This file contains information and sample SAS programs to create a permanent SAS dataset for users who want to use SAS in processing the MEPS data
provided in this PUF release.
There are two ways to create a permanent SAS dataset, using either the SAS transport data file (H178IF1.SSP) or the ASCII data file (H178IF1.DAT) supplied in this PUF release. Section A provides a sample SAS program for the first alternative, which is to convert the SAS transport data file to a
regular SAS dataset using the SAS PROCedure: XCOPY. Section B provides a sample SAS program for the second alternative, which is to read data from the ASCII data file using a SAS DATA step with INFILE, INPUT, and LABEL
statements. Section C explains format-related SAS statements that a user may optionally use when working with the SAS dataset. Examples of SAS programs (DATA step or PROC) are provided in all three sections, primarily for the benefit of inexperienced users. Section D contains complete SAS statements
that must be used in the programs described in Sections B and C. INCLUDED BELOW ARE NOTES APPLICABLE TO USERS OF SAS VERSION 8 OR HIGHER.

permanent SAS dataset from the data files provided in this PUF release. A. A Sample SAS Program for Converting the SAS Transport File to a
The SAS PROCedure XCOPY will read a SAS transport file and convert the data to regular SAS format, storing the output in a permanent SAS dataset.
This permanent SAS dataset can then be used for all future processing and analyses. Below is a sample SAS program that can be used to convert the SAS transport
file to a permanent SAS dataset (in a Windows environment, with SAS V8 or higher).
LIBNAME PUFLIB 'C:\MEPS\SASDATA'; FILENAME IN1 'C:\MEPS\DOWNLOAD\H178IF1.SSP'; PROC XCOPY IN=IN1 OUT=PUFLIB IMPORT;
RUN; SAS transport files, SAS data files, and SAS program files each should be
stored in separate locations (directory names). Storing different types of SAS files in one location can cause errors with converting or retrieving data.
Below are SAS statements to print a list of variables and a few sample records from the permanent SAS dataset:
PROC CONTENTS DATA=PUFLIB.H178IF1; TITLE 'List of Variables in MEPS H178IF1 SAS Dataset'; RUN;
PROC PRINT DATA=PUFLIB.H178IF1 (OBS=20); TITLE 'First 20 Observations in MEPS H178IF1 SAS Dataset'; RUN;
The LIBNAME statement tells SAS the location (directory name) to store the permanent SAS dataset which is output by PROC XCOPY. The FILENAME statement tells SAS the location (complete directory and file name) of the input SAS
<pre>NOTES: 1) If you have an error reading a SAS data file you created, the problem</pre>
may be a result of where you are storing and/or how you are retrieving the data. First check the data library for multiple releases of SAS files (e.g., V8 or higher with file extensions of '.SAS7BDAT' and V6
with file extensions of '.SD2') stored in the same location. a) You can avoid errors when reading these files by including the SAS release within the LIBNAME statement - e.g.,
LIBNAME PUFLIB V8 'C:\MEPS\SASDATA'; or
b) Store SAS data files with different file extensions such as .SD2 and .SAS7BDAT, in separate folders (do not co-mingle V8 and V6 files in the same folder);
or c) When importing transport files, output the SAS dataset to a different library than the one which contains the downloaded
SAS transport file - e.g., LIBNAME PUFLIB 'C:\MEPS\SASDATA'; FILENAME IN1 'C:\MEPS\DOWNLOAD\Hxxx.SSP';
PROC XCOPY IN=IN1 OUT=PUBLIB IMPORT; RUN; 2) The names used in the LIBNAME and FILENAME statements shown
above (i.e., PUFLIB, IN1) are arbitrary; they are only temporary aliases.
3) The directory and file names used in the LIBNAME and FILENAME statements shown above are Windows syntax and may need to be modified for other operating systems such as UNIX, MAC/OS, VMS, or OS/2.
4) H178IF1 is the internal SAS dataset name (also the PC file name, without the extension) prior to the creation of the SAS transport data
file. After running PROC XCOPY, the output SAS dataset assumes the same dataset name (or file name). Hence, in the example above, a file named H178IF1.SAS7BDAT will be created under the C:\MEPS\SASDATA directory when PROC XCOPY runs successfully.
5) The SAS transport file H178IF1.SSP was created from a SAS V9 data file, using PROC COPY. This file has been tested for use with
SAS V8 or higher. This file may work with earlier versions of SAS, although it has not been tested with those versions. Users who are unable to use this SAS transport file should instead convert the ASCII
data file H178IF1.DAT to a SAS dataset as described in Section B. B. A Sample SAS Program for Converting the ASCII Data File to a Permanent
SAS Dataset The complete SAS statements (INPUT and LABEL) included in Section D are
intended to save time for those users wishing to create a permanent SAS dataset from the H178IF1.DAT ASCII data file. These statements must be used in combination with other SAS statements to create the appropriate SAS
program, as shown below. To use the statements provided in Section D to create a SAS program, you will need an ASCII text editor. If you are using an interactive form of SAS (Windows, UNIX, OS2, etc.), use the editor provided as part of the SAS software.
