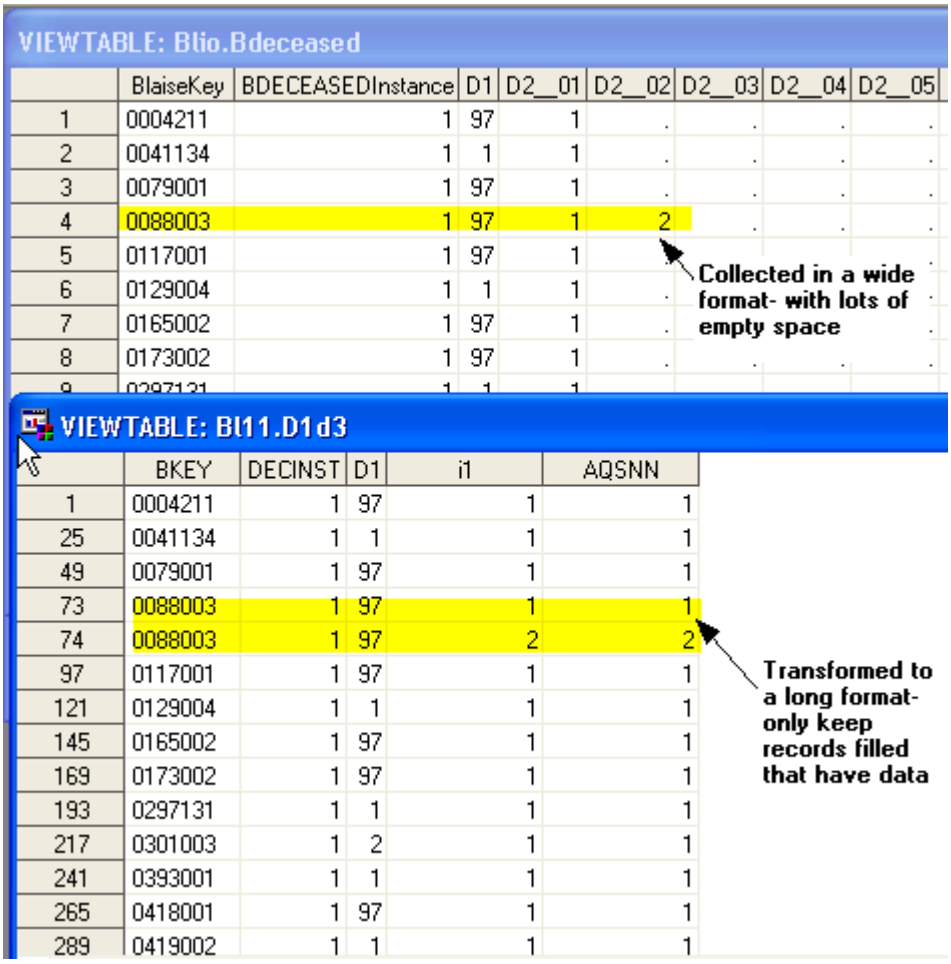
4. SAS Setups

Once the user has decided on the tables and variables they need based on the meta information display, they turn their focus to defining the SAS setups. There are three basic types of SAS setups available: (a)

Long, (b) Wide, and (c) Merge. The Long format setup takes a file that is collected in a wide format and flips it vertically (Figure 3). We end up with a more compact file especially when the data we collect is formatted using our standard 24 person array but we only ever have a couple of mentions.

