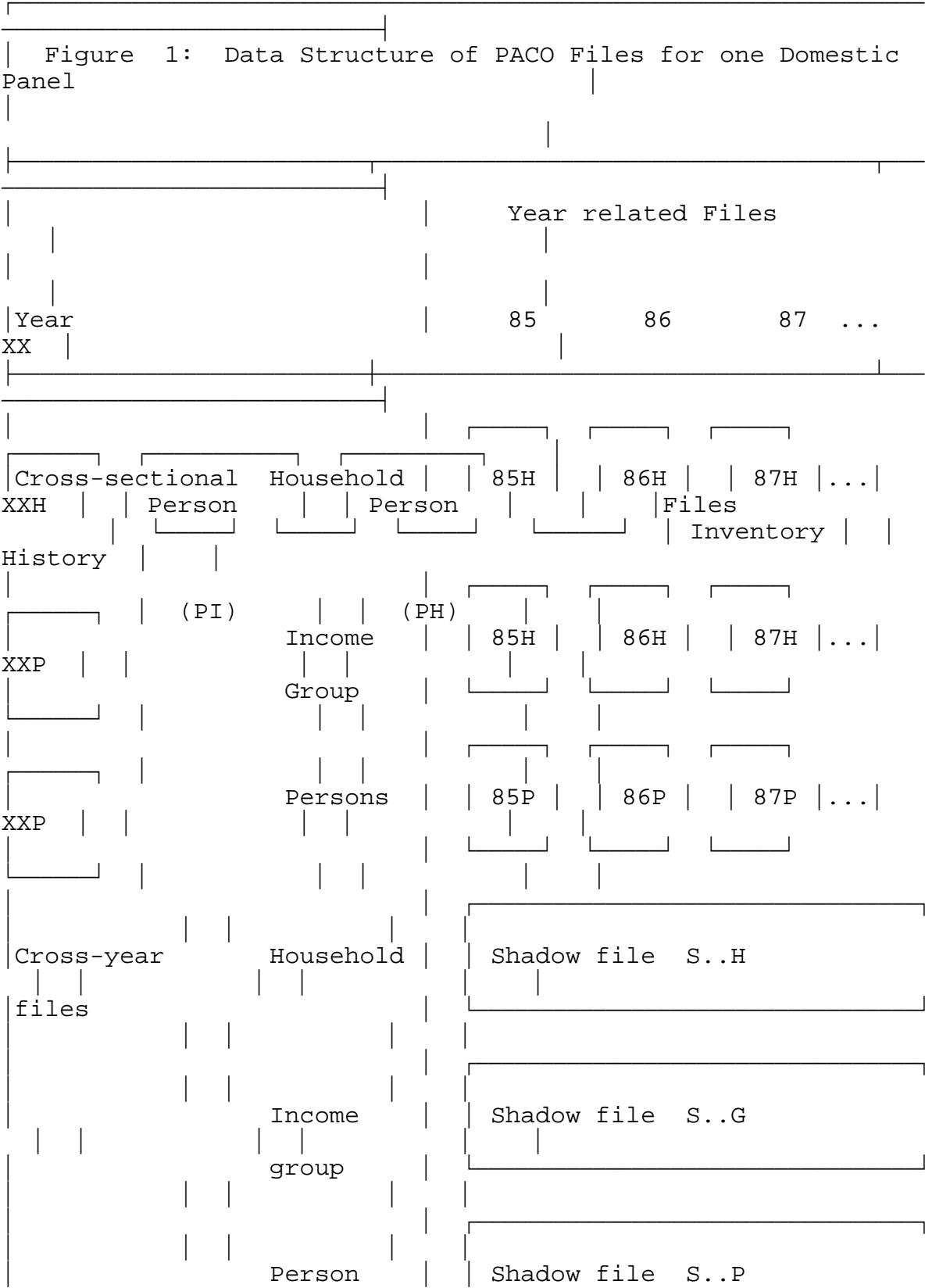
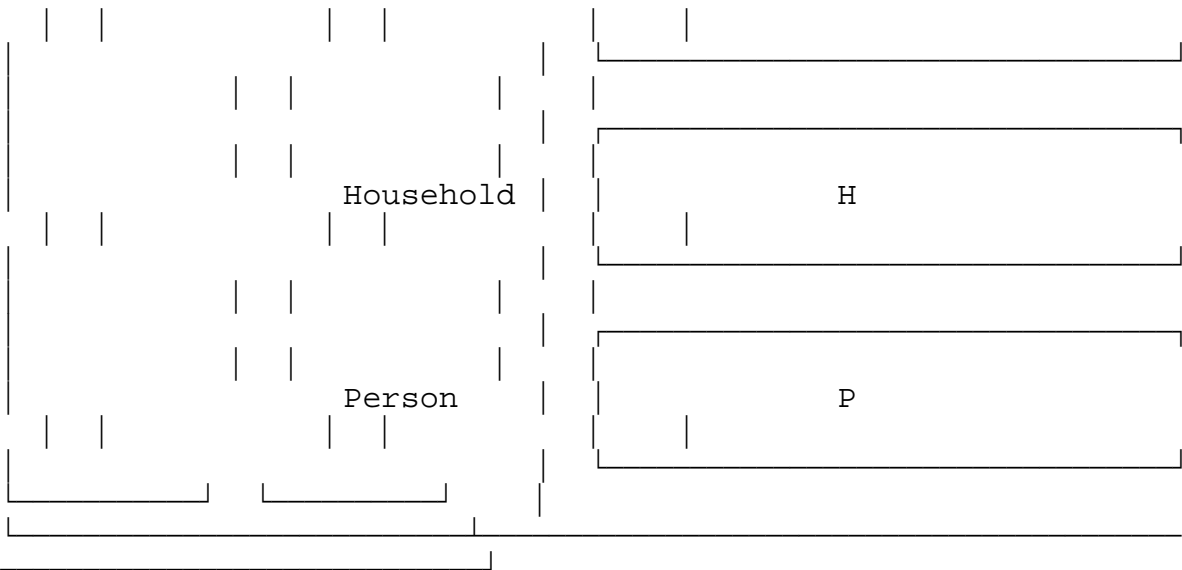


## Technical Specification of PACO Database

1. PACO Panel Database
  - 1.1. Structure of PACO Database Files
  - 1.2. Update Procedure for PACO Database files
  - 1.3. Convention for PACO file names
2. Data Structure of different PACO Files
  - 2.1. Cross-sectional Files
    - 2.1.1. Cross-sectional Files for Households
    - 2.1.2. Cross-sectional Files for Income Groups
    - 2.1.3. Cross-sectional Files for Persons
  - 2.2. Person Inventory Files
  - 2.3. Person History Files
  - 2.4. Cross-year Files for Households
  - 2.5. Cross-year Files for Persons
  - 2.6. Shadow Files
  - 2.7. Description of Shadow Files
    - 2.7.1. Shadow files for Household variables
    - 2.7.2. Shadow files for Income group variables
    - 2.7.3. Shadow files for Person variables
3. Production Requirements and Rules
  - 3.1. Production Requirements for PACO National Partners
  - 3.2. Production Requirements for PACO Data centre
  - 3.3. Procedure for Creation of PACO Result Files
4. Convention for PACO program names
5. Examples for PACO Programs
  - 5.1. Solution using yearly PACO Programs
    - 5.1.1. Description of the example
    - 5.1.2. Program listing of the example
  - 5.2. Solution using SPSS Macros
    - 5.2.1. Introduction into SPSS Macros
    - 5.2.2. Program listing of the example
6. Discussion of the proposed program structure

1. PACO Panel Database  
1.1. Structure of PACO Database Files





**Year related Files** contain data for one year and may contain data from different waves.

**Person Inventory File** contains "constant" variables for individuals.

**Person History File** contains data about individuals which (a) can not be related to one specific year of the PACO time period (85 -->) and (b) information about time periods prior to the start period of the panel studies.

**Cross-year Files** contain data for different years which could be accessed in a longitudinal perspective.

**Shadow files** contain not standardized variables from panel studies under original variable names.

Note: Yearly cross-sectional files on the individual level should contain records for all individuals (adults and children) in the household. The majority of variables for children records will be missing.