Following is a sample SAS program that will convert the ASCII data file to SAS format:
LIBNAME PUFLIB 'C:\MEPS\SASDATA'; FILENAME IN1 'C:\MEPS\DOWNLOAD\H178IF1.DAT';
DATA PUFLIB.H178IF1; INFILE IN1 LRECL=59; INPUT; * to user: insert the complete INPUT statement that is
provided in Section D; LABEL; * to user: insert the complete LABEL statement that is provided in Section D;
RUN; Here is an explanation of the SAS statements used in the program above.
LIBNAME statement: This tells SAS the location (directory name) of the permanent SAS dataset.
FILENAME statement: This tells SAS the location of the input ASCII data file. DATA statement: This signifies the beginning of a SAS DATA step and specifies the output SAS dataset, referencing the LIBNAME entry (PUFLIB) and assigning
an internal SAS dataset name (H178IF1). In the example, after the successful completion of the DATA step, a PC file named H178IF1.SAS7BDAT would have been created in the C:\MEPS\SASDATA directory.
INFILE statement: This tells SAS the location (directory and file name) of the input ASCII data file. Also provided is the logical record length (59
bytes), with the default of RECFM=V implied when this parameter is omitted. LRECL and RECFM are optional parameters in the INFILE statement. With regard to these options, please note the following: the ASCII data file H178IF1.DAT
contains a 2-byte carriage return/line feed at the end of each record. When
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM=V option is used, the LRECL option must be specified as the logical
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM=V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H178IF1.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H178IF1.DAT). Note that if the RECFM option is omitted, then the default
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM=V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H178IF1.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H178IF1.DAT). Note that if the RECFM option is omitted, then the default option of RECFM=V is automatically used, and LRECL should be specified as the logical record (59 for H178IF1.DAT). INPUT statement: This specifies the input record layout, giving names and the
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM=V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H178IF1.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H178IF1.DAT). Note that if the RECFM option is omitted, then the default option of RECFM=V is automatically used, and LRECL should be specified as the logical record (59 for H178IF1.DAT).
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM=V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H178IF1.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H178IF1.DAT). Note that if the RECFM option is omitted, then the default option of RECFM=V is automatically used, and LRECL should be specified as the logical record (59 for H178IF1.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H178IF1.DAT). Variable type (numeric or character) is also defined via the INPUT statement. LABEL statement: This associates descriptive names with the SAS variables.
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM=V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H178IF1.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H178IF1.DAT). Note that if the RECFM option is omitted, then the default option of RECFM=V is automatically used, and LRECL should be specified as the logical record (59 for H178IF1.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H178IF1.DAT). Variable type (numeric or character) is also defined via the INPUT statement.
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM=V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H178IF1.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H178IF1.DAT). Note that if the RECFM option is omitted, then the default option of RECFM=V is automatically used, and LRECL should be specified as the logical record (59 for H178IF1.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H178IF1.DAT). Variable type (numeric or character) is also defined via the INPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM=V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H178IF1.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H178IF1.DAT). Note that if the RECFM option is omitted, then the default option of RECFM=V is automatically used, and LRECL should be specified as the logical record (59 for H178IF1.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H178IF1.DAT). Variable type (numeric or character) is also defined via the INPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files.
