



Introduction to the FRS

A few words of Introduction	2
The FRS Database	2
<i>Weeklyisation</i>	2
<i>'LATEST' directories</i>	2
<i>STANDARD directories</i>	3
The Hierarchical dataset	3
<i>Background</i>	3
<i>Key variables</i>	4
<i>Normalisation principle</i>	4
<i>Table Descriptions</i>	5
The Flatfile	8

A few words of Introduction

The Family Resources Survey collects information on the incomes and circumstances of private households in the United Kingdom (or Great Britain before 2002-03).

The survey is sponsored by the Department for Work and Pensions. Whilst it was designed with our needs specifically in mind, it contains information that will be of interest to researchers and analysts from a wide range of disciplines in both the public and private sectors.

The survey includes household characteristics; income and receipt of Social Security benefits; tenure and housing costs; assets and savings; carers and those needing care; and employment.

This 'Introduction to the FRS' is aimed at providing a background to how the data is collected and then information on using the dataset that is derived from the survey. Further information on the survey can be found in our annual report.

The FRS Database

There are two versions of the FRS dataset; a fully edited version called Release 1 and an unedited version called Release 0. These are both available in either Hierarchical or Flatfile format and are stored on both London Unix boxes.

When data for a FRS year is updated, or amended, a new release is created for users. The old release is not deleted, so can still be accessed if required.

To avoid problems in accessing the data users are given two choices, to either use the Latest data available or select the required dataset release.

Weeklyisation

Virtually all amounts on the FRS are weeklyised – ie the figure given is pounds per week. The original period codes are provided so you can get back to the original answer if you wish. For example payamt=340.02279452, paypd=5 suggests the respondent gave a figure of 1477:50 per month. The exceptions are lump sums, which have a period code of 95, and certain figures where we ask for all payments over a certain time period (for example all bonus payments in the last 12 months) which do not have associated period codes.

'LATEST' directories

The virtual 'Latest' directories listed below provide links to the latest official edited releases for each survey year. They are automatically re-linked to any new data as soon as it is released. Using these directories in your libname statements, you can guarantee always to be using the latest official release, without having to update your libname statements.

Londu03 (ASD14)	Londu02
<code>/data1/latest/frs0304</code> <code>/data1/latest/frs0203</code> <code>/data1/latest/frs0102</code> <code>/data1/latest/frs0001</code> <code>/data1/latest/frs9900</code> <code>/data1/latest/frs9899</code> <code>/data1/latest/frs9798</code> <code>/data1/latest/frs9697</code> <code>/data1/latest/frs9596</code> <code>/data1/latest/frs9495</code>	<code>/repdata/sas_surveys/frs/latest/frs0304</code> <code>/repdata/sas_surveys/frs/latest/frs0203</code> <code>/repdata/sas_surveys/frs/latest/frs0102</code> <code>/repdata/sas_surveys/frs/latest/frs0001</code> <code>/repdata/sas_surveys/frs/latest/frs9900</code> <code>/repdata/sas_surveys/frs/latest/frs9899</code> <code>/repdata/sas_surveys/frs/latest/frs9798</code> <code>/repdata/sas_surveys/frs/latest/frs9697</code> <code>/repdata/sas_surveys/frs/latest/frs9596</code> <code>/repdata/sas_surveys/frs/latest/frs9495</code>

STANDARD directories

The actual directories listed below are the permanent directory names for the current releases. Using these directories in your libname statements you can guarantee always to be using the same data regardless of any new releases that may be provided in the future.

