make full use of the glossary

Aims Supply information sufficient in range, depth, and brevity for a newcomer to the CPP to quickly, accurately, and ‘easily’ analyze the data but understand and appreciate its complexity.

Document the CPP data as found in the public domain at NARA in 2002 and

Describe the rationale for, and document the technical aspects of, its manipulation into the system datasets supplied/will be supplied here to aid the end user in tailoring the data to their specific needs; be they analyst, statistician, or programmer, seasoned CPP analyst or newcomer.

Audience: research analysts statistical programmers electronic data managers

Beginning, intermediate, and senior levels of each: including Fellows in the Department of Pediatrics, Johns Hopkins School of Medicine.

Few have substantial expertise in **all** three areas, each are essential in the research process and deficiency in any of: research report writing, the quantitative analyses/reports on which they are based, and the capture and manipulation of the electronic data on which both are founded, will detract from the utility of these data.

Native English speakers or proficient non-native speakers. However, much use of text formatting is designed to guide the user to what is considered important or potentially problematic.

Assumptions Readers have a minimum expertize in **each** of analytic, statistical, and computer techniques to understand the trade-off’s inherent in, and the complexity of, producing a version of the CPP data for ready analysis on a Personal Computer.

Major problems in accessing/using CPP data for analytic purposes as found in 2001:

Institutional Nominally in the public domain at NARA available for a ‘small’ fee (>$500).

Electronic data Eighty column punch-card record format stored (EBCDIC) on reel-reel tape media (32) for use on an IBM mainframe computer. Undocumented versions in ASCII format (63) on CD-ROM available on an ad-hoc basis from NICHD..

Documentation 6,000 pages on 29x microfiche (up to 98 pages - 7 by 14 - on each microfiche), 75 microfiche total, compiled/written ten years after data collection ended by persons not involved in the design or conduct of the CPP.

Dispersed, ad-hoc hard-copies of original documentation in the possession of individuals in Government, private research corporations, and Universities.

Size and complexity Helped by deliberate redundancy/repetition in all new documentation.

Media Transfer: microfiche documentation. (75 microfice 29x)

Microfiche A copy of the 75 microfiche was transferred to an electronic format. First, an attempt was made to scan the 'fiche using OCR for capturing the text of the 6,000 pages in a manipulable and searchable form. However, microfiching is a photographic process and poor originals and/or poor photographing (focusing and developing) will lead to poor copies. The originals appear to have been much xeroxed copies of typewritten originals: far too blurry and inconsistently typed for OCR. The Adobe Acrobat Distiller (capture as image/'picture') was used, as the Adobe Acrobat PDF Writer (capture as text - manipulable and searchable) could not be used.

An **Adobe portable document format electronic file (PDF) for each of the 75 microfiche titled) 0001.PDF through 0073.PDF, 0073-A.PDF, 0073-B.PDF** (1.6Mb-2.1 Mb, 153Mb in total: they compress little) is supplied.

The title of each microfiche/corresponding original 'fiche/PDF is given in Appendix XX.

The PDF files [are/WILL BE] grouped according to substantive area of interest.

1. Details of the positions of the data items on the reel-reel tapes [WILL BE] deleted

2. A blurred introduction, repeated many times, has been retyped and will be supplied once.

3. Bookmarks [WILL BE] added to aid navigation within these large PDF files. II.A prenatal and medical history forms.pdf. is an example.

Media transfer: NARA reel-reel-tape data (32 tapes) to ASCII

nara media transfer.pdf

Data The **32 public use** IBM/EBCDIC data files were successfully written in ASCII format to 2 CD-ROM disks by NARA (<1Mb-0.5 Gb, 0.8 Gb total, *nara media transfer.pdf*).

There are two large datasets/files:	**MDF0378.ASC**	501Mb (101 Mb zipped)
	VARFILE.ASC	95 Mb (15 Mb zipped)

The **other 30 'work'** ASCII datasets *.ASC total 99Mb (0.1-24Mb, 15Mb zipped)

Filenames Each of the ASCII datasets corresponds to one of the reel-reel tapes, but the ASCII datasets were renamed by NARA staff .pdf in DOS 8.3 format.

File format Most 25/32 (The notable exception isVARFILE.ASC), are in 80-column punchcard format i.e. there may be more than one record/line for an individual/case in each dataset.

File content VARFILE.ASC is the most important dataset of 7 which was manipulated to **one record per case** originally. It contains a selection of over 1200 variables.

MDF0378.ASC contains all 6.1 million card records from March 1978, 6700 variables.
The other 30 files, the 'work' files, are subsets of the data pertaining to certain topics.

CPP electronic data: ASCII; SAS, SPSS, and STATA system data sets.

The CPP data will be supplied as:	1.) ASCII data from NARA	*.ASC
or in compressed format (Winzip 8.1)	2.) SAS system datasets,	*.SAS7BDAT
	3.) SPSS portabl datasets	*.POR
	4.) STATA system datasets	*.DTA
	5.) ASCII data from NICHD	*.

1. SAS\Windows 8.2e is the program used here to read the 32 *.ASCII files into a system format.

2. SPSS system datasets were written from SAS using DBMSENGINES 7.0 (DBSPSSX/PR), opened in SPSS 11.0, and then saved as SPSS portable files using SPSS 11.0.

3. SPSS has a distinct advantage over SAS in storing variable value labels within the system data set. In SAS, these values are kept with separate syntax files, applying them using PROC FORMAT. SPSS is limited to 8 characters in variable names

Why these formats? While transfer to ASCII files on CD-ROM makes the CPP accessible to many more researchers, they still must read it into a statistical or database software package to generate descriptive reports and inferential statistics for analytic purposes, the two most used in this research genre are SAS and SPSS. Database packages do not compute the inferential statistics necessary for advanced statistical modeling, and spreadsheets have storage and memory issues in these large (# of records ‘long’ - 6.1 million in MDF0378.ASC, and # of variables ‘wide’ - >6,700).