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM=V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H178IF1.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H178IF1.DAT). Note that if the RECFM option is omitted, then the default option of RECFM=V is automatically used, and LRECL should be specified as the logical record (59 for H178IF1.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H178IF1.DAT). Variable type (numeric or character) is also defined via the INPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; PROC FORMAT LIBRARY=PUFLIB;
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM=V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H178IF1.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H178IF1.DAT). Note that if the RECFM option is omitted, then the default option of RECFM=V is automatically used, and LRECL should be specified as the logical record (59 for H178IF1.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H178IF1.DAT). Variable type (numeric or character) is also defined via the INPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE;
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM=V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H178IF1.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H178IF1.DAT). Note that if the RECFM option is omitted, then the default option of RECFM=V is automatically used, and LRECL should be specified as the logical record (59 for H178IF1.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H178IF1.DAT). Variable type (numeric or character) is also defined via the INPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D;
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM=V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H178IFI.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H178IFI.DAT). Note that if the RECFM option is omitted, then the default option of RECFM=V is automatically used, and LRECL should be specified as the logical record (59 for H178IF1.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H178IF1.DAT). Variable type (numeric or character) is also defined via the INPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; VALUE;
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM=V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H178IF1.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H178IF1.DAT). Note that if the RECFM option is omitted, then the default option of RECFM=V is automatically used, and LRECL should be specified as the logical record (59 for H178IF1.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H178IF1.DAT). Variable type (numeric or character) is also defined via the INPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; RUN; Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; OPTIONS FMTSEARCH=(PUFLIB);
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM-V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H1781F1.DAT). If RECFM-F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H1781F1.DAT). Note that if the RECFM option is omitted, then the default option of RECFM-V is automatically used, and LRECL should be specified as the logical record (59 for H1781F1.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H1781F1.DAT). Variable type (numeric or character) is also defined via the INPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; RUN; Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; OPTIONS FMTSEARCH=(PUFLIB); PROC FREQ DATA=PUFLIB.H1781F1; TABLES / LIST MISSING; FORMAT Varnam1 fmtnam1. Varnam2 fmtnam2;
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM=V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H178IF1.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H178IF1.DAT). Note that if the RECFM option is omitted, then the default option of RECFM=V is automatically used, and LRECL should be specified as the logical record (59 for H178IF1.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H178IF1.DAT). Variable type (numeric or character) is also defined via the INPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; RUN; Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; OPTIONS FMTSEARCH=(PUFLIB); PROC FREQ DATA=PUFLIB.H178IF1; TABLES / LIST MISSING;
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM-V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H178IF1.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H178IF1.DAT). Note that if the RECFM option is omitted, then the default option of RECFM=V is automatically used, and LRECL should be specified as the logical record (59 for H178IF1.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H178IF1.DAT). Variable type (numeric or character) is also defined via the INPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; RUN; Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; OPTIONS FMTSEARCH=(PUFLIB); PROC FREQ DATA=PUFLIB.H178IF1; TABLES / LIST MISSING; FORMAT varnaml fmtmaml. Varnam2 fmtnam2; * to user: substitute varnaml and fmtnaml with actual variable names and format names; * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statement; TITLE 'Frequency Distributions'; RUN;