	Release 1 (edited)	Release 0 (unedited)
Londu03 (ASD14)	/data1/frs0304/frs0304d /data1/frs0203/frs0203e /data1/frs0102/frs0102a /data1/frs0001/frs0001d /data1/frs9900/frs9900e /data1/frs9899/frs9899f /data1/frs9798/frs9798f /data1/frs9697/frs9697f /data1/frs9596/frs9596g /data1/frs9495/frs9495q	/data1/frs0304/frs0304r0 /data1/frs0203/frs0203r0 /data1/frs0102/frs0102r0 /data1/frs0001/frs0001r0 /data1/frs9900/frs9900r0 /data1/frs9899/frs9899r0 /data1/frs9798/frs9798r0 /data1/frs9697/frs9697r0 /data1/frs9596/frs9596r0 /data1/frs9495/frs9495r0
Londu02	/repdata/sas_surveys/frs/frs0304/frs0304a /repdata/sas_surveys/frs/frs0203/frs0203d /repdata/sas_surveys/frs/frs0102/frs0102a /repdata/sas_surveys/frs/frs0001/frs0001d /repdata/sas_surveys/frs/frs9900/frs9900e /repdata/sas_surveys/frs/frs9899/frs9899f /repdata/sas_surveys/frs/frs9798/frs9798f /repdata/sas_surveys/frs/frs9697/frs9697f /repdata/sas_surveys/frs/frs9596/frs9596g /repdata/sas_surveys/frs/frs9495/frs9495q	/repdata/sas_surveys/frs/frs0304/frs0304r0 /repdata/sas_surveys/frs/frs0203/frs0203r0 /repdata/sas_surveys/frs/frs0102/frs0102r0 /repdata/sas_surveys/frs/frs0001/frs0001r0 /repdata/sas_surveys/frs/frs9900/frs9900r0 /repdata/sas_surveys/frs/frs9899/frs9899r0 /repdata/sas_surveys/frs/frs9798/frs9798r0 /repdata/sas_surveys/frs/frs9697/frs9697r0 /repdata/sas_surveys/frs/frs9596/frs9596r0 /repdata/sas_surveys/frs/frs9495/frs9495r0

The FRS data exists in both in a series of hierarchical tables and also in the form of a flatfile. A description of both structures is given below.

The Hierarchical dataset

Background

The FRS database consists of about 24 (depending on year) hierarchical, normalised tables, each table relating to a particular level (e.g. household, benefit unit) or type of information (e.g. pensions). Three main versions exist: the first covering edited and imputed fully co-operating households (the main database used by analysts); the second covering unedited data, as received from the survey contractors; and the third covering partially co-operating households (held for reference). Other data sets with different structures exist for non-responding households and other types of information relating to the data, e.g. a transactions data set of edits applied.

General-purpose tables covering household, benefit unit, adult and child information will contain a record for each household, benefit unit, adult and child in the sample. Other more specialised tables will contain records on the basis of the relevant routing of the questionnaire/circumstances of respondents. For example, the owner table will only contain records for those households who are owner-occupiers. Similarly, the benefits table will only contain records for benefits that are received by respondents. Individual records are uniquely identified by a combination of one or more key variables.

The household table (and, from 2000-01, benunit and adult tables too) also hold the variable 'gross', which is a weight variable. This serves two purposes: firstly to scale figures to the total population and secondly to compensate for non-response to the survey. It is advisable

to use grossed data for all analyses to reduce the amount of bias introduced by non-response.

Key variables

The highest level in the hierarchy is the household level. Records in this table (*househol*) are identified by the key variable *sernum* (serial number). Serial number is made up of a combination of the area, address and household codes, plus an additional figure to denote the year. Each household may consist of a number of benefit units, records for which are identified by the additional key variable *benunit*. Each benefit unit (and household) will consist of a number of adults and/or children, whose records are each identified by a unique *person* number as the third key. (Note *sernum* and *person* are enough to uniquely identify individuals in the sample.)

Below these main levels, other key variables exist depending on the table of interest. For example, an individual may receive a number of State and non-State benefits. Information about each benefit is held in a separate record in the *benefits* table. The key variables *sernum*, *benunit*, *person*, plus the additional key, benefit, uniquely identify each record.

Similarly, a household that is buying its house with one or more mortgages may have one or more endowment policy/PEP/Unit Trust investments covering each loan. Information on these policies is held in the *endowmnt* table. Each record is referenced by *sernum*, *mortseq* (mortgage sequence number) and *endowseq* (policy sequence number relating to that mortgage).

Key variables of the same name have the same coding frame across the database and can be used to relate information from different tables. For example, to look at the type of direct payments made by respondents alongside the total amount that is paid, the tables *benefits* and *dsspay* would be related by the keys *sernum*, *benunit*, *person*, *benefit*. Similarly, to look at the age of recipients of benefits, the *benefits* and *adult* tables would be related using *sernum*, *benunit*, *person*. Or to look at contributions made by someone outside the household to mortgages, the tables *mortgage* and *mortcont* would be related by *sernum*, *mortseq*.