1.2 Update Procedure for PACO Database files

Procedure for adding new years into the Data Structure of PACO Files
<b>Year related Files:</b> Creating of additional yearly files

**Person Inventory File:** Adding records for new individ.

**Person History File:** Updating file, if changes has occurred and adding records for new individuals

**Cross-year Files:** Enlarging the files with additional variables

### 1.3. Convention for PACO file names

Year related files: fccxxuv

Non-year related files: fccuuvv

where **f** denotes the file type (see figure 2)

P = **P**aco Result file

S = Paco **S**hadow file

T = **T**emporary production file

I = **I**ntermediary production file

where **cc** denotes the country

GE = **G**ermany      LU = **L**uxembourg

FR = **F**rance      UK = **U**nited **K**ingdom

US = **U**sa

where **xx** denotes the year

85 = 19**85**    86 = 19**86**    ... **xx** = 19**xx**

where **u** or **uu** denotes the target files for PACO

H = **H**ousehold    G = income **G**roup

P = **P**erson

PI = **P**erson **I**nventory

PH = **P**erson **H**istory

where **v** or **vv** denotes the version if necessary

1,2,...3 or A,B,...Z

Examples for file names:

PLU85H: **P**aco resultfile for **L**uxemburg in year 19**85** on  
**H**ousehold level

PFR86P: **P**aco resultfile for **F**rance in year 1986 on Person  
level

SGEP : **S**hadow file **G**ermany on **P**erson level

## 2. Data Structure of different PACO Files

### 2.1. Cross-sectional Files

For each year of available data there will be two cross-sectional files, one for household variables, one for person variables. Lorraine and Luxembourg will have one additional cross-sectional file for income groups.

Each cross-sectional file contains data for one year and may contain data from different waves.

#### 2.1.1 Cross-sectional Files for Households

The file structure should be as follows:

##### a) Organizational Variables:

HL01 : Country  
 HL02 : Year  
 HxxL03 : ID-Household  
 HxxL04 : Pre-Year ID-HH  
 HxxL07 : ID-Reference Person  
 HxxL08 : ID-Spouse of Reference Person  
 HL12 : Case ID  
 HL13 : Random Group ID

##### b) Analysis Variables:

Hxx001-Hxx199 All Income variables (including aggregated variables from individual and income group level)

Hxx201-Hxx299 Demographic variables on household level

Hxx601-Hxx699 Housing variables

HxxWEIG Household Weight (cross-sectional)

##### c) Sort Key:

HxxL03 in ascending order

##### d) File name

P....H

### 2.1.2 Cross-sectional Files for Income Groups

These files can only be created for Lorraine(France) and Luxembourg.

#### a) Organizational Variables:

GL01 : Country  
 GL02 : Year  
 GxxL03 : ID-Household(HH)  
 GxxL04 : Pre-Year ID-HH  
 GxxL05 : ID-Group  
 GL12 : Case ID  
 GL13 : Random Group ID

#### b) Analysis Variables:

Gxx001- All available Income variables on level Income Group  
 Gxx199 (do not aggregate individual income variables into the income group level here)

#### c) Sort Key:

GxxL03 and GxxL05 in ascending order

#### d) File name

P....G

### 2.1.3 Cross-sectional Files for Persons

#### a) Organizational Variables:

PL01: Country  
 PL02: Year  
 PxxL03: ID-Household  
 PxxL04: Pre-Year ID-Household(HH)  
 PxxL05: ID-Group  
 PL06: ID-Individual  
 PxxL07: ID-Reference Person  
 PxxL08: ID-Spouse of Reference Person  
 PxxL09: ID-Spouse  
 PL10 : ID-Father  
 PL11 : ID-Mother  
 PL12 : Case ID

PL13 : Random Group ID

b) Analysis Variables:

Pxx001-Pxx199 All available Income variables on the individual level

P201 Sex

P202 Birth year

Pxx201-Pxx299 Demographic variables

Pxx301-Pxx399 Labour Force variables

Pxx801-Pxx999 Other variables

PxxWEIG Individual Weight (cross-sectional)

c) Sort Key:

PL06 in ascending order

d) File name

P....P

## 2.2. Person Inventory Files

This file contains one record for each person.

a) Organizational Variables:

PL01 : Country

PL06 : ID-Individual

PL10 : ID-Father

PL11 : ID-Mother

b) Analysis Variables:

P201 : Sex

P202 : Birth year

P203 : Birth month

P401-P499 Family Background variables

c) Sort Key:



PL06 in ascending order

d) File name

P..PI

### **2.3. Person History Files**

This file contains one record for each person.

a) Organizational Variables:

PL01 : Country

PL06 : ID-Individual

b) Analysis Variables:

P301-P399 : Work history

P401-P499 : Education variables

P501-P599 : Marriage and Fertility

... : Spell data

c) Sort Key:

PL06 in ascending order

d) File name

P..PH

note: Depending on the number and type of variables it may be necessary to create more than one Person History File in future.

### **2.4. Cross-year Files for Households**

This file will contain all original households and split-off households which occurred within the panel years. This file would allow access to the data in a longitudinal manner without prior matching of Cross-sectional household files. The main reason for creating such a file is only related to performance.

At that moment, it is not planned to create such a file.

provisional file name:

P..H

## 2.5. Cross-year Files for Persons

This file will contain all original panel members and all new panel members. This file would allow access to the data in a longitudinal manner without prior matching of Cross-sectional individual files.

At that moment it is not planned to create such a file.

provisional file name:

P..P

## 2.6. Shadow Files

The creation of the PACO variables needs in many cases recodes and aggregations on original variables of the national panel studies in order to define consistent PACO variables. The consequence of this procedure is that for some variables there is more detailed information in the original panel studies than in the PACO files. A further consequence is that it is not possible to recode crucial variables - if necessary - in a different manner.

To avoid such consequences, we are creating shadow files, which contain untransformed original variables from the national panel studies. A subset of original variables is sufficient for this purpose. We need only those original variables in the shadow files in cases where the information in the PACO variables is not identical with the original variables. We do not need those original variables in the shadow files which are only renamed or where the values (codes) have been changed for existing categories.

### a) Examples for **necessity for having original variables in shadow files:**

- Variable: \_xx044 (Received private Cash Inter-household Transfers) is created by aggregating three original income variables named 'a', 'b', 'c'. In order to have the possibility to reconstruct the income aggregate \_xx044, the original variables from the panel 'a', 'b', 'c' will be stored under their original names into the shadow file.

- Variable: Pxx205 (Relationship to Reference Person in HH) is recoded from one original variables named 'd' which has 50 different values into a PACO variable with only 12 different values. Variable 'd' should be put into the shadow file.

- Variable: Pxx206 (Foreign Nationals) is a yes/no variable. The original variable 'e' from which Pxx206 is derived, has

information about exact nationalities , should therefore be put into the shadow files, if data protection rules allow such a procedure.

b) Examples where it is **unnecessary** to hold original variables in shadow files:

- Variable: `_xx044` (Received private Cash Inter-household Transfers) contains only one income source. The documentation tells the user which original variable this is.

- Variable: `_xx009` (Self-Employment Income from enterprises) contains not only self-employment Income from enterprises but also income from independent professionals, which should be properly in variable `_xx010`. This fact can be found in the documentation.

- For variable `Pxx205` (Relationship to Reference Person in HH) the original categories have not been changed but only the codes (values) of the categories.

\_ Variable: `Hxx002` (Wages and Salaries) or some more other income variables on the household level are a summation of individual incomes. We do not need shadow variables for variables, which are created via aggregations of information from a lower level.

For some of the PACO variables it may be necessary to use a large number of original variables and to use a lot of transformation commands. For such variables it is unpracticable to hold all those original variables in shadow files. For such cases a compromise must be found between transparency and the number of original variables in the shadow file.

Summarizing we can say that it is not the aim to store the complete set of variables from original panel studies - which was used to create the standardized PACO variables - as shadow variables. Only a subset of those original variables which cannot be reconstructed from the PACO variables should be stored as shadow variable.

We do not want to create shadow files for each cross-sectional file but only one for all years together (Cross-year files).

## **2.7. Description of Shadow Files**

### **2.7.1. Shadow files for Household variables**

In this file each original household has one record; for each single split-off from the household there will be one additional record.