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM-V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H178IF1.DAT). If RECFM-F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H178IF1.DAT). Note that if the RECFM option is omitted, then the default option of RECFM-V is automatically used, and LRECL should be specified as the logical record (59 for H178IF1.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H178IF1.DAT). Variable type (numeric or character) is also defined via the INPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; RUN; Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; OPTIONS FMTSEARCH=(PUFLIB); PROC FREQ DATA=PUFLIB.H178IF1; TABLES / LIST MISSING; FORMAT varnaml fmtnaml. Varnam2 fmtnam2; * to user: substitute varnam1 and fmtnam1 with actual variable names and format names; * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statement; TILE 'Frequency Distributions'; RUN; Here is an explanation of the SAS statements used above.
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECFM=V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for B1781F1.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for B1781F1.DAT). Note that if the RECFW option is omitted, then the default option of RECFM=V is automatically used, and LRECL should be specified as the logical record (59 for H1781F1.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H1781F1.DAT). Variable type (numeric or character) is also defined via the INPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; * to user: insert the complete set of VALUE statements found in Section D; FORMAT varnaml fmtnaml. Varnam2 fmtnam2; RUN; Below is an example of how to use the SAS formats defined by the PROC FORMAT FORMAT varnam1 fmtnaml. Varnam2 fmtnam2; * to user: substitute varnam1 and fmtnam1 with actual variable names and format names; * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statement; RUN; Here is an explanation of the SAS statements used above. LIBNAME statement: This tells SAS the location (directory name) of the SAS format library. Please
converting to a PC-SAS file, the LEECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECPH-V option is used, the LEECL option must be specified as the logical record length (e.g., 59 for H787EFL.DAT). If RECPM-F is used, then the LEECL value must be specified as the logical record length plus 2 (61 for H787EFL.DAT) and the LEECL value must be specified as the logical record of RECPM-V is automatically used, and LRECL should be specified as the logical record (59 for H787EFL.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H18FEL.DAT). Variable type (numeric or character) is also defined via the IMPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBNAME FUFLIB 'C:\MEPS\SASDATA'; PROC FORMAT LIBRARY=DUFLIB; VALUE; **COUNTY OF THE COUNTY OF THE SAS SASDATA'; PROC FORMAT LIBRARY=DUFLIB; PROC FORMAT LIBRARY=DUFLIB; PROC FORMAT LIBRARY=DUFLIB; PROC FREQ DATA=PUFLIB.H178IF1; TABLES / LIST MISSING; FORMAT varnand fintnaml. varnam2 fmtnam2; ** to user: substitute varnand and fmtnaml with actual variable names and format names; ** Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statement; ITIDE 'Frequency Distributions'; ** TITLE 'Frequency Distributions'; ** to user: substitute varnand and fmtnaml with actual variable names and format names; ** Insert the FORMAT statement provided in Section D, if you are using all the variables in the TA
converting to a PC-SAS file, the LERCL option should be used to specify the record length by PC-SAS. If the RECEMEV option is used, the LERCL option must be specified as the logical record length (e.g., 59 for H187EFI.DAT). If RECEMEP is used, then the LERCL value must be specified as the logical record length plus 2 (61 for H187EFI.DAT). Note that if the RECEMEN option is omitted, then the default option of RECEMEV is automatically used, and LRECL should be specified as the logical record (59 for H178FFI.DAT). INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H178FFI.DAT). Variable type (numeric or character) is also defined via the INPUT statement. LAREL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; PROC FORMAT LIBRARY=PUFLIB; VALUE; ***XALUE; ***RUN; Below is an example of how to use the SAS formats defined by the PROC FORMAT Procedure: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; OPTIONS FMYSSARCH=(PUFLIB); PROC FREQ DATA=PUFLIB.H178IF1; TABLES / LIST MISSING; FORMAT varnand Indimand. Varnand Indimand with actual variable names and format names; *** to user: substitute varnand and fmtnamd with actual variable names and format names; *** to user: substitute varnand and fmtnamd with actual variable names and format names; *** Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statements. ITTLE 'Frequency Distributions'; RUN; Here is an explanation of the SAS statements used above. LIENAME statement:
converting to a PC-SAS file, the LEECL option should be used to specify the record length by PC-SAS. If the RECKHEW option is used, the LEECL option must be specified as the logical record length (e.g., 59 for HT/SIFL)DATY. IF RECKHEF is used, then the LEECL value must be specified as the logical record length plus 2 (61 for HT/SIFL)DATY. Note that if the RECKM option is omitted, then the default option of RECKMEV is automatically used, and LRECL should be specified as the logical record (59 for HT/SIFL)DATY. INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (HI/SIFL)DATY. Variable type (numeric or character) is also defined via the INPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBRAME PUFLIB 'C:\MEFS\SASDATA'; PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; * to user: substitute varianl and fintnaml; * to user: substitute varianl and fintnaml with actual variable names and format names; * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statements; RIN; Here is an explanation of the SAS statements used above. LIBNAME statement: This tells SAS the location (directory name) of the SAS format library. Please note that SAS datasets (file name extension is 'SASTBACA' for SAS V6) and format libraries (file name extension is 'SASTBACA' for SAS V8 or higher and 'SC2
converting to a PC-SAS file, the LHRCL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECEMP option is used, the LHRCL option must be specified as the logical record length (e.g., 5) for H718FIP.DATY. If RECYMP- is used, then the LRCCL value must be specified as the logical record length plus 2 (61 for H718FIP.DAT). Note that if the RECEMP option is omatted, then the default option of RECYMP is automatically used, and LHRCL should be specified as the logical record (59 for H718FIP.DAT). IMPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H718FIP.DAT). Variable type (numeric or character) is also defined via the IMPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBNAME PUFLIB 'C:\MEPS\SASDATA'; PROC FORMAT LIBRARY=PUFLIB; VALUE; **XALUE; **XALUE; **XALUE; **PROC FORMAT LIBRARY=PUFLIB; PROC FORMAT PUFLIB 'C:\MEPS\SASDATA'; OPTIONS FRIESARGH=(PUFLIB); PROC FREQ DATA-PUFLIB.H178IF1; TABLES / LIST NISSING; FORMAT variables in the TABLES statement; TITLE 'Prequency Distributions; **XOPTIONS FRIESARGH=(PUFLIB) SAS the location (directory name) of the SAS format library. Please note that SAS datasets (file name extension is 'SASS'BDAT' for SAS Vo in higher and 'SDZ' for SAS Vo and format library. The specifics the SAS format library. PROC FORMAT statement: This identifies the SAS procedure that will make SAS formats according to VALUE statements. Formats will be stored in a file name (file name c
converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the RECTE-W option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H78TFI.DAT). If RECTEM-F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for H78TFI.DAT). Note that if the RECTEM option is omitted, then the default option of RECTEM-V is automatically used, and LRECL should be specified as the logical record (59 for H78TFI.DAT). INFUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H78TFI.DAT). variable type (numeric or character) is also defined via the INFUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. Sec Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBRAME PUFLIB 'C:\MEPS\SASDATA'; PROC FORMAT LIBRAMY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; * to user: substitute variable set of VALUE statements found in Section D; VALUE; * to user: substitute variable in the TABLES STATEMT; PROC FORMAT LIBRAMY=PUFLIB; * TORMAT variant fatnami. Variant fatnami with actual variable names and format names; * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES Statements: VERY STATEMENT OF THE VALUE STATEMENT OF SAS V6 on higher and 'SD2' for SAS V6 on be stored under the same directory. PROC
converting to a PC-SRS file, the LARCL option should be used to specify the record length to avoid use of a default record length by PC-SRS. If the RECEM-V option is used, the LARCL Option must be specified as the logical record length (e.g., 59 for HISHITLAND). If FRENTH-I is used, then the LARCL value must be specified as the logical record length plus 2 (61 for HISHITLAND). Note that if the RECEM Option is omitted, then the default option of RECEM-V is automatically used, and LARCL should be specified as the logical record (39 for HISHITLAND). LANDU statement: This apecifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (HISHITLAND). Variable type (numeric or character) is also defined via the ISPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data [iles. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBRAME PUPLIB 'C:\MMES\SASDATA'; PROC FORMAT LIBRARY=PUFLIB; VALUE; VALUE; RUN; Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure: LIBRAME FUFLIB 'C:\MMES\SASDATA'; OPTIONS FMYSTARCH=(FUFLIB); PROC FREQ DATA=FUFLIB.HITSIF1; TABLES / LIST MISSING; FORMAT varnand intomal. Varnam2 fmtnum2; to user: substitute varnam1 and fontman with actual variable names and format immers and the variables in the TABLES statement; TITLE 'Prequency Distributions; RUN; Here is an explanation of the SAS statements used above. LIBRAME Statement: This tells SAS the location (directory name) of the SAS format library. Please note that SAS datacents (file name extension is SAS/BACAT for SAS vo or higher and 'SC2' for SAS vo)