Creating these formats. SAS is the de facto ‘gold standard’, and its chief advantage is its data management capability without resort to higher order programming like C or Basic. SQL functionality is included in PROC SQL. The trade-off is the steep learning curve and relative complexity compared to SPSS’ data step. The statistics generated by both are, for most purposes, equivalent. SPSS switched its efforts to utilizing Windows GUI capabilities earlier, and does not retain the mainframe, command interpreter ‘feel’ still noticeable in SAS. STATA is a more recent arrival and may overtake both SAS and SPSS in popularity. It is still far less widely used, but it is included here because it is the basic package used for training at JHU Medical Institutions.

DBMS Engines provides the capability of writing other system data sets directly from SAS. SAS reads SPSS transport files (usually *.POR) using PROC CONVERT syntax and the menu \data\table dialog box. However, there are considerable technical difficulties in producing STATA datasets and they are not yes supplied: datasets are limited to 2047 variables and its methods of storing numbers as 8 byte floats is causing transcription problems within DBMSENGINES and SAS.

SPSS has had the capability of directly opening post SAS 6 data sets (*.sas7bdat) since Version 10.0.05 but this author has found that it does not always handle large (wide) datasets (>1,000 variables) well and cannot read SAS datasets created with SAS compression option on (COMPRESS=YES).

Since SAS and SPSS now read each others system files directly i.e. without the intermediary step of creating binary transport/export files, the system data formats may be supplied. While our aim is use on a PC, some researchers, particularly if operating in a mainframe environment other than UNIX (SAS Windows and SAS UNIX files are identical for most practical purposes and DBMS engines uses the same engine for both) may not care to, or have difficulty with, upload(ing) other system data sets.

SPSS users: punchcard records are ‘GROUP NESTED’ and may be read directly into SPSS (.pdf) from ASCII files. Modify rest of SAS syntax using Find and Replace, Textpad, or any regular expression capable editor. A sample program is given for forms ped-1 and ped-2 (SPSS read in example.pdf), but most users should be able to use SAS or SPSS datasets in Windows software, without resorting to direct input from ASCII.

STATA users STATA does not handle large datasets well: it reads the whole dataset into RAM so there must be more RAM than data set size. Also, there has been considerable technical difficulties in reading data directly from ASCII into Stata and considerable technical difficulties in conversion to Stata data sets usin SAS and DBMS/ENGINES dbstata ver=64 engine.

Therefore, it may be better for Stata users to select their variables from the documentation and create their own dataset with only the variables they want. Example programs and data dictionaries (varnumv.sta stataex.txt)) and technical tips are supplied. The infix statement is used for fixed column input.

Variable naming convention

The large (>6700) number of variables and SPSS' limitation of 8 character variable names makes it difficult to name variables in a manner which informs the reader of their content. However, incorporating information in the name can save considerable time in finding and using them.

Since these data were collected in 80-column punchcard record format the field number in the documented 'definition of codes' and punch card number has been used as the basis for the variable name, and the explanation of that name i.e. how to determine what information/variable it contains by referring to the 'Definition of codes'. The definition of codes is part of the microfiche/pdf documentation (section II, .pdf) and it also exists as a much xeroxed hardcopy impact typed in 1978 after creation of the electronic Master Data File.

Field is <u>not</u> the same as 'variable', more aptly line/paragraph in definition of codes.

Each variable name, limited to 8 characters,

1. Character 1: F for field   F#######

2. Character 2 and 3: field number from the definition of codes   F23#####

3. Characters 5 through 8: punchcard number   F23#5678

4. Character 4 is usually an underscore, acting as a visual break and placer.   F23_5678

Thus, f23_5678 would read 'field 23 of the definition of codes for card 5678', which is card 5 for data collection form 678, which is form 78 of series 6.

This may be looked up in the definition of codes for that form and its punchcards either on the microfiche/pdf or in the 1978 hardcopy.

In some cases, usually the first field in a punchcard F1, the field refers to several items of information in a previous punchcard for that form.
For example, F1 of card 2 for form 678 could refer to fields 1 through 5 of

card 1 for form 678. As we would put it, there are five variables in F1 for card 2 for 678.

This is typical for dates mmddyy which are one field in the documentation but we split into 3 (mm dd yy) variables; and age, sex, race collected on each card for a form.

In these instances the fourth character F##4####, usually an underscore,
is used and F1_2678 is broken down into f1a/b/c/d/e2678 or a date

F##a2678 F##b2678 F##c2678

Of course, any user may create their own variable name if they read data directly from ASCII format or rename variables, but the naming convention tells both where the variable's data was read from and directions to its explanation in the definition of codes without reference to cumbersome indexes in the documentation.

THIS NAMING CONVENTION HAS **NOT** BEEN USED FOR THE COMPENDIUM 'VARFILE' WHICH IS TREATED SEPARATELY (see varfile users guide.pdf). The varfile variables have been numbered according to the order of their definition/description in hardcopy documentation (1972) and not their order in the data set (column-order). NOTE THAT NEITHER THE VARFILE VARIABLE NAME OR THE 1972 DOCUMENTATION REFER EXPLICITLY TO THE FORM or PUNCHCARD. This definitive information is found in the Varfile 'fiche/pdf documentation. However, it is clear in most cases and (.pdf) helps.

Compact Disc case insert and jewel case labels:

This is the first attempt, and it will be much improved in the next version, including printing on a sharper laser printer. The aim is to use it as a ready reference and 'quick and dirty' first-time user's guide.