## a) Organizational Variables:

HL01 : Country  
 HL03 : ID-Household

## b) Identifiers from Original Panel Data

..... : ID-Household number (use original variable names)  
 ..... :

## c) Shadow Variables:

..... : Original variables (use original variables names, if  
           possible; the original variable names must contain the  
           reference year)  
 ..... : "  
 .....: "

## d) Sort Key:

HL03 in ascending order

## e) File name

S..H

**2.7.2. Shadow files for Income group variables**

## a) Organizational Variables:

GL01 : Country  
 GL03 : ID-Household(HH)  
 GL05 : ID-Group

## b) Identifiers from Original Panel data

..... : ID-Household number (use original variable names)  
 ..... :

## c) Shadow Variables:

..... : Original variables (use original variables names, if  
           possible; the original variable names must contain the  
           reference year)  
 ..... : "  
 .....: "

## d) Sort Key:

GL03 and GL05 in ascending order

e) File name

S..G

### 2.7.3. Shadow files for Person variables

a) Organizational Variables:

PL01 : Country  
PL06 : ID-Individual

b) Identifiers from Original Panel data

..... : ID-Individual number (use original variable names)  
..... :

c) Shadow Variables:

..... : Original variables (use original variables names, if  
possible; the original variable names must contain the  
reference year)  
..... : "  
.....: "

d) Sort Key:

PL06 in ascending order

d) File name

S..P

### **3. Production Requirements and Rules**

#### **3.1 Production Requirements for PACO National Partners**

- Each partner creates from the the national panel database national SPSS work files with the same structure and with same name for files (table, record type) as the national panel study uses. Ideally the SPSS systemfiles should contain all original variables but it must contain as an absolute minimum those variables which are necessary for PACO variables. These files must be sent to the PACO centre. They will be stored on the PACO-Archive.

- The structure of the SPSS programs should be as similar as possible to the proposed one.

- The SPSS program names and the names of the data files should correspond to the definition in this proposal.

- Each partner sends all his SPSS production programs to the PACO data centre. Each step and each program used for creating PACO variables and and PACO result files from the national SPSS work files must be documented and supplied.

- Each partner transmits the PACO result files (file names starting with letter 'P' and 'S' to PACO Data centre. Temporary and intermediary files starting with letter 'T' and 'I' need not to be sent to the centre. In the later stage of PACO when the whole process is tested, we need to exchange only the SPSS programs.

- Each partner sends the required documentation to the data centre.

#### **3.2 Production Requirements for PACO Data centre**

- The PACO-ARCHIVE stores the data from national panel data sets (SPSS work files) as SPSS systemfiles with the same structure and with the same name for files (tables, record types) as was used by the national panel study.

- PACO Datacenter always uses the same version of data and updates his PACO-ARCHIVE if the the national study changes its data.

- PACO Data centre modifies the FILE HANDLE commands and reruns the SPSS programs from its partners on his data from the PACO-Archive and produces the final version of the PACO files.

- The documentation from the partners will be integrated into the documentation system.



### 3.3 Procedure for Creating PACO Result Files

Figure 2 on page 15 explains the production procedures.

#### A) Produce National SPSS work files.

**Step 1:** Create national SPSS readable work files from the national Panel database and send data to PACO Data centre.

#### B) Production of variables for one topic, one unit and one specific year (see Schema page ...)

**Step 2:** Keep/Rename Variables (Programs with Modus R (Rename))

a) Keep the necessary original variables for one topic (e.g. demography), one unit (e.g. persons) and one specific year (e.g. 1985) from the national SPSS work files.

b) Rename the original variables -if necessary- to a new (constant) variable name and write them to a national temporary production file.

**Step 3:** Write Shadow Variables (Programs with Modus R (Rename))

Write shadow variables - if necessary - under their original variable names to the national intermediary Shadow file.

**Step 4:** Create PACO Variables (Program with Modus C (Create))

a) Use the previous in Step 2 produced national temporary production file with the constant variables names and create the PACO variables.

b) Assign the variable and value labels.

c) Write the PACO variables to the national intermediate production file.

#### C) Production of variables for another topic

Repeat Steps 2, (3) and 4 for other topics (e.g. income)

#### D) Production of one yearly national Paco result files for one unit (e.g. persons)

**Step 5:** Match all Intermediary production files for one year (Program with Modus M (match))

a) Match all previous created national intermediate productions files for one unit and for one year.

b) Sort the files.



c) Create the final PACO result file for ony year.

**E) Production of one specific yearly national Paco result file for other units (e.g. households)**

Repeat steps 2 to 5 for other units (e.g. households)

**F) Production of Paco result files for other years**

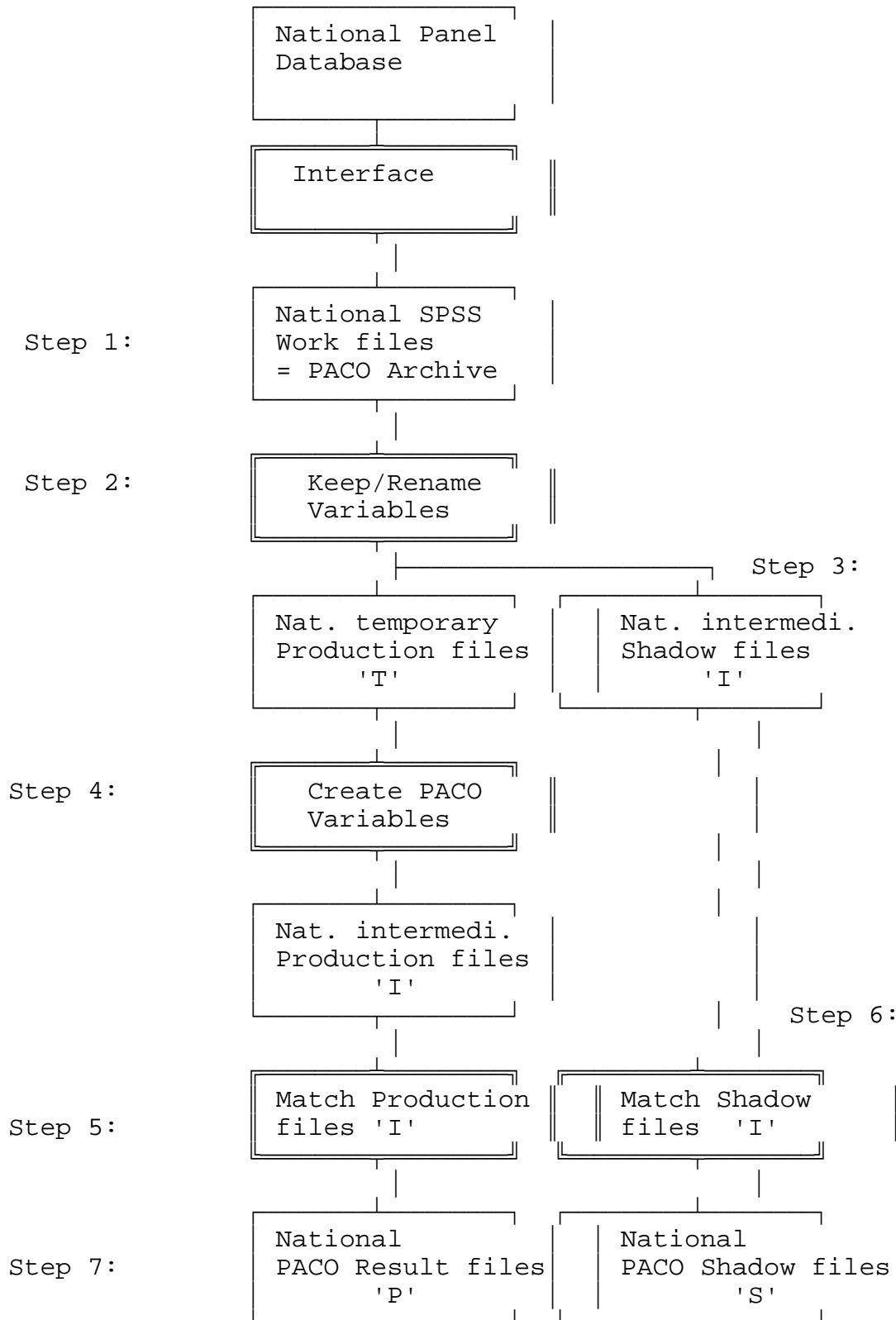
Repeat steps 2 to 5 for all other years (e..g.1986) fand for all units