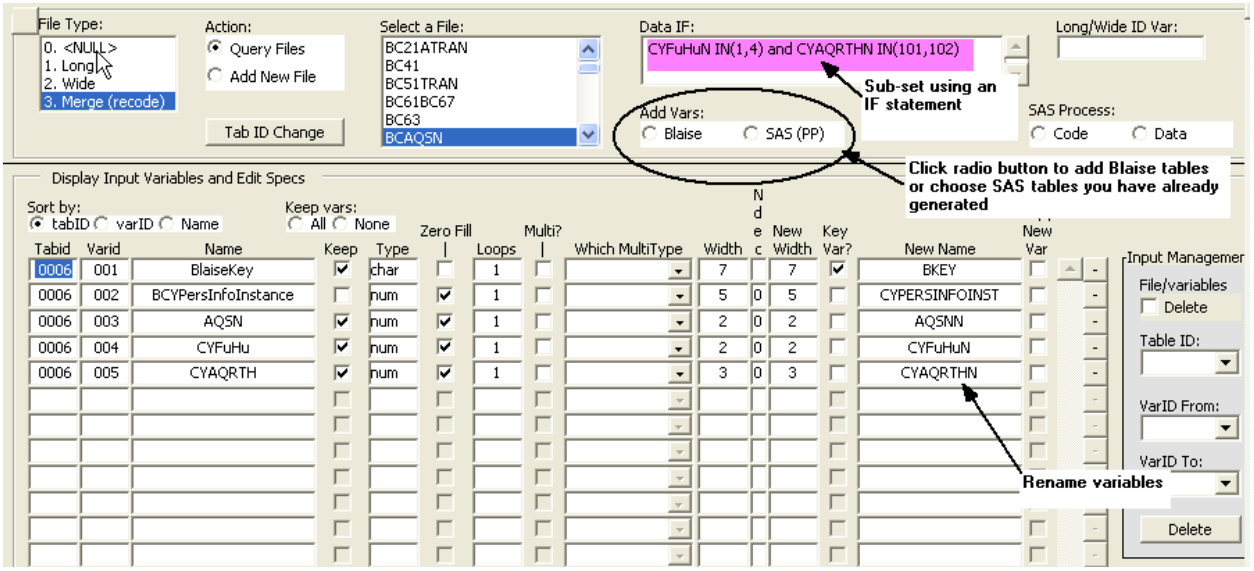
The Wide format set up transforms in the opposite way. It takes data collected in a looped format and widens it out so that there is one record per case. For example, we use this format for mortgages. We collect a maximum of two. but we release it flattened with mortgage one variables followed by mortgage two variables. The SAS syntax that is required to produce these two transformations is lengthy. DETL is able to produce SAS code quickly, accurately with just a few clicks from the user.

Figure 3: Example of SAS Table Transformation – Wide to Long



The most commonly used set up is the Merge type. The user is selecting tables extracted at the block level and merging them with other similar content variables. The Merge type set up is flexible. The user can merge together raw tables directly from Blaise or newly created DETL tables can be merged with Blaise tables (Figure 4). The user must define the merge keys between the tables. He or she can also rename variables, prefill system missing data with zeros, and subset for specific record types.
One of the examples in the PSID using a merge setup is the creation of the housing variables. We ask about mortgages and foreclosures up to a maximum of two in a looped format. First we widen the loop and then we merge this wide table back with the anchor table so that we end up with one record for every case with the housing variables even if the family had neither a mortgage nor a foreclosure.