Normalisation principle

As far as possible, the database is normalised, i.e. duplication is eliminated and redundancy (number of skipped – not asked - values in any given table) is kept to a minimum. This requirement influences where variables are held (although some level of redundancy is accepted to draw together information relating to a single topic). For example, although the question *anymon* (whether any money left in current account at the end of each month) relates to assets, it is held on the *adult* rather than *assets* table. This is because *anymon* relates to only one of 20 different asset types and is only asked once of each adult. If held as a separate variable on the *assets* table in the majority of records it would have a skipped value, and this would be wasteful. In the *benefits* table this type of requirement is met mainly by the use of multi-purpose variables (*var1-var3*), which are used to hold responses to a variety of different questions asked of different benefits.

Table Descriptions

The table below gives an overview of the information held in each of the tables of the FRS data set, together with key variables used to identify individual records. For a more detailed description including variable names, please refer to the full year-specific documentation.

Table	Key variables ¹	Description
accounts	SERNUM, BENUNIT, PERSON, ACCOUNT (<i>account type, held by each person</i>)	Income from interest/dividend bearing assets and savings together with (for a subset of records) the value of National Savings products for the accounts/investments held by adults. Each record relates to a type of investment (current account, savings account etc). Adults may have more than one type of investment, each record giving the total interest/dividends received (if they have more than one account of that type). For National Savings products, if the adult is not routed into the assets questions, a banded figure for the value of the investment is collected (those entering the assets block will have an accounts record but with this variable skipped).
admin	SERNUM	Household level fieldwork admin data, e.g. final outcome codes. Each record relates to a household in the sample.
adult	SERNUM, BENUNIT, PERSON	Responses to various questions asked of adults. Each record relates to an adult in the sample (complete coverage of all adults).
assets	SERNUM, BENUNIT, PERSON, ASSETTYPE (<i>asset type</i>), SEQ (<i>sequence number for that group of assets</i>)	Value and other information about assets and savings held by adults. Unlike the accounts table, each record relates to an individual investment (savings account, ISA, PEP, shares with a single company etc), except for certain National Savings Products where the total value of specific types (e.g. income bonds, capital/deposit bonds) are recorded together. Adults may hold more than one investment of the same type. Data are collected for the subset of adults routed into the assets block of the FRS questionnaire. Assettype has the same coding frame as account on the accounts table. Seq is the sequence number through one or a group of assets, as determined by the block of the questionnaire, e.g., one block covers, NSB ordinary and investment accounts, TESSAs, Gilts, and other types of saving. Please refer to more detailed documentation for the maximum number of records per asset type.
benefits	SERNUM, BENUNIT, PERSON, BENEFIT (<i>benefit type, held by each person</i>)	Amount and other information on State and non State benefits received by adults in the sample. Also includes information on the total amount of direct payments as part of IS/JSA, as well as information on Social Fund loans held and future receipt of some benefits. Each record relates to an individual benefit. An adult may receive more than one benefit but not more than one of the same type.
benunit	SERNUM, BENUNIT	Benefit unit level data. Each record relates to a benefit unit in the sample (complete coverage of all benefit units). Up to 7 benefit units.
care	SERNUM, BENUNIT, NEEDPER (<i>person receiving the care</i>)	Information on those needing care. Each record relates to an individual in the household or various categories of individual outside the household (e.g. parent or child outside the household/client of a voluntary organisation). Needper is therefore