converting to a PC-SRS file, the LARCL option should be used to specify the record length to avoid use of a default record length by PC-SRS. If the RECEMP-V option is used, the LARCL option must be specified as the logical record length (0.9., 59 for HINSHILADY). If RECEMP-I is used, then the LARCL value must be specified as the logical record length plus 2 (61 for HINSHILADY). Note that if the MCDFW option is cnitted, then the default option of RECEMP-V is automatically used, and LARCL should be specified as the logical record (59 for HINSHILADY). INFUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCLI data file (HINSHILADY). Variable type (numeric or character) is also defined via the IMPUT statement. LABEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBRAME PUPLIB 'C:\MEPS\SASDATA'; PROC FORMAL LIBRARY=PUPLIB; VALUE 'C:\MEPS\SASDATA'; PROC FORMAL LIBRARY=PUPLIB; VALUE 'C:\MEPS\SASDATA'; PROC FORMAT STATEMENT (FUPLIB); PROC FREQ DATA=PUPLIB 'C:\MEPS\SASDATA'; OPTIONS PHYSEARCH=(PUPLIB); PROC FREQ DATA=PUPLIB 'C:\MEPS\SASDATA'; OPTIONS PHYSEARCH=(PUPLIB); PROC FORMAT Library variand finthman! varianal and fattomal with actual variable names and format names: **LIBRAMS PUPLIB 'C:\MEPS\SASDATA'; OPTIONS PHYSEARCH=(PUPLIB); PROC FORMAT STATEMENT (FUPLIB);
converting to a PC-SAS file, the LEECL option should be used to specify the record longth to avoid use of a default record longth by PC-SAS. If the RECEPT option is used, the LEECL option must be specified as the logical record length (e.g., 59 for HINSET). TAND. If RECEPT is also the LEECL value must be specified as the logical record length plus 2 (61 for HINSET). Mose that if the RECEPT option is omitted, then the default object of the logical record (59 for HINSET).DAT). THE RECEPT is also defined as the logical record (59 for HINSET).DAT) where the logical record (59 for HINSET).DAT) was also the logical record (59 for HINSET).DAT). Variable type (numeric or character) is also defined via the INSET states (which become SAS variables) in the ASCIT data file (HINSET).DAT). Variable type (numeric or character) is also defined via the INSET states. LIBEL statement: This associates descriptive names with the SAS variables. RUN statement: This tells SAS to execute all commands up to this point. See Saction A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Format—reluted SAS Statements If a user wents to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBRARE FULLE "C: WHEPS\SASADATA"; FROC FORMAT LIBRARY=PUPLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section 0; VALUE; * to user: substitute varianal and fintmand with actual variable names and format names; * Insert the FORMAT statement provided in Section 0, if you are using all the variables in the TAMIAS statement; TITLE Trequency Distributions; * TOWART Variables in the TAMIAS statement; TITLE Trequency Distributions; * TITLE Trequency Distributi
<pre>converting to a PC-88S file, the LEECL option should be used to specify the record length to cavoid used of a default record length by PC-88A. If the RECHEV Option is used, the LEECL option must be specified as the logical record length to-g., 55 or HETSTELDAY). If RECHEV is used, then the LEGGIVATION of the specified as the logical record length plus I (6 the LEGGIVA IS automated to specified as the logical record length plus I (6 the LEGGIVA IS automated as the logical record length plus I (6 the LEGGIVA IS automated as the logical record (59 for HITSTELDAY). INPUT statement: This specifies the input record layout, giving names and the logical record (59 for HITSTELDAY). INPUT statement: This specifies the input record layout, giving names and the logical record (59 for HITSTELDAY). INPUT statement: This specifies the input record layout, giving names and the logical record (59 for HITSTELDAY). INPUT statement: This specifies the input record layout, giving names and the logical record (50 for HITSTELDAY). INPUT statement: This specifies the input record layout, giving names and the logical record (50 for HITSTELDAY). INPUT statement: This specifies the input record layout, giving names and the logical record (50 for HITSTELDAY). INPUT statement: This specifies descriptive names with the SAS variables. INPUT statement: This specifies descriptive names with the SAS variables. INPUT statement: This specifies descriptive names with the SAS variables. C. Optional Format-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIBRAME PUTLIN 'C:\MEPS\SASDATA'; OPTIONS FATSHAMCHE (PUTLIN)' TABLES / LIBRAME PUTLIN 'C:\MEPS\SASDATA'; OPTIONS FATSHAMCHE (PUTLIN)' TABLES / LIBRAME SING; OPTIONS FATSHAMCHE (</pre>
converting to a FC-88S file, the LECU option should be used to specify the record length to avoid used of a default record length by PC-88A. If the RECTH-Voytion is used, the LENGT option must be specified as the logical record length (i.e., 5) for HISBYL-DAY). If RECTH-Pi is used, then the control conduction of RECTH-Voytion as for HISBYL-DAY. RECTH-VOYTION IS NOT HISBYL-DAY). RECTH-VOYTION IS NOT HISBYL-DAY. C. Optional Pormat-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Selow is a SAS program that will accomplish this: LIDNAME PUTLID 'CLAMPS/SAGDATA'; PROC FORMAT LIBRARY-MUTLIN, VALUE; **CONTINUED IN THE CLAMPS/SAGDATA'; POPTIONS FATERAGE—(PUFLID); PORC FRO DAYA-BUTLIN.HISBYL; TABLES / LICH MISSING; POPTIONS FATERAGE—(PUFLID); PORC FRO DAYA-BUTLIN.HISBYL; **A TO USER' SUBMITED WAS ADDITED. **A TO USER' SUBMITED HISBYL-DAYA TO ADDITED. **A TO USER' SUBMITED HISBYL-DAYA TO ADDITED. **A TO USER' SUBMITED HISBYL-DAYA TO ADDITED. **A TO USER' SUBMITED. **A TO USER' S