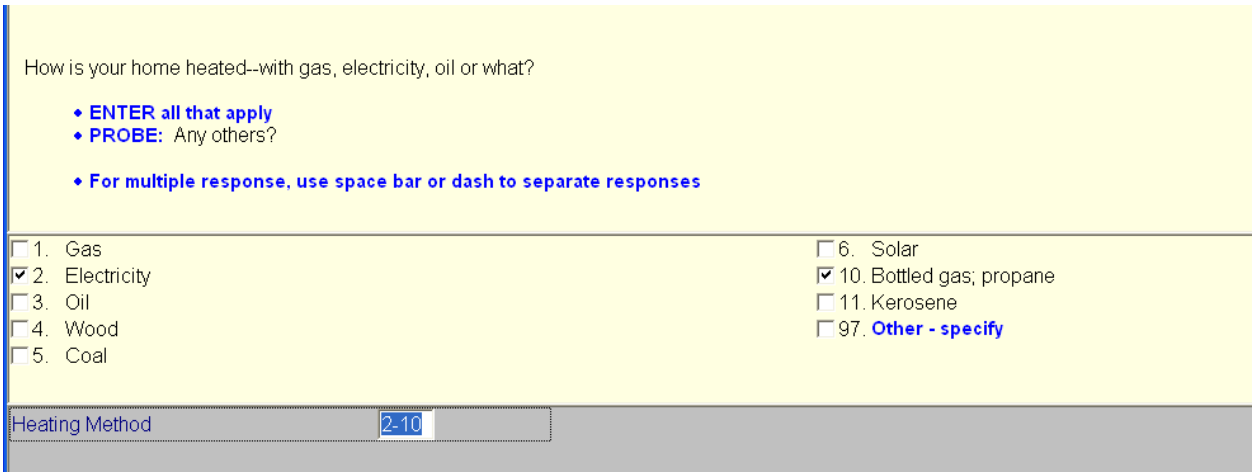
Figure 4: DETL Merge Type Setup Screen



5. Handling Multi-Mention Variables

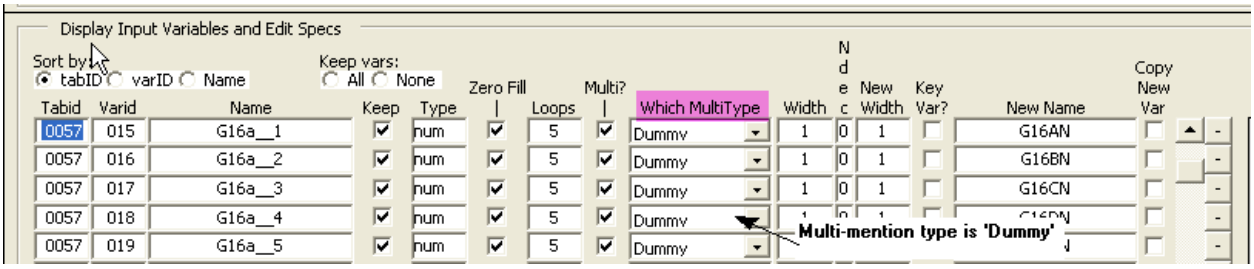
The PSID has many multi-mention type variables. DETL allows them to be transformed into the output type needed quickly and easily. The first type output we may need is the standard mention order type. The question asks 'How is your home heated – with gas, electricity, oil or what?'

Figure 5: Multi-mention question



In DETL, if we like to have the output as mentions, we simply mark the multi-mention type to 'OK' meaning 'as is' in mention order. But more frequently analysts do not want to see mentions displayed in order of how they were chosen but rather, they want to see if the respondent said yes to any of the options. In DETL, there is 2nd option for multi-mentions 'Dummy'. If one chooses to transform to a dummy indicator, the output for each mention is displayed as a True/False type (Figure 6).

Figure 6: Multi-mention question - Dummy

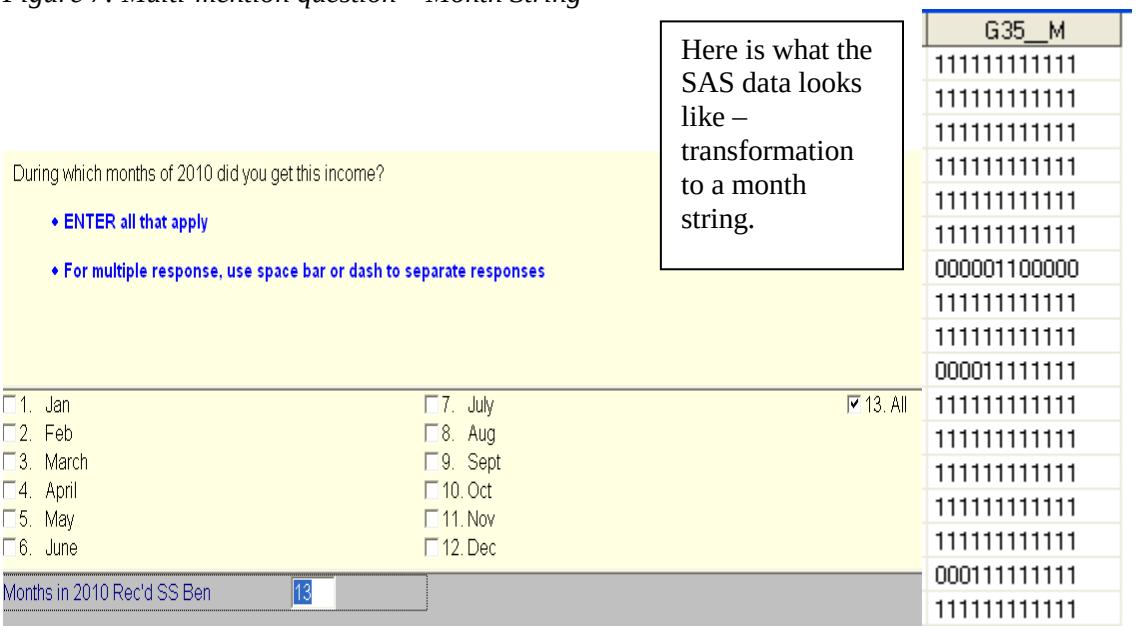


G16AN	G16BN	G16CN	G16DN	G16EN
0	0	0	0	0
0	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	1	0
0	0	0	0	0
1	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Here is what the SAS data looks like – rather than mention order, it is a True/False type whether it was selected.