¹ to identify individual records – in order of hierarchy

Table	Key variables ¹	Description
		equivalent to person for household members, but has additional codes for non-household members. For these cases, benunit is set to 1.
child	SERNUM, BENUNIT, PERSON	Responses to various questions relating to children. Each record relates to a child in the sample (complete coverage of all children). Information is collected by proxy from responsible adults.
dsspay	SERNUM, BENUNIT, PERSON, BENEFIT (<i>benefit type</i>), DSSPAY (<i>type of DSS direct payment, within each benefit</i>)	Information on the payments deducted from Income Support/Jobseeker's Allowance by the DSS to pay directly for different items (rent arrears, water charges, fines, maintenance payments etc). Each record relates to a type of deduction. Benefit type is either IS or JSA direct payments, using the same keys as in the benefits table. All variables in this table are key variables. Up to 10 different direct payment types can be identified at dsspay.
endowmnt	SERNUM, MORTSEQ (<i>mortgage sequence number</i>), ENDOWSEQ (<i>endowment policy sequence number, within each mortgage</i>)	Information on endowments/pension plans/ PEPs/Unit Trusts etc being used to cover the mortgage. Each record relates to an individual policy. A household buying their property with a mortgage may have more than one policy covering more than one loan (to a maximum of 4 policies per loan).
Extchild	SERNUM, BENUNIT, EXTSEQ (<i>sequence number of children living outside the benefit unit</i>)	Information on children aged 16-24 living outside the household/benefit unit who are currently receiving full or part time education. Each record relates to an external child to a benefit unit. Each benefit unit may have more than one 16-24 year old living outside the household (to a maximum of 4).
Househol	SERNUM	Information collected at the household level. Each record relates to a household (complete coverage of all households).
Insuranc	SERNUM, INSSEQ (<i>sequence number of insurance policy within household</i>)	Information on insurance policies held by household members. Each record relates to an individual policy. Binary variables POLINS1-14 identify household members covered by each policy (e.g. POLINS1=1 means person 1 is covered). A household may have more than one insurance policy (to a maximum of 6).
Job	SERNUM, BENUNIT, PERSON, JOBTYPE (<i>jobs held by each adult</i>)	Information on jobs held by an adult as an employee or self employed. Each record relates to an individual job. Up to 3 jobs may be identified. Also includes details of the last job held for those not currently working but who have worked in the last 12 months.
Maint	SERNUM, BENUNIT, PERSON, MAINTSEQ (<i>sequence number of maintenance payments made by each person</i>)	Information on maintenance payments made by adults. Each record relates to an individual maintenance payment. An adult may have more than one payment (to a maximum of 4).
Mortcont	SERNUM, MORTSEQ	Information on contributions made by someone outside the household towards mortgage payments of owner

Table	Key variables ¹	Description
	(mortgage sequence number), CONTSEQ (contribution sequence number, within each mortgage)	occupiers. Each record relates to an individual contribution. Each mortgage (mortseq) may have more than one contribution (contseq), to a maximum of 6.
Mortgage	SERNUM, MORTSEQ (mortgage sequence number)	Information on mortgages held on the property for house purchase or essential repairs. Each record relates to an individual mortgage. The full set of questions are asked separately for each purchase loan (to a maximum of 2). Mortseq does not necessarily take consecutive values: 1 and 2 relate to house purchase and will be consecutive, i.e. no household has mortseq=2 without mortseq=1. However, mortseq=3 relates to loans for essential repairs (including for those who own their property outright). A household may have mortseq=3 only, or mortseq=1 and mortseq=3. Whether a household has taken out loans other than for house purchase (and reason why) is held on the owner table.
Oddjob	SERNUM, BENUNIT, PERSON, ODDSEQ (odd job sequence number, odd jobs held by an adult)	Information on odd jobs or occasional fees for work or professional advice carried out in the last four weeks by an adult (includes work as a babysitter or mail order agent). Each record relates to an individual odd job/work as babysitter/mail order agent. This does not include any regular commitment. Information is held on up to 3 odd jobs, plus one record for babysitting and/or mail order work.
Owner	SERNUM	Information on owner occupiers (those buying with a mortgage and owned outright). Each record relates to such a household. Includes those who part own/part rent (shared ownership), who will also have a renter record.
Penamt	SERNUM, BENUNIT, PERSON, BENEFIT (benefit type), AMTTYPE (amount type for each benefit)	For those who are in receipt of retirement pension or widow's benefits (Widow's Pension, Widowed Mother's Allowance, Bereavement benefit) and who consult an order book, details of pension components as recorded on their order book. Each record relates to one component (e.g. Basic pension, age addition, Attendance Allowance). An adult may be in receipt of more than one benefit and each benefit may have more than one component. Up to 18 components can be identified at amttype.
Penprov	SERNUM, BENUNIT, PERSON, STEMPPAY, PROVSEQ	Introduced for 2001-02. Holds details of pension provision. Each record relates to one pension – stemppay describes the type of pension (occupational/personal/stakeholder).
Pension	SERNUM, BENUNIT, PERSON, PENSEQ (pension sequence number)	Information on non-State pensions held by an adult. Each record relates to an individual pension. Like mortseq, pensseq may not have consecutive values, since it relates to the routing of individual pension types. Please see more detailed documentation for the maximum number of records per pension type.
Rentcont	SERNUM, RENTSEQ (sequence number for	Information on contributions made by someone outside the household towards rent of those renting their property. Each record relates to an individual contribution. Each household may have more than one