**G) Match of Shadow files for all years**

**Step 6:** Finally Produce shadow files for households, income groups and persons (Programs with Modus M (match).

**H) Send data and programs to PACO Data centre (Step 7:)**

**Figure 2: Procedure for Creating PACO Result Files**





#### 4. Convention for PACO program names

Year related programs:           tmccxxuv  
 Non-year related programs:   tmccuuvv

where **t** denotes the variable theme or the type of result file

B = **B**iography  
 C = **C**onjoint (partner)  
 D = **D**emography  
 E = **E**ducation  
 H = **H**ousing  
 I = **I**dentifier  
 L = **L**abour force  
 O = **O**ther variables  
 P = **P**ACO result file  
 R = **R**evenue (income)  
 S = **S**hadow result file  
 T = **T**ime use  
 W = **W**eights

where **m** denotes the modus

A = **A**ggregate  
 C = **C**reate  
 E = **E**xecution  
 M = **M**atch  
 P = **P**repare  
 R = **R**ename

where **cc** denotes the country

GE = **G**ermany   LU = **L**uxembourg  
 FR = **F**rance    UK = **U**nited **K**ingdom  
                   US = **U**Sa

where **xx** denotes the year

85 = 19**85**   86 = 19**86**   ... xx = 19**xx**

where **u** or **uu** denotes the target files (units) for PACO

H = **H**ousehold   G = income **G**roup  
 P = **P**erson  
 PI = **P**erson **I**nventory

PH = **P**erson **H**istory

where **v** or **vv** denotes the version if necessary

1,2,...3 or A,B,...Z

Examples for Program names:

DRLU85P: **D**emographic variables with modus **R**ename for  
**L**uxemburg in year **1985** on **P**erson level

DCLU85P: ... with modus **C**reate

LCGE86H: **L**abour force variables with modus **C**reate for **G**ermany  
in year **1986** on **H**ousehold level

RCLU87G: **R**evenue (income) variables with modus **C**reate for  
**L**uxemburg in year **1987** on income **G**roup level

DCUKPI : **D**emographic variables with modus **C**reate for **U**K for  
**P**erson **I**nventory file

## 5. Examples for PACO Programs

### 5.1. Solution using yearly PACO programs

#### 5.1.1 Description of the example

This example is used in order to explain in detailed form the proposed PACO Program structure:

(1) Creating three cross-sectional PACO variables (Pxx002,Pxx204,Pxx302) for the individual level and producing two cross-sectional yearly PACO result files (PLU85P, PLU86P) for Luxembourg.

(2) Using three original variables ('a','b','c' from the first wave and ('d','e','f') from second wave. These variables are retrieved from two national SPSS work files (PERSON85, PERSON86).

(3) Storing two original variables 'b' and 'e' from national work files onto one longitudinal shadow file (SLUP).

**Summary for the program listings, which follows:**

See figure 3 for data structure and figure 4 for program structure.

a) Name of the used national SPSS work files: PERSON85  
PERSON86

b) Variables used from the SPSS work files:

Variable 'a' = Salary 1985 (PERSON85)  
Variable 'd' = Salary 1986 (PERSON86)

Variable 'b' = Family Status 1985 (PERSON85)  
Variable 'e' = Family Status 1986 (PERSON86)  
==> both variables must be stored onto shadow  
file SLUP.

Variable 'c' = Weekly Hours 1985 (PERSON85)  
Variable 'f' = Weekly Hours 1986 (PERSON86)

c) Name of the PACO Result files being created:

PLU85P : containing variables P85002 p85204 p85302  
PLU86P : containig variables P86002 p86204 p86302

d) Name of the PACO Shadow Result file being created:

SLUP : containing variable 'b' and 'e'

e) Using following temporary production files:

tlu85p1 tlu85p2 tlu85p3  
tlu86p1 tlu86p2 tlu86p3

f) Using following intermediary production files:

ilu85p1 ilu85p2 ilu85p3 ilu85p4  
ilu86p1 ilu86p2 ilu86p3 ilu86p4

g) Programs used to keep/rename original variables:

RRLU85P RRLU86P  
DRLU859 DRLU86P  
LRLU85P LRLU86P

h) Programs used to create PACO variables:

RCLU85P RCLU86P  
 DCLU85P DCLU86P  
 LCLU85P LCLU86P

i) Programs used to create final PACO result files:

Program PMLU85P produces result file PLU85P by matching  
 files: ILU85P1 ILU85P3 ILU85P4  
 Program PMLU86P produces result file PLU86P by matching  
 files: ILU86P1 ILU86P3 ILU86P4

j) Program used to create final PACO shadow file:

Program SMLUP produces shadow file SLUP by matching files:  
 files: ILU85P2 ILU86P2

k) EXEC PELU85P demonstrate the possibility of running several SPSS programs from one batch file.

Notes for PACO programs:

a) In this example the IBM/CMS version of the SPSS File Handle command is used. **The syntax of the File Handle commands differ from operation system to operating system.**

b) In this example only one variable is created in each program. **In real PACO programs not only one, but all variables for the same topic and for the same unit would be created by one program .**

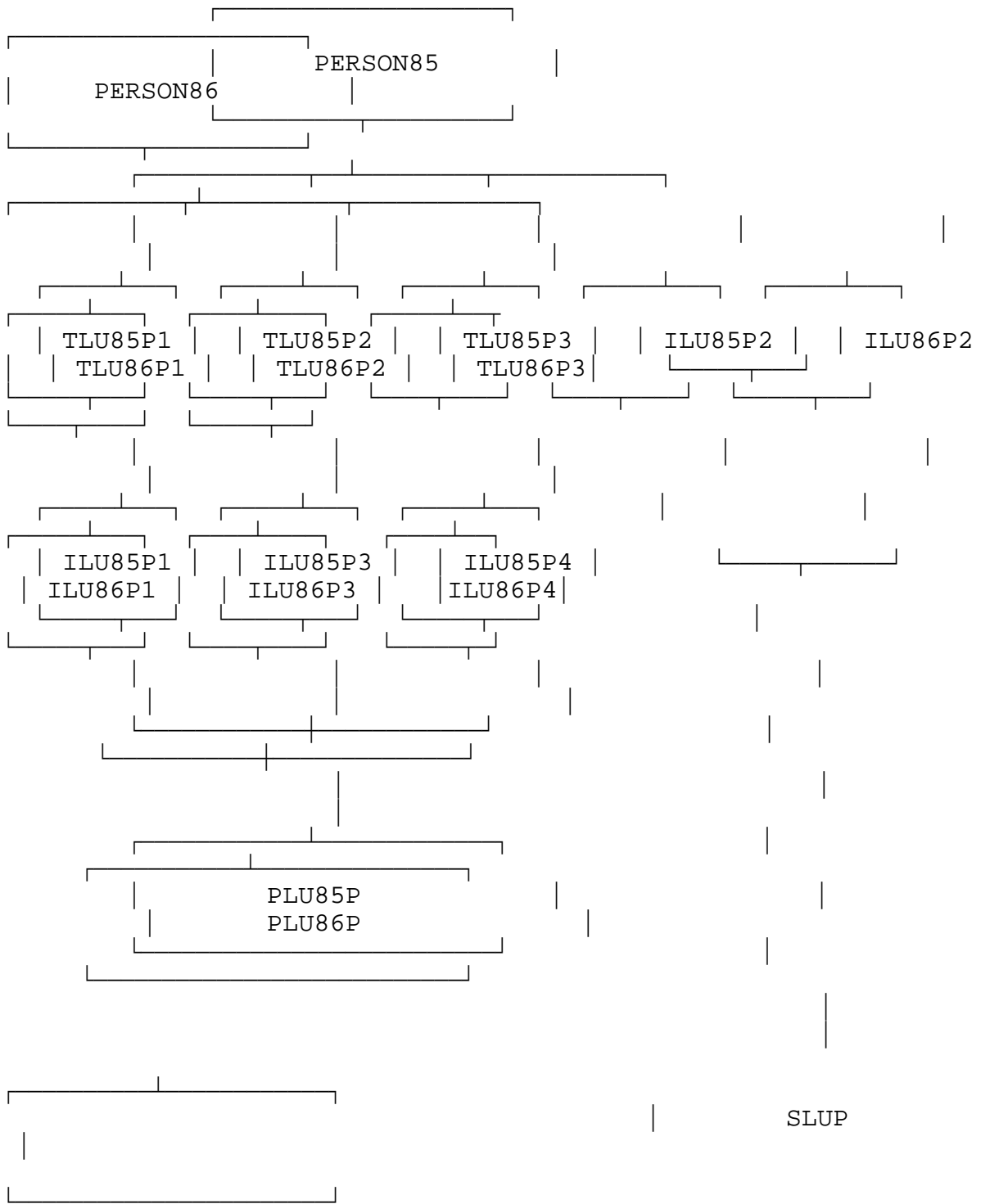
c) It is strongly recommended to use the FILE HANDLE command of SPSS to define files and **not** using the possibility of defining the file within the GET FILE command. The valid command in SPSS for Windows, e.g.:

```
GET FILE = 'c:\data\test.sys'
```

should be replaced by

```
FILE HANDLE TEST / NAME = 'c:\data\test.sys'
GET FILE=TEST
```