A third and very common multi-mention type we use in the PSID is for months. We ask a question like 'How much did you receive for other welfare?' and then the follow up question 'During which months was that?' The interviewer can select the relevant months or they can choose code 13- all the months. For editing, we like to see it as a month string. Editors can quickly and easily see which months the respondent was receiving income and can swiftly and accurately correct months as necessary during the editing phase (Figure 7).

Figure 7: Multi-mention question – Month String



6. Producing SAS Code, Log and Tables

Once the data manager is satisfied with their setups, they simply press a radio button to either generate the SAS program (so that it may be run from SAS itself) or they can choose to both generate the SAS program, run it producing the SAS table and save the log all at the same time. The program, output table and logs are all saved to the location determined in the parameter setup.

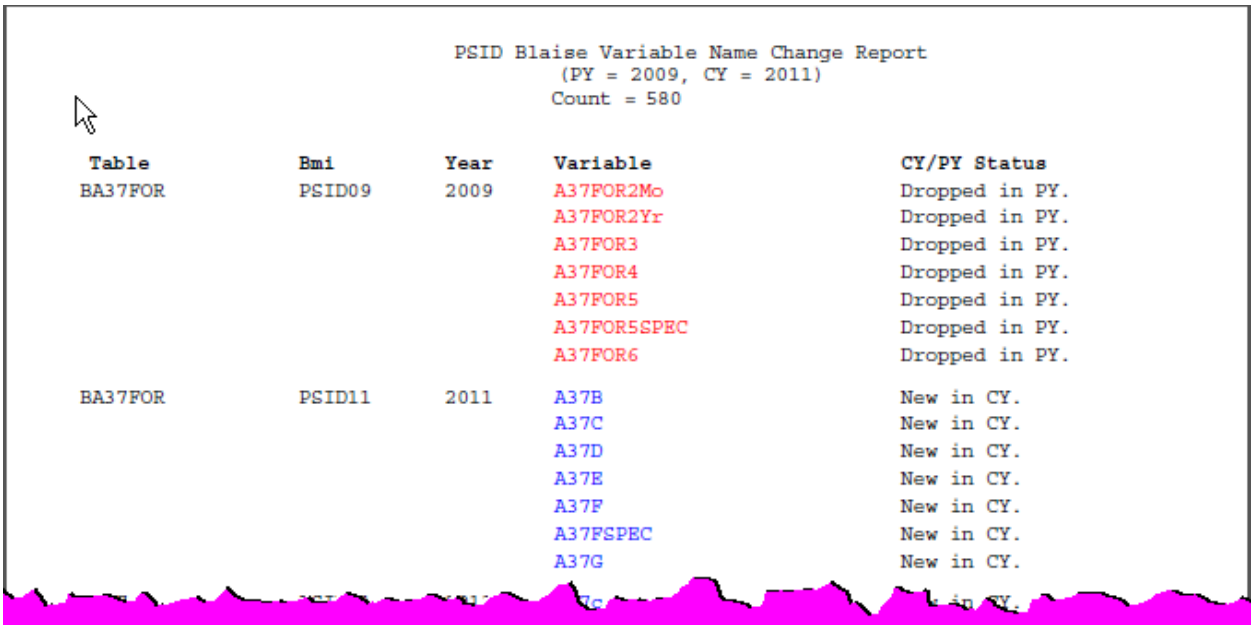
7. Pulling Setups Forward for the Next Wave

As a panel study, the PSID continues to have a need to be able to take advantage of all the data preparatory work in the prior wave. Because a great deal of effort goes into the DETL setups, our programmer was able to bring setups forward into the next wave; the caveat being that variables that were dropped disappear from the setups for the current wave and variables that are new need to be added to the appropriate existing setups.

DETL provides the user with critical between wave reports. The first report indicates which variables have been dropped and which variables have been added. The change report can be generated at the study, bmi or block level. A total of 580 variables were changed from main instrument (PSID_Main.bmi) between 2009 and 2011 waves of data collection, and 687 variables were changed from all bmi's. Figure 8 below lists variables added and dropped from the foreclosure (BA37FOR) module/block. By using this detail report, the setups for the current wave can be carefully edited (Figure 8).