converting to a PC-88 file, the INNICL option should be used to specify the RECEMP Trecord length to avoid use of a default record length by PC-88. If the RECEMP Option is used, the INNICL option must be specified as the logical record length (e.g., 50 for HI/SEILANT). If RECEMP is used, then for HI/SEILANT). Note that if the RECEMP option is coticed, then the default option of RECEMP VI automatically used, and IRECL should be specified as the logical record (39 for HI/SEILANT). Note that if the RECEMP option is coticed, then the default option of RECEMP VI automatically used, and IRECL should be specified as the logical record (39 for HI/SEILANT). NUTUS statement: This seef is the HI/SEILANT). ANNUT statement: This associates descriptive names with the SAS variables. RAMEL statement: This tells SAS to execute all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional rormat-related SAS statements If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIDRAME POTLED 'C:\METRIC NAMEL POTLED 'C:\M
converting to a PC-SAS file, the LANGL option should be used to specify the PCCORD (1974 to avoid use of a defoult record length VPC-SAS.) If the RECTH-YoyLoo is used, the LANGL option must be specified as the logical record of the post of the RECTH-YoyLoo is used, the LANGL option must be specified as the logical record (1976 for RIFEITLANT). Mote that if the RECTH option is somitted, then the default option of ENCH-Yo Ha automatically used, and ERCL should be specified as the logical record (59 for RIFEITLANT). HEPT estaments finds specified the input record layou, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCH data file (RIFEITLANT). LANGL statement: This sussciates descriptive names with the SAS variables. RIM statement: This tells SAS to execuse all commands up to this point. See Section A.1 above for tips on retrieving and storing the permanent SAS data files. C. Optional Formal-related SAS Statements if a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this: LIMMANN HULLEY CLAMMEN(SASSANTA'); PROC FORMAT LIMMAN-PUPLID; VALUE; NULLE; HOW, RICHEST SASSANCH(VELID); PROC FORMAT LIMMAN-SASSANTA'; DOTIONS EXISSANCH(VELID); PROC PROD NOTA-SHIELEN HIMSPE; THAN S / LIST MISSING; DOTIONS EXISSANCH(VELID); PROC PROD NOTA-SHIELEN HIMSPE; THAN S / LIST MISSING; PROCE OF DOMA-SHIELEN HIMSPE; THE STRUMB SASSANCH (VELID); ROW, RICHEST SASSANCH (VELID); ROW, PROC PROM NOTA-SHIELEN HIMSPE; THE STRUMB SASSANCH (VELID); ROW, PROC PROM NOTA SHIELEN HIMSPE; THE STRUMB SASSANCH (VELID); ROW, ROW Is statement: This sells SAS the location (directory name) of the SAS LOTMAT HIMSPE. LIMMANE SHIELENCH THE SASSANCH SAS SHOWN IN THE SASSANCH SAS
converting to a PC-SAS file, the INNICL option should be used to specify the PCCOVAL CONCOVAL CONCOVAL STATES OF A STATES AND AND ASSESSED AND ASSESSED AND ASSESSED AND ASSESSED AND ASSESSED ASSESSED AND ASSESSED ASSESS
converting to a PC-SAS (lie, the EECC option should be used to specify the PECCAC Length to avoid use of a calcular corror length by PC-SAS. II the EECCP-TOPION is used, the LENGTH COLLON BY SET (1997) option is used, the LENGTH COLLON BY SET (1997) option is used to be specified as the logical record length plus 2 (cl for HIMEN-LAW). Note that if the SECT MORPH OPTION IS CONTROL (S) for HIMEN-LAW). Note that if the SECT MORPH OPTION IS CONTROL (S) for HIMEN-LAW). HIMEN STANDARD (S) FOR HIMEN-LAW), and ANNEL should be specified as the logical record clysus, giving names and the beginning and smaling column positions for data items (which become SAS variables) in the ACCT data Tile (HIMEN-LAW). HERE statement: This acclosed IT (HIMEN-LAW), variable type (numeric or Canacter) is also declined via the HIMEN statement. C. Optional Pormat-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format flower and files. C. Optional Pormat-related SAS Statements If a user wants to use formats for the SAS variables, a SAS format flower must first be created. Pelow is a SAS program that will accomplish this: LINHAND FORDING (S) METASASONTA'; DECOVER FORDING (S) METASASONTA'; DECOVER FORDING (S) METASASONTA'; OPTIONS PRESSASONE (PUPLIS); HERE TO USED (S) METASASONTA'; OPTIONS PRESSASONE (PUPLIS); HERE TO USED (S) METASASON (PUPLIS); HERE TO USED (S
converting to a EC-SEC Cite, the EEECT option should be used to specify the record length to avoid use of a default record length by PC-SEC. If the survey option is used, the LAMILL uprion must be specified as the logical record length of the SEC SEC. A SEC. If the SEC SEC. A SEC. If the SEC SEC. If t
converting to a TC-88 file, the LEECT ontion should be used to specify the record longth by the Anal. If the INCREDIO of the used, the LECT option must be specified as the logical record longth by the Anal. If the INCREDIO option is used, the LECT option must be specified as the logical record longth plus 2 (6) for NYREL-DATE. Note that if the SECR option is used plus 2 (6) for NYREL-DATE. Note that if the SECR option is described by used, and LARCH should be upscified as the logical record (39 too BYSTEL-DATE). HIBST statements This specifies the input record layout, giving names and the beginning and ending noise magnifies the input record layout, giving names and the beginning and ending noise magnifies the input record layout, giving names and the beginning and ending noise magnifies the input record layout, giving names and the beginning and ending noise magnifies. LARE statements This associates descriptive names with the SAS variables. AND statements This associates descriptive names with the SAS variables. C. options Format-volated NAS Statements It a user would to use formats for the SAS variables, a SAS format library must first be created. Scalar is a SAS program that will accomplish this: LINEARS PUPILS 'C:\UEEPSVARSONATA'; proc PUBLS 'C:\UEEPSVARSONATA'; proc PUBLS 'C:\UEEPSVARSONATA'; orrivors purposate that system the complete set of VALUE statements found in Section 10, if you are using all the variables in the TABLES statements used shown. LINEARS PUPILS 'C:\UEEPSVARSONATA'; orrivors purposate pupils 'C:\UEEPSVARSONATA'; orrivors pupils 'C:\UEEPSVARSO