Figure 3: Data Structure used for the example



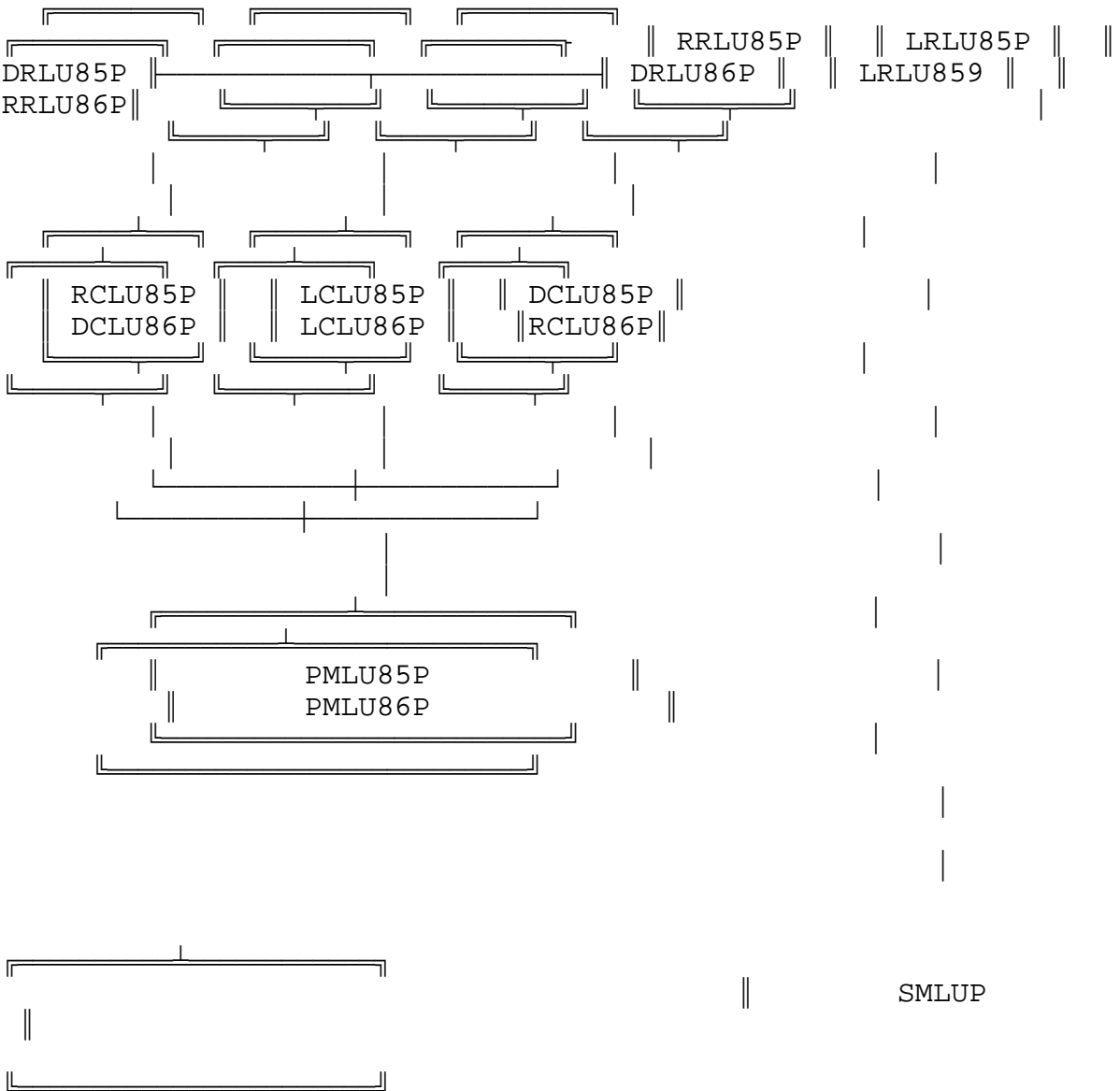
National SPSS work files: PERSON85 PERSON86

Temporary production files: TLU85P1 TLU85P2 TLU85P3

	TLU86P1	TLU86P2	TLU86P3
Intermediary production files:	ILU85P1	ILU85P2	ILU85P3 ILU85P4
	ILU86P1	ILU86P2	ILU86P3 ILU86P4
PACO Result files:	PLU85P	PLU86P	
PACO Shadow Result file:	SLUP		



Figure 4: Program Structure used for the example



Programs used to keep/rename original variables: RRLU85P

RRLU86P	
	DRLU859
DRLU86P	
	LRLU85P
LRLU86P	
Programs used to create PACO variables:	
RCLU86P	RCLU85P
	DCLU85P
DCLU86P	
	LCLU85P
LCLU86P	
Programs used to create final PACO result files:	PMLU85P
PMLU86P	
Program used to create final PACO shadow file:	SMLUP

### 5.1.2 Program listing of the example

#### a) Production for 1985:

Program RRLU85P reads the SPSS work file PERSON85 and renames the original variable 'a' (salary) to a new (constant) variable name 'salary' and writes to temporary file TLU85P1.

#### Program RRLU85P:

```
file handle person85 /name = 'person85 spssxfl *'
file handle tlu85p1 /name = 'tlu85p1 spssxfl T '
get file=person85
    /keep= a ...
save outfile=tlu85p1/keep= a ...
    /rename= (a=salary) (...=...)
```

Program RCLU85P reads file TLU85P1 and assigns variable 'salary' to the corresponding PACO variable Pxx002. In a next step this variable is renamed replacing xx by 85 and saved to intermediate file ILU85P1.

#### Program RCLU85P:

```
file handle tlu85p1 /name = 'tlu85p1 spssxfl * '
file handle ilu85p1 /name = 'ilu85p1 spssxfl T '
get file=tlu85p1
*
compute pxx002 = salary
*
* further assignments of income variables
*
variable lable pxx002 'Wages and Salaries 85'
...
save outfile=ilu85p1/keep= pxx002 ....
    /rename= (pxx002=p85002) (...=...)
```

Program DRLU85P reads the SPSS workfile PERSON85 and renames the original variable 'b'(family status) to a new (constant) variable name 'famstand' and writes to temporary file TLU85P2. In the last step the original variable 'b' is saved to the intermediate shadow file ILU85P2.

**Program DRLU85P:**

```
file handle person85 /name = 'person85 spssxfl *'
file handle tlu85p2 /name = 'tlu85p2 spssxfl T '
file handle ilu85p2 /name = 'ilu85p2 spssxfl T '
get file=person85
    /keep= b ...
save outfile=tlu85p2/keep= b ...
    /rename= (b=famstat) (...=...)
save outfile=ilu85p2/keep= b
```

Program DCLU85P reads file TLU85P2 and recodes variable 'famstand' into the corresponding PACO variable Pxx204. In a next step this variable is renamed replacing xx by 85 and saved to intermediate file ILU85P3.

**Program DCLU85P:**

```
file handle tlu85p2 /name = 'tlu85p2 spssxfl * '
file handle ilu85p3 /name = 'ilu85p3 spssxfl T '
get file=tlu85p2
*
recode famstand (1,2=1) (3,4=2) (5=3) (6=4) (7,8=5) into
    pxx204
*
* further recodes and transformations for other demographic
* variables
*
...
variable labels pxx204 'FAMILY STATUS 85'
value labels    pxx204 1 'married' 2 'separated' 3 'single'
    4 'divorced' 5 'widowed'
...
save outfile=ilu85p3/keep= pxx204 ....
    /rename= (pxx204=p85204) (...=...)
```

Program LRLU85P reads from the SPSS workfile PERSON85 and renames the original variable 'c' (Weekly Hours) to a new (constant) variable name 'workhour' and writes to temporary file TLU85P3.