Similarly a second report alerts the user to width changes so that adjustments can be made in the current year setups where the variable name has remained the same but the width has been changed.

Figure 8: Variable change report – between 2009 and 2011



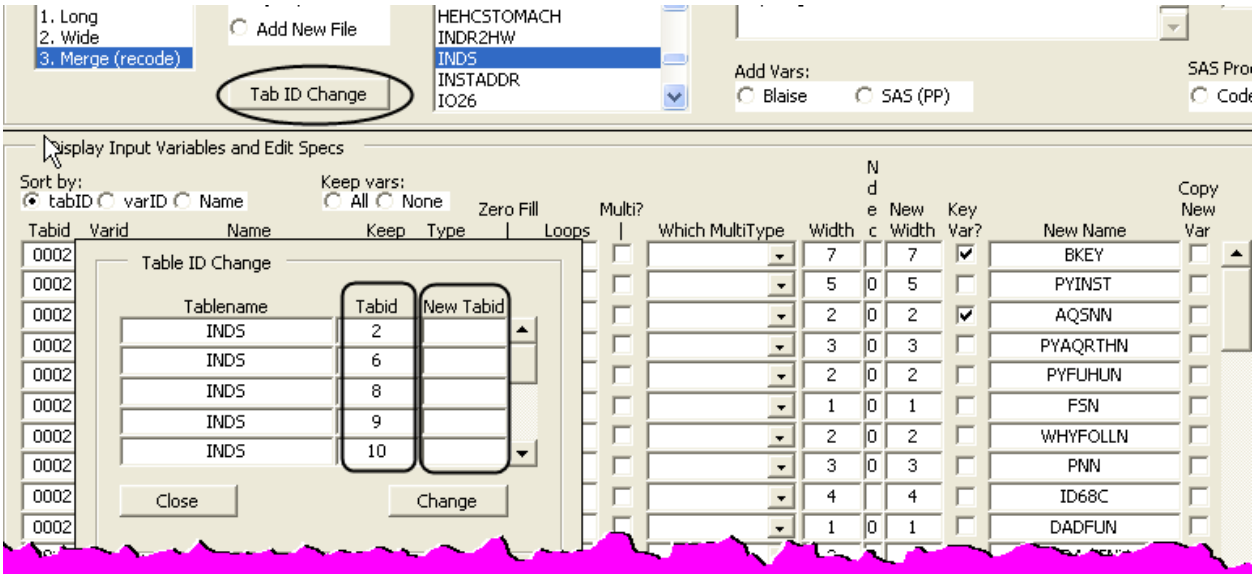
PSID Blaise Variable Name Change Report
(PY = 2009, CY = 2011)
Count = 580

Table	Bmi	Year	Variable	CY/PY Status
BA37FOR	PSID09	2009	A37FOR2Mo	Dropped in PY.
			A37FOR2Yr	Dropped in PY.
			A37FOR3	Dropped in PY.
			A37FOR4	Dropped in PY.
			A37FOR5	Dropped in PY.
			A37FOR5SPEC	Dropped in PY.
			A37FOR6	Dropped in PY.
BA37FOR	PSID11	2011	A37B	New in CY.
			A37C	New in CY.
			A37D	New in CY.
			A37E	New in CY.
			A37F	New in CY.
			A37FSPEC	New in CY.
			A37G	New in CY.

8. Table Number Changes between Pretest and Production

The use of DETL starts immediately when data are delivered from test or pretest stages of data collection instrument or questionnaire. After uploading metadata information into the Oracle database, SAS setups and data files are created to analyze data from various development cycles. Based on this analysis, the Blaise data model changes significantly during an instrument development cycle: for example to add new blocks or variables or update a code frame. These structural adjustments modify the table order and thus void the prior SAS setups created by the data user (Figure 9).