Table	Key variables ¹	Description
	<i>contribution to rent</i>	contribution (rentseq), to a maximum of 5.
Renter	SERNUM	Information on renters (shared ownership, renting, rent free and squatting tenure types). Each record relates to such a household. Those who part own/part rent will also have an owner record.
Vehicle	SERNUM, VEHSEQ (vehicle sequence number)	This table is rotated off (ie the questions are not asked) every other year starting from 2000-01. Information on vehicles owned or in continuous use by household members. Each record relates to a vehicle. Each household may have more than one vehicle (to a maximum of 8).

The Flatfile

Whereas data for the same benefit unit is split into numerous tables in the hierarchical dataset, the flatfile contains rows for each benefit unit in the years' dataset. The following example shows how the data for a particular household is held in both formats:

On the hierarchical dataset, the data for this household is linked by a serial number on the HOUSEHOL table. For example, in this case of a household made up of a family of four; one benefit unit with information at this level is stored on the BENUNIT table. Information about the husband and wife is then stored on the ADULT table, with a separate record for each person. Similarly, information on the two children is held on the CHILD table (one record for each child).

The flatfile would take all this information about this benefit unit (all four family members) and store it on one single line starting with household variables, then benefit unit variables, head of household variables, spouse variables, child 1 variables, and so on.

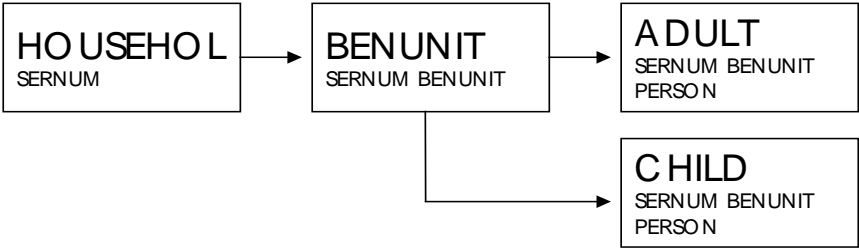
(See 'FRS datasets' flow chart below)

Variables have different names on either dataset, despite holding identical information. Variables from the hierarchical datasets are mapped to the flatfile using suffixes to distinguish the person to whom the variable relates where necessary. Continuing the example from above:

	Hierarchical Variable (Table)	Flatfile variable
Age of Husband	Age (Adult)	AgeHd
Age of Wife	Age (Adult)	AgeSp
Age of First Child	Age (Child)	AgeC1
Age of Second Child	Age (Child)	AgeC2
Amount of Child Benefit received	Benamt (Benefits)	CBSp
Etc.....		

FRS Data sets

Hierarchy



BU Flat file

HH variables	BU variables	Head variables	Spouse variables	Child 1 variables	Child 2 variables	Child 9 variables
-----------------	-----------------	-------------------	---------------------	----------------------	----------------------	-------	----------------------

Guide to the guides (1)

This is a guide to how to use the documentation. Rather than going through each document separately it takes a look at what you would do to answer particular questions. Click on the page number to jump to the item.

Introduction.....	1
Where do I start?	1
What does the FRS have on...?	1
I've found the question, now which variable is that?	2
What does this variable mean?	2
What does this answer mean?	2
Who is asked this?	2
How reliable is this?.....	3
What about earlier years?	3
What about earlier releases?	3
This result looks odd, what's going on?	3

Introduction

The two most useful documents are the *Questionnaire Instructions* (in Word) and *Hierarchical Documentation* (in Excel, under 'Metadata'). Both are year specific. *Questionnaire Instructions* has the exact question asked for every variable, together with interviewer guidance, and sometimes an indication of who is asked the question and some of the checks carried out. *Hierarchical Documentation* holds every variable with its label, possible values, and technical info.