**Program LRLU85P:**

```
file handle person85 /name = 'person85 spssxfil *'
file handle tlu85p3 /name = 'tlu85p3 spssxfil T '
get file=person85
    /keep= c ...
save outfile=tlu85p3/keep= c ...
    /rename= (c=workhour) (...=...)
```

Program LCLU85P reads file TLU85P3 and recodes variable 'workhour' into the corresponding PACO variable Pxx301. In a next step this variable is renamed replacing xx by 85 and saved to intermediate file ILU85P4.

**Program LCLU85P:**

```
file handle tlu85p3 /name = 'tlu85p3 spssxfil T '
file handle ilu85p4 /name = 'ilu85p4 spssxfil T '
get file=tlu85p3
*
if (workhour le 10) pxx302 = 3
if (workhour gt 10 and workhour le 20) pxx302 = 2
if (workhour gt 20 ) pxx302 = 1
*
* further recodes and assignments of labour force variables
*
variable lable pxx302 'EMPLOYMENT TYPE 85'
value lables pxx305 1 'full' 2 'part' 3 'some hours'
...
save outfile=ilu85p4/keep= pxx302 ....
    /rename= (pxx302=p85302) (pxx...=p85...)
```

Program PMLU85P matches all yearly related intermediate files from 1985 for persons ILU85P1, ILU85P3 and ILU85P4 (in our examples three) and creates the final PACO resultfile PLU85P.

**Program PMLU85P:**

```
file handle ilu85p1 /name = 'ilu85p1 spssxfil * '
file handle ilu85p3 /name = 'ilu85p3 spssxfil * '
file handle ilu85p4 /name = 'ilu85p4 spssxfil * '
file handle plu85p /name = 'plu85p spssxfil T '
match files file = ilu85p1
```

```

        /table = ilu85p3
        /table = ilu85p4
        /table = ...
sort file by ...
save outfile=plu85p

```

### **Production for 1986:**

Program RRLU86P reads the SPSS workfile PERSON86 and renames the original variable 'd' (salary) to a new (constant) variable name 'salary' and writes to temporary file TLU86P1.

#### **Program RRLU86P:**

```

file handle person86 /name = 'person86 spssxfil *'
file handle tlu86p1 /name = 'tlu86p1 spssxfil T '
get file=person86
    /keep= d ...
save outfile=tlu86p1/keep= d ...
    /rename= (d=salary) (...=...)

```

Program RCLU86P reads file TLU86P1 and assigns variable 'salary' to the corresponding PACO variable Pxx002. In a next step this variable is renamed replacing xx by 86 and saved to intermediate file ILU86P1.

#### **Program RCLU86P:**

```

file handle tlu86p1 /name = 'tlu86p1 spssxfil T '
file handle ilu86p1 /name = 'ilu86p1 spssxfil T '
get file=tlu86p1
*
compute pxx002 = salary
*
* further assignments of income variables
*
variable lable pxx002 'Wages and Salaries 86'
...
save outfile=ilu86p1/keep= pxx002 ....
    /rename= (pxx002=p86002) (...=...)

```

Program DRLU86P reads the SPSS workfile PERSON86 and renames the original variable 'e' (family status) to a new (constant) variable name 'famstand' and writes to temporary file TLU86P2. In the last step the original variable 'e' is saved to the intermediate shadow file ILU86P2.

**Program DRLU86P:**

```
file handle person86 /name = 'person86 spssxfil *'
file handle tlu86p2 /name = 'tlu86p2 spssxfil * '
file handle ilu86p2 /name = 'ilu86p2 spssxfil T '
get file=person86
    /keep= e ...
save outfile=tlu86p2/keep= e ...
    /rename= (e=famstat) (...=...)
save outfile=ilu86p2/keep= e
```

Program DCLU86P reads file TLU86P2 and recodes variable 'famstand' into the corresponding PACO variable 'Pxx204' in a next step this variable is renamed replacing xx by 86 and saved to intermediate file ILU86P3.

**Program DCLU86P:**

```
file handle tlu86p2 /name = 'tlu86p2 spssxfil * '
file handle ilu86p3 /name = 'ilu86p3 spssxfil T '
get file=tlu86p2
*
```

```

recode  famstand (1,2=1) (3,4=2) (5=3) (6=4) (7,8=5) into
          pxx204
*
*  further recodes and transformations for other demographic
*  variables
*
...
variable labels pxx204 'FAMILY STATUS 86'
value labels    pxx204 1 'married' 2 'separated' 3 'single'
                  4 'divorced' 5 'widowed'
*
...
save outfile=ilu86p3/keep= pxx204 ....
          /rename= (pxx204=p86204) (...=...)

```



Program LRLU86P reads from the SPSS workfile PERSON86 and renames the original variable 'f' (Weekly Hours) to a new (constant) variable name 'workhour' and writes to temporary file TLU86P3.

**Program LRLU86P:**

```
file handle person86 /name = 'person86 spssxfl *'
file handle tlu86p3 /name = 'tlu86p3 spssxfl T '
get file=person86
    /keep= f ...
save outfile=tlu86p3/keep= f ...
    /rename= (f=workhour) (...=...)
```

Program LCLU86P reads file TLU86P3 and recodes variable 'workhour' into the corresponding PACO variable Pxx302. In a next step this variable is renamed replacing xx by 86 and saved to intermediate file ILU86P4.

**Program LCLU86P:**

```
file handle tlu86p3 /name = 'tlu86p3 spssxfl T '
file handle ilu86p4 /name = 'ilu86p4 spssxfl T '
get file=tlu86p3
*
if (workhour le 10) pxx302 = 3
if (workhour gt 10 and workhour le 20) pxx302 = 2
if (workhour gt 20 ) pxx302 = 1
*
* further recodes and assignments of labour force variables
*
variable label pxx302 'EMPLOYMENT TYPE 86'
value labels pxx302 1 'full' 2 'part' 3 'some hours'
...
save outfile=ilu86p4/keep= pxx302 ....
    /rename= (pxx302=p86302) (pxx...=p86...)
```

Program PMLU86P matches all yearly related intermediate files from 1986 for persons ILU86P1, ILU86P3 and ILU86P4 (in our examples three) and creates the final PACO file PLU86P.

**Program PMLU86P:**

```
file handle ilu86p1 /name = 'ilu86p1 spssxfl T '
file handle ilu86p3 /name = 'ilu86p3 spssxfl T '
file handle ilu86p4 /name = 'ilu86p4 spssxfl T '
```

```

file handle plu86p   /name = 'plu86p  spssxfil T '
match files file    = ilu86p1
                    /table = ilu86p3
                    /table = ilu86p4
                    /table = ...
sort file by ...
save outfile=plu86p

```

#### c) Production of the shadow file

Program SMLUP matches all yearly related intermediate shadow files from all available years, here: ILU85P2 ILU86P2 and creates the final PACO shadow file SLUP.

#### **Program SMLUP:**

```

file handle ilu85p2  /name = 'ilu85p2  spssxfil T '
file handle ilu86p2  /name = 'ilu86p2  spssxfil T '
file handle slup      /name = 'slup      spssxfil T '
match files file     = ilu85p2
                    /file  = ilu86p2
                    /file  = ...
                    /file  = ...
sort file by ...
save outfile=slup

```

#### d) Executing a batch of SPSS program for production of 1985

**EXEC PELU85P:** (Batch file for IBM/CMS Operating System)

```

EXEC SPSSX RRLU85P
EXEC SPSSX RCLU85P
EXEC SPSSX DRLU85P
EXEC SPSSX DCLU85P
EXEC SPSSX LRLU85P
EXEC SPSSX LCLU85P

```

A similar effect can be realized using the INCLUDE command of SPSS.

## 5.2 Solution using SPSS Macros

### 5.2.1 Introduction into SPSS Macros

The creation of yearly PACO files can be realized by writing SPSS programs for one year. For each additional year very similar programs can be created by modifying the existing programs slightly. (see PACO Technical Documentation). Such a procedure creates a large number of SPSS programs, especially if the design is a modularized one. Furthermore the programs for each year will be different in following aspects:

- the name of the files used are different
- the original variable names for the same item are different between waves (in some panel studies).
- the Paco variables names differ between years, because the reference year is one component of the variable names
- the variable labels are different (also containing the year)

A good program design would require, however, programs which are more or less identical for each year, using parameter for items which change from year to year.