Figure 9: Table number changes between stages of instrument development

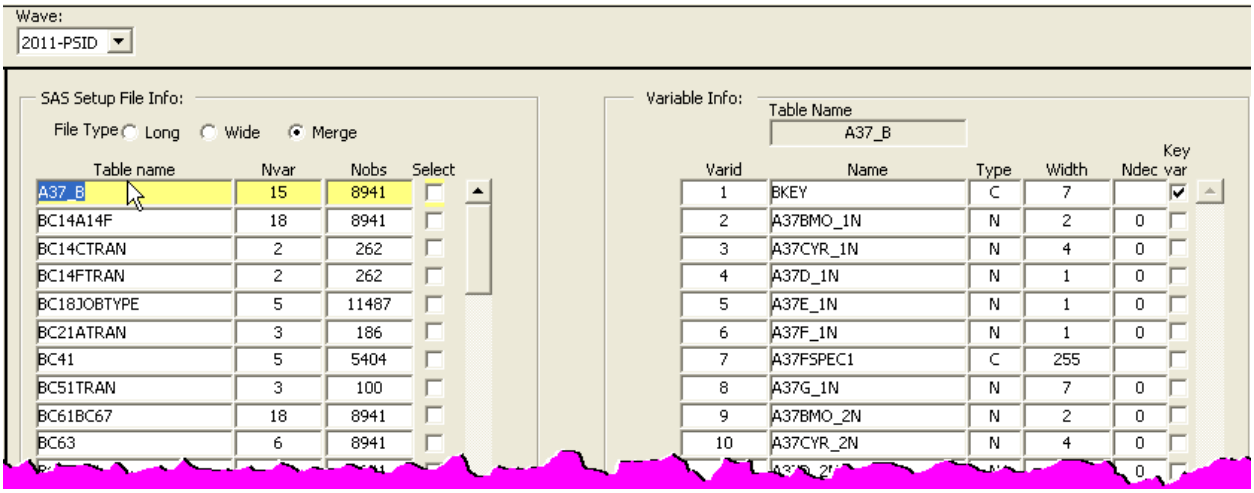


If the setup is based on a single table and it has few variables, then it can be updated manually. However if the setup consists of several variables from many tables, manual editing is time consuming and is prone to error. The "Tab ID Change" function of DETL is an elegant way of making such changes. As shown in Figure 9, the "Tab ID Change" function pops up a small window with three columns. Using Tabid to New Tabid map created from old and new bmi's, the table numbers can be changed by providing values to the New Tabid column. After applying changes, users can create new SAS setups and data files.

9. Miscellaneous Features

DETL collects metadata information about the setup created by the users. As mentioned earlier, users can create setups only or setups and data. The metadata information is displayed in a master-detail relationship as shown in Figure10 below.