converting to a PC-SS (ile, the LEECT online should be used to seedity the record longth by no avoid use of a default record longth by PC-MAN. If the record longth he avoid use of a default record longth by PC-MAN. If the record longth he seed to the control of the post
converting to a DC-SR file, the IMECT option should be used to specify the concol Length by C-SAR. If the record Length to C-SAR. If the record Length (col., 5) Los UNIVERSAL TO LANGE AND LANG
enverting on a PC-BASE file, the LaECT option should be used to specify the second length in various on a control record length in the second length in various of a control record length levil, 50 con 1179121.0007. RECOVER Length levil, 50 con 1179121.0007. If RECOVER is seed, then the the EMECOV value make the preceding of the Polymonian of Length Second Length levil, 50 con 1178121.0007. All the Machine of Length Indian Control of Control 1178121.0007. All the Machine of Length Indian Control (1970 of 1178121.0007). RECOVER Length
encovering of a PC-SNA file, the LBECK, option should be used to specify the momend inequal navaria was a control respond ings to a world and the positive process (length in.g., 10 for MINTEP). IT MERROR is used, there is a LBECK, value must be especified as the logical recover inequal to the positive of the MINTEP, 2017. Broke that I the MINTEP, 2017. It must be a positive of a the logical recover of the format in the mintep of the section of the minter of th
encovering to a PC-BAS file, the LBECK, option should be used to specify the record langer to covering an experimental record langer to covering an experimental record langer to covering a control record record langer to the specified as the region of record langer to the specified as the region of record langer to the specified as the region of record langer to the specified as the region of the record langer to the specified as the region of the record langer to the specified as the region of the record langer to the record la
converting to a pt task file, the MERCA cytion should be used to specify the record length (n.g., 10 to MINISTICAT). It MERCANCY is know, then the common length (n.g., 10 to MINISTICAT). It MERCANCY is know, then the common length (n.g., 10 to MINISTICAT). It MERCANCY is know, then the common length (n.g., 10 to MINISTICAT). It MERCANCY is know, the the common length (n.g., 10 to MINISTICAT). HAVE whatements this secretion the insur record length, giving names and the beginning and ending not ame positions for data there (which become 28% variations) is the Additional in (MINISTICAT). Workland upon (numeric or started) is laid edited with the INFO TENEDRAL (MINISTICAT). Workland (MINISTICAT) is a secondate development of the MERCANCH and I in (MINISTICAT). Workland (MINISTICAT) is a secondate development of the MERCANCH (MINISTICAT). Workland (MINISTICAT) is a secondate development of the MINISTICAT (MINISTICAT) is a secondate development of the MINISTICAT (MINISTICAT) is a secondate development of the MINISTICAT (MINISTICAT) is a secondate of the MINISTICAT (MINISTICAT (MINIS
converting to a fitting the insert cretion should be used to specify the record larged to continue of an effort in service larged by the record larged to continue of an effort of a fitting the record larged to continue of a fitting to the service of the International Continues o
converting to a greated file, the LABOLT option should be used to specify the record length to world use of a default record length of the Labolt record length to world use of a default record length file. If the record length for a seed a feet to logical record length place ? (eff for option of SECTHY is a statematically used, and LEGG should be specified as the logical record of SECTHY is a statematically used, and LEGG should be specified as the logical record of SECTHY is a statematically used, and LEGG should be specified as the logical record of SECTHY is a statematically used. HARDY recomments This specifies the input record Lapout, giving rames and the Despination of the labol of the SECTHY data file (PTSFT-RAT). Workshould type from the SECTHY data file (PTSFT-RAT). Workshould the specime of the SECTHY data file (PTSFT-RAT). Workshould the specime of the SECTHY data file (PTSFT-RAT). Workshould the specime of the SECTHY data file (PTSFT-RAT). Workshould the specime of the SECTHY data file the SE

'0' < - HIGH = 'VALID ID'

19 = '19 PANEL 19' 20 = '20 PANEL 20'

VALUE \$CONDIDX

VALUE \$DUPERS

VALUE EVENTYPE

1 = '1 MVIS'
2 = '2 OPAT'
3 = '3 EROM'
4 = '4 STAZ'
5 = '5 DVIS'
6 = '6 OMED'
7 = '7 HVIS'
8 = '8 PMED'

VALUE \$EVNTIDX

VALUE PANEL