The optimal solution would be to have PACO programs which are global and can be re-used each year with an absolute minimum of necessary modifications. Such type of programs could be written by using formal parameters for year specific elements. Such a procedure can be realized using the SPSS MACRO facility.

The sample SPSS macros which are listed on later pages show the possibilities of writing global PACO programs. These global programs has following properties:

- a) The FILE HANDLE commands contain the year as parameter.
- b) Macro strings are used to hold item correspondence lists.
- c) One part of the macro variables handles the problems of changing variable names between years.
- d) Another part of the PACO variables defines PACO result variables ( containing implicit the reference year).
- e) After appropriate definition of the macro variables it is possible to use the macro variables in all further SPSS commands without any restrictions.
- f) Some of the created macro variables can be also used to document the files.
- g) When executing the macro programs the output listing of

SPSS shows the expanded commands in normal (readable) form.

h) The SPSS listing displays the original variable names in the case of changing variable names between waves.

**Advantages of using SPSS macros in PACO programs:**

- a) When writing such type of macro programs it is possible to run identical PACO programs for different years with one macro call!
- b) It is only necessary to write and to update a much smaller number of programs.
- c) The integration of a new wave of data into PACO is heavily supported.

**5.2.2 Program listing of the example**

A: Program 'RENAM.MAC' is the macro program sample of the PACO Keep/rename programs.

B: Program 'RENAM.SPS' calls the 'RENAM' macro.

C/D: Listings 'RENAM.83' and 'RENAM84' (shows the expanded version of the 'RENAM.MAC').

E: Program 'CREATE.MAC' is the macro sample program of the PACO Create programs.

F: Program 'CREATE.SPS' calls the 'CREATE' macro.

G: Description of the variables used in the macro programs.

A) \* '\*\*\* Program renam.mac \*\*\*'.

```

set messages=both.
set printback=both.
*----- define files -----
set mprint = on.
define !psidpro
    (start      = !token(1)
    /end        = !token(1)
    /cases      = !token(1)
    /prognam    = !enclose('(',',')')).
!let !site = 'ceps-windows'.
!do !xx = !start !to !end.
/* definition of files */
!let !tusxxpf  = !concat('tus',!xx,'pf').
!let !tusyypf  = !concat('c:\spsswin\test\tus',!xx,'pf.sys')
file handle !tusxxpf / name = !quote(!tusyypf).
!let !indivxx  = !concat('indiv',!xx).
!let !indivyy  = !concat('h:\psid\worksys\indiv',!xx,'.sys').
file handle !indivxx / name = !quote(!indivyy).
!let !hvwifxx  = !concat('hvwife',!xx).
!let !hvwifyy  = !concat('h:\psid\worksys\hvwife',!xx,'.sys').
file handle !hvwifxx / name = !quote(!hvwifyy).
/* define item correspondence lists */
!if (!xx = 83) !then
!let !hhnu = !concat ('v8802   ',
    'v10002   ','v11102   ','v12502   ','v13702   ').
!let !stellh = !concat('v30416   ',
    'v30447   ','v30482   ','v30518   ','v30556   ').
!let !famsta = !concat('v9276   ',
    'v10426   ','v11612   ','v13017   ','v14120   ').
!ifend
/* define pointer to strings */
!if (!xx = 83) !then
!let !anf = 1.
!ifend
!if (!xx = 84) !then
!let !anf = 9.
!ifend
!if (!xx = 85) !then
!let !anf = 17.
!ifend
!if (!xx = 86) !then
!let !anf = 25.
!ifend
!if (!xx = 87) !then
!let !anf = 33.
!ifend
/* retrieve constant variables */
!let !hhnum    = !substr(!hhnu,!anf,8).
!let !stellhv  = !substr(!stellh,!anf,8).
!let !famstan  = !substr(!famsta,!anf,8).

```

```

* ----- start of normal spss program -----
title 'KEEP OF DEMOGRAPHIC VARIABLES FOR US DATA'.
subtitle YEAR !xx USING PROGRAM: !prognam .
get file=!indivxx
    /keep=!hhnum persnum
    !stellhv.

n !cases.
compute !famstan = -2.
match files table=!hvwifxx
    / file = *
    by persnum
    /keep=
    persnum !hhnum !famstan !stellhv.
missing values !famstan (-2,0,9).
frequencies variables = !famstan,!stellhv.
drop document.
document this file contains demographic variables for
individual level.
document ' original file handle name:' !tusxxpf.
document '          original file name:' !tusyypf.
document '          original program:' !prognam.
document '          original site:' !site.
document '          type of file: temporary production
file'.
document '          country: us'.
document '          year:' !xx.
document '          target unit: P'.
document 'input: national work file :' !indivyy.
document 'input: national work file :' !hvwifyy.
file label prog: !prognam sort by persnum .
save outfile=!tusxxpf
    /map/compressed.
get file=!tusxxpf.
display documents.
display sorted labels.
title .
subtitle .
execute.
!doend
!enddefine.

```

B) \* '\*\*\* program renam.sps \*\*\*'.

```

set messages=both.
set printback=both.
include file = 'renam.mac'.
!psidpro start = 83 end = 84 cases = 100   prognam =
(drusxxpb).

```

## C) Listing Renam.83

```

FILE HANDLE tus83pf / NAME = 'c:\spsswin\test\tus83pf.sys'.
FILE HANDLE indiv83 / NAME = 'h:\psid\worksys\indiv83.sys'.
FILE HANDLE hvwife83 / NAME = 'h:\psid\worksys\hvwife83.sys'.
TITLE 'KEEP OF DEMOGRAPHIC VARIABLES FOR US DATA'.
SUBTITLE YEAR 83 USING PROGRAM: drusxxpb.
GET FILE= indiv83 /KEEP= v8802 PERSNUM v30416 .
N 100.
COMPUTE v9276 = -2.
MATCH FILES TABLE= hvwife83 / FILE = * BY PERSNUM /KEEP=
PERSNUM
    v8802 v9276 v30416 .
MISSING VALUES v9276 (-2,0,9).
FREQUENCIES VARIABLES = v9276 , v30416 .
DROP DOCUMENT.
DOCUMENT THIS FILE CONTAINS DEMOGRAPHIC VARIABLES FOR
INDIVIDUAL LEVEL.
DOCUMENT ' original file handle name:' tus83pf.
DOCUMENT '          original file name:'
c:\spsswin\test\tus83pf.sys.
DOCUMENT '          original program:' drusxxpb.
DOCUMENT '          original site:' ceps-windows.
DOCUMENT '          type of file: temporary production
file'.
DOCUMENT '          country: us'.
DOCUMENT '          year:' 83.
DOCUMENT '          target unit: P'.
DOCUMENT 'input: national work file :'
h:\psid\worksys\indiv83.sys.
DOCUMENT 'input: national work file :'
h:\psid\worksys\hvwife83.sys.
FILE LABEL PROG: drusxxpb SORT BY PERSNUM.
SAVE OUTFILE= tus83pf /MAP/COMPRESSED.
GET FILE= tus83pf.
DISPLAY DOCUMENTS.
DISPLAY SORTED LABELS.
TITLE.
SUBTITLE.
EXECUTE.

```

## D) Listing Renam.84

```

FILE HANDLE tus84pf / NAME = 'c:\spsswin\test\tus84pf.sys'.
FILE HANDLE indiv84 / NAME = 'h:\psid\worksys\indiv84.sys'.
FILE HANDLE hvwife84 / NAME = 'h:\psid\worksys\hvwife84.sys'.
TITLE 'KEEP OF DEMOGRAPHIC VARIABLES FOR US DATA'.
SUBTITLE YEAR 84 USING PROGRAM: drusxxpb.
GET FILE= indiv84 /KEEP= v10002 PERSNUM v30447 .
N 100.
COMPUTE v10426 = -2.
MATCH FILES TABLE= hvwife84 / FILE = * BY PERSNUM /KEEP=
PERSNUM
    v10002 v10426 v30447 .
MISSING VALUES v10426 (-2,0,9).
FREQUENCIES VARIABLES = v10426 , v30447 .
DROP DOCUMENT.
DOCUMENT THIS FILE CONTAINS DEMOGRAPHIC VARIABLES FOR
INDIVIDUAL LEVEL.
DOCUMENT ' original file handle name:' tus84pf.
DOCUMENT '          original file name:'
c:\spsswin\test\tus84pf.sys.
DOCUMENT '          original program:' drusxxpb.
DOCUMENT '          original site:' ceps-windows.
DOCUMENT '          type of file: temporary production
file'.
DOCUMENT '          country: us'.
DOCUMENT '          year:' 84.
DOCUMENT '          target unit: P'.
DOCUMENT 'input: national work file :'
h:\psid\worksys\indiv84.sys.
DOCUMENT 'input: national work file :'
h:\psid\worksys\hvwife84.sys.
FILE LABEL PROG: drusxxpb SORT BY PERSNUM.
SAVE OUTFILE= tus84pf /MAP/COMPRESSED.
GET FILE= tus84pf.
DISPLAY DOCUMENTS.
DISPLAY SORTED LABELS.
TITLE.
SUBTITLE.
EXECUTE.