Figure 10: Metadata of SAS setup files

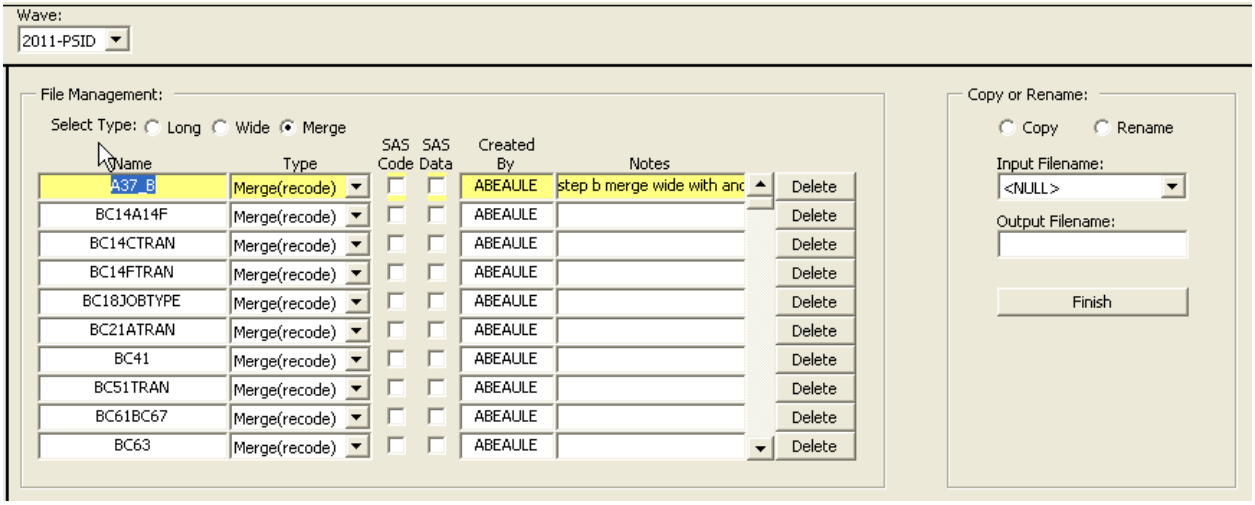


For PSID 2011, there are 203 files created by DETL, out which 19 are long, 7 are wide and 177 are merge type/format. The number of variables (nvar) and number of observations (nobs) are displayed in the left hand side block of figure 10. The variable information, (order, name, type, width, ndec, etc.), from the selected file is displayed in the right hand side panel. This information is helpful to users as they are able to see the structure of output data tables from the same interface and make changes to the setups if needed.

DETL also offers file management utility, where file setup can be renamed or copied to a new name. User can add processing notes to a file or can delete files that are no longer needed. For further details about this interface. See Figure 11 on the next page.

Another useful feature of DETL is the collection of "data if statements" in one tab together where users can review them with proper context (i.e., along with other files) and make changes if needed. When satisfied, then one can proceed to creating SAS setup and data files again. A modified 'if statement' may change the number of observations. The process will update file metadata and users should check meta information before moving to the next steps.

Figure 11: File management utility



10. Data Model and Interface Development (Programming Details)

DETL was developed using Oracle Forms/Reports 6i and Oracle Database 9.2. The development process started in 2003 and the application went into production in early 2004. Later on, it was customized for other studies and additional features were added as needed by the data processing team.

To develop this tool, the first task is to extract metadata from Blaise data model (bmi) and merge it with metadata extracted from the dataout tables (SAS tables). During this process, the looped variables need special attention because there is a one-to-many relationship between the bmi variables and the dataout variables. Then, the next task is to develop relational data model or schema that includes tables, views, functions and procedures.

Figure 12.1 describes the relationship among Blaise-to-SAS tables. A new row is inserted into blz_domain for each wave of data collection. Then blz_tables, blz_variables and blz_fmt_map are populated. Blz_freqs table is updated during data extraction process and information from this table is used to create SAS setups as discussed earlier in Sections 3 and 4. Blz_questions and blz_universe are part of the relationship. The information from these tables is joined with blz_variables using a one-to-many relationship between PVNAME and VNAME. Similarly, SAS format names are extracted from SAS data tables and then information is linked back to Blaise TCodes using blz_fmt_map and blz_formats. It should be noted that the data model is not normalized. Social science data are problematic to model into a normalized relational database and we hope to achieve this goal during next upgrade.

Figure 12.2 describes the relationship among SAS setup tables, specified in Section 4 and further explained in Sections 5 and 6 of this paper. The SAS setup module is using an Oracle database for storage and interview data are in SAS tables. The next step is to integrate the user interface with SAS, so that users are able to send tasks to SAS and get the results back to the calling environment. The bridge between SAS and Oracle is created by a SAS/Access interface to Oracle.

As explained in previous sections, this tool creates a SAS program per user specifications and it is submitted to SAS for data processing. The application uses SAS/Macros in the background to generate SAS code and calls SAS to execute the program.