```



```

E) * '*** create.mac ***'.

set messages=both.
set printback=both.
*----- define files -----.
set mprint = on.
define !psidpro
    (start      = !token(1)
    /end        = !token(1)
    /cases      = !token(1)
    /prognam    = !enclose('(',',')')).

    "
    " (identical with renam.mac)
    "

!let !iusxxpc = !concat('ius',!xx,'pc').
!let !iusyypc = !concat('c:\spsswin\test\ius',!xx,'pc.sys')
file handle !iusxxpc / name = !quote(!iusyypc).
    "
    " (identical with renam.mac)
    "

/* retrieve constant variables */
!let !hhnum = !substr(!hhnu,!anf,8).
!let !stellhv = !substr(!stellh,!anf,8).
!let !famstan = !substr(!famsta,!anf,8).
/* define paco variables */
!let !Pxx203 = !concat('P',!xx,'203').
!let !Pxx204 = !concat('P',!xx,'204').
!let !Pxx205 = !concat('P',!xx,'205').
* ----- start of normal spss program -----.
title 'CREATE OF DEMOGRAPHIC VARIABLES FOR US DATA'.
subtitle YEAR !xx USING PROGRAM: !prognam .
get file=!tusxxpf.
n !cases.
frequencies variables= !stellhv !famstan.
*-----.
numeric !Pxx203 (F2.0) !Pxx204 (F2.0) !Pxx205 (F2.0).
*----- FAMILY STATUS -----.
compute !Pxx203 = !famstan.
recode !Pxx203 (2=10) (1=20) (3=30) (4=40) (5=50) (else=-2).
if (missing(!famstan) and !stellhv eq 20) !Pxx203 = 20.
if (missing(!famstan) and !stellhv eq 90) !Pxx203 = 20.
missing values !Pxx203 (-2).
*----- RELATIONSHIP TO HEAD -----.
compute !Pxx204 = !stellhv.
recode !Pxx204 (10=10) (20,90=20) (22=31) (88=32) (30=41)
(33=42)
                (35=43) (83=44) (38=45) (37=50) (60=61)
(65=62)
                (50=70) (57=81) (58=82) (40=91) (47=92)
(48=93)

```

```

(66=101) (67=102) (68=103) (69=104) (72=105)
(73=106)
(70=107) (71=108) (74=109) (75=110) (95=111)
(96=112)
(97=113) (98=120) (else=999).
*----- NATIONALITY -----
compute !Pxx205 = -2.
missing values !Pxx205 (-2).
*.
*.
*.
variable labels !Pxx203 FAMILY STATUS !xx .
value labels !Pxx203
    10 'SINGLE'
    20 'MARRIED'
    30 'WIDOWED'
    40 'DIVORCED'
    50 'SEPARATED'
    51 'SEPARATED BY FACTS'
    52 'SEPARATED BY LAW'
    -2 'INAPPLICABLE'
'.
variable labels !Pxx204 RELATIONSHIP TO HEAD !xx .
value labels !Pxx204
    10 'HEAD'
    '
    20 'LEGAL SPOUSE OF HEAD'
    '
    "
    "
    "
    120 'OTHER NONRELATIVES'
'.
variable labels !Pxx205 NATIONALITY !xx .
value labels !Pxx205
    -2 'INAPPLICABLE'
'.
*.
*.
*.
frequencies variables= !Pxx203 !Pxx204.
*.
drop document.
document this file contains demographic variables for
individual level.
document ' original file handle name:' !iusyypc.
document ' original file name:' !iusxxpc.
document ' original program:' !prognam.
document ' original site:' !site.
document ' type of file: intermed. production
file'.
document ' country: us'.
document ' year:' !xx.
document ' target unit: P'.

```

```

document 'input: temporary prod.file:' !tusyypf.
file label prog: !prognam sort by persnum .
save outfile=!iusxxpc
    /keep=!hhnum persnum
    !Pxx203 !Pxx204 !Pxx205
    /map/compressed.
get file=!iusxxpc.
display documents.
display sorted labels.
title .
subtitle .
execute.
!doend.
!enddefine.

```

F) \* '\*\*\* Program create.sps \*\*\*'.

```

set messages=both.
set printback=both.
include file = 'create.mac'.
!psidpro start = 83 end = 84 cases = 100   prognam =
(drusxxpb).

```

G) Explanation of macro arguments:

- a) start: start year e.g. 85
- b) end: end year e.g. 86
- c) cases: no. of cases used for the SPSS 'N' command  
e.g 100 = 100 cases, 999999 = all cases
- d) program: character string for documentation reason (must  
enclosed in brackets).

note: In the call to the macro the arguments are not allowed  
to be prefixed by "!". In the macro the arguments must  
be used with the prefix "!".

2) Explanation of created macro variables

a) global macro variables

!xx: current year

!tusxxpf: "file handle name" of one specific temporary Paco  
file

!tusyypf: "file handle specification " of one specific  
temporary Paco

```

file
!iusxxpc:  "file handle name" of one specific intermediary
           PACO file
!iusyyyc:  "file handle specification" of one specific
           intermediary PACO file

```

b) macro variables to build the SPSS "FILE HANDLE" command for specific SPSS work files (used in the rename programs only)

variables used in the example:

```

!indivxx  : "file handle name" of one specific SPSS work file
!indivyy  : "file handle specification" of one specific SPSS
           work file

!hvwifexx : "file handle name" of one specific SPSS work file
!hvwifeyy : "file handle specification" of one specific SPSS
           work file

```

c) macro variables to declare specific constant variable names (used in rename and create programs)

variables used in the example:

```

!hhnum    : hh identifier
!stellhv  : relation to head (Stellung zum HV)
!famstan  : family status (Familienstand)

!hhnu     : string: item correspondence list for !hhnum
!stellh   : string: item correspondence list for !stellh
!famsta   : string: item correspondence list for !famsta

```

d) macro variables to declare specific PACO variables (used in create program)

variables used in the example

```

!pxx203   : family status
!pxx204   : relationship to head
!pxx205   : nationality

```

note:

- macro variables names a) and d) could be universal for all national PACO production programs; they are standardized (see Technical Description).
- macro variables b) are specific to the names for each national SPSS work files; they are not standardized. The

file names correspond to the name of the national work files.

- macro variables c) are specific for national temporary files; each programmer could create his own macro variables names; they are not standardized;

## 6. Discussion of the proposed program structure

Advantages of this program structure for integrating additional yearly data:

a) The PACO transformation commands remain completely unchanged in the programs which create PACO variables (here: RC....P,DC....P,LC....P)

b) In the programs which create variables (here: RC....P,DC....P,LC....P) and which match the files (here: PM....P) it is only necessary to modify the reference year by a global change with the editor.

c) One has to manually modify only the programs which renames the variables from the SPSS workfile (here: RR....P, DR....P,LR....P) by renaming the original variable name with the constant variable name.

d) The strong modularisation approach of the proposed program structure allows also - by using batch files or include files - to run many small SPSS programs in one 'GO'.

e) In the above used (simple) example one could argue that the programs which keep/rename variables (programs with modus 'R') could be integrated into the programs which create variables (programs with modus 'C'). But there are two reasons not to do so. (1) Performance is much better, when using small temporary production files instead accessing every time one big data file. (2) In some cases it is necessary not only to rename variables but also match different data files, sort files and to do other transformation on variables before the PACO variables can be created.

f) The modularisation approach together with predefined names for programs and files guarantees:

- that the PACO programs from different panel studies will have similar structure
- better communication between partners and PACO data centre
- a minimum of modification for integrating new waves
- long life of written SPSS programs

what is new:

## 2.6 One Paragraph