Figure 12.1: ER diagram of Blaise and dataout tables

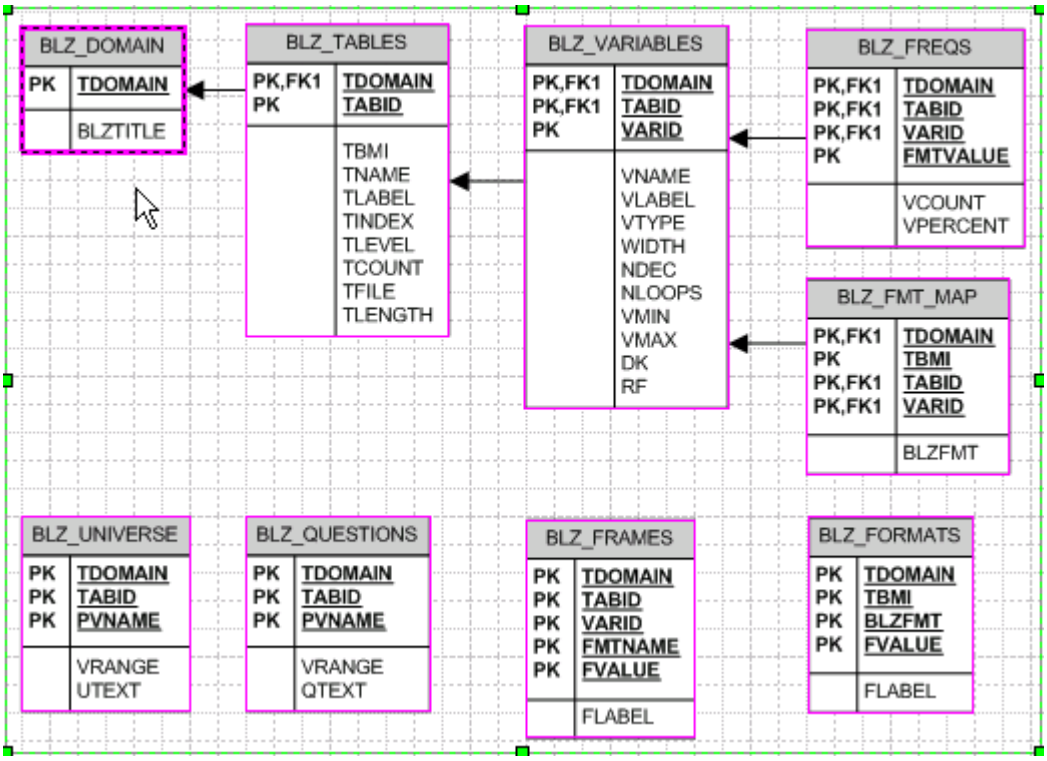
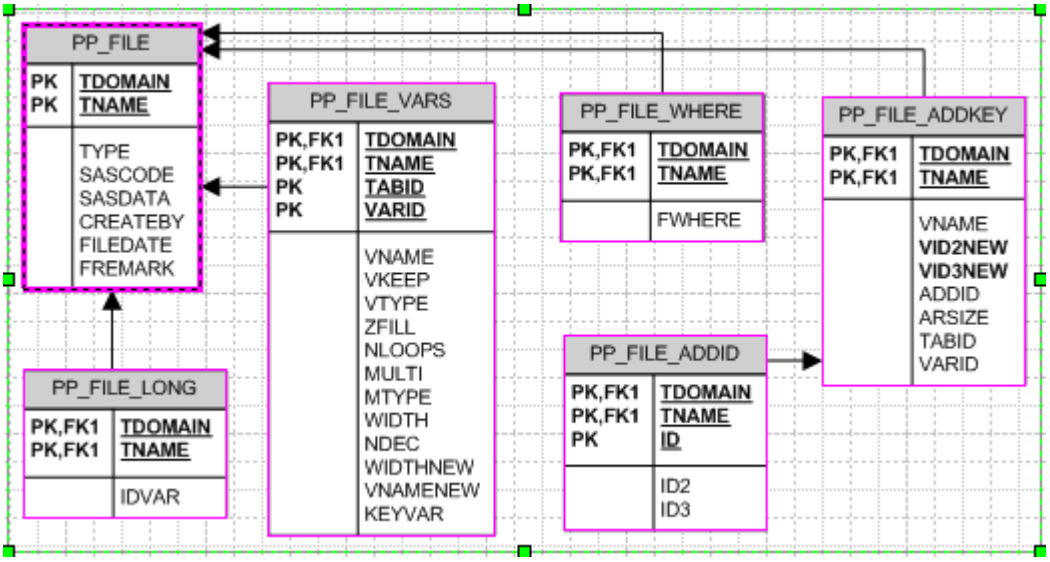


Figure 12.2: ER diagram of SAS setup tables



The SAS data, log and output are written to the folder specified in Section 3. DETL also has a function to check for errors in the log file and report them back to the calling interface.

11. Conclusion and Summary

The DETL tool developed by the PSID programming staff has enabled the processing staff to methodically see all the changes that were made from one wave to the next and update first pass setups accordingly. DETL informs the user about the meta information about each variable including the loop level, the question text, the code frame and even the raw frequencies. Using this information, the user can create first step files and do basic transformations and variable renaming without having to write out the long SAS syntax to complete these.

The SAS programs generated from DETL can be run while also saving a copy of the SAS program and the log file with one click. DETL is an important tool for the processing staff that has saved time and reduced errors and confusion by providing all the Blaise meta information to the user to make informed decisions about variables, where they should be placed, and at what level.