Hierarchical Documentation also includes information on derived variables, which we create and which therefore won't be in the questionnaire instructions.

The FRS publication gives background information and some methodological details.

For flatfile users, *Flatfile Mapping* may have to be used to link flatfile variables to the appropriate hierarchical variables.

Where do I start?

The 'general', ie non-year specific, guidance is suitable for new users. *Introduction to the FRS* is a basic guide to using the survey. *Analysing the FRS* gives advice on how to analyse the datasets and weight data, with some example SAS programs. The FRS publication introduction and methodology give an overview of how the survey works.

What does the FRS have on...?

The best place to start is probably *Questionnaire Instructions*.

If you are interested in a broad area (pension income) then start with the list of question blocks. There are two lists, one at the start which covers questions asked once for each household and one half way through (exact page reference given after the first block) for questions asked to each benefit unit/adult.

If you are interested in something more specific (health insurance policies) then you can try and identify the relevant blocks, or you can do a word search on the name. Be careful – a search for insurance policy will fail if the questionnaire talks about insurance policies. Try 'insurance' or 'insurance pol'. If you are getting a lot of matches, try *Hierarchical*

Documentation. Again do a search on relevant words. This document only has the short labels for each variable so there are fewer words to look through, but things may not be labelled as you expect. For example if you wanted Educational Maintenance Allowance, neither of the three words would find the match as the label refers to EMA.

I've found the question, now which variable is that?

Generally the variable will follow the question name. Where more than one answer may be correct it may feed into separate variables, which will be the question name with a suffix. For example, the question Emppay "Do you contribute to any of these pension schemes" shows three types of pension scheme, and a fourth option of 'none'. The answer to scheme type1 goes into variable emppay1, the answer to type 2 goes into emppay2 and 3 into emppay3. More than one of the three variables may have a positive answer.

If you can't find it, *Hierarchical Documentation* gives the question name for each variable. It also gives the table the variable will be found on. Be careful, some variables may be on two tables (eg age on adult and child).

Flatfile Mapping will take you from the hierarchical name to the flatfile name.

What does this variable mean?

Two options, depending on whether it is a derived variable or not.

In either case, *Hierarchical Documentation* has every variable with its label beside it. It will also have an indicator for whether or not it is a derived variable, and if not derived it will give the question it comes from.

If it is not derived *Questionnaire Instructions* gives the exact question asked, with some details and guidance. There is an index of question names at the end. Sometimes a question will lead to several variables with different suffixes, eg the question emppay matches variables emppay1, emppay2, emppay3, so if doing a search knock off any trailing digits.

There are also multi-purpose variables Var1, Var2,... on the benefits table, which hold different information for different benefits. Look at *Benefits* for further information.

If it is derived, the DV spec will give the exact definition and derivation. If you can't immediately identify the required spec, use *Derived Variables Metadata*.

What does this answer mean?

All the answers are held in numeric format. *Hierarchical Documentation* has the key.

Some variables hold the answers 1 for yes, 2 for no and 3 for none. These are set-type variables. They come in groups (emppay1 emppay2 emppay3) which have come from one question (emppay) which has several options (1, 2, 3) plus an option (4) for none of these. More than one of the first three can apply. So if they answer yes to at least one of 1, 2, 3 the relevant variable(s) holds 'yes' and the other(s) 'no'. If they say none, all three variables will hold 'none'. There is never a mixture of 'yes's and 'none's in one group of set types.

Who is asked this?

Initially, *Questionnaire Instructions* gives you some idea. If you want the exact routing *Questionnaire Routing* holds it. Unfortunately this is written in computerese, and can be hard to follow.

More generally, the methodology chapter of the publication gives details of who is eligible for the survey.

How reliable is this?

The methodology chapter in the FRS publication includes estimates of standard errors and design factors for certain estimates. Multiply the two to get the true standard error. It also holds details of the imputation and editing we do – areas with a high level of imputation may be less robust. The assets data is a particular problem.

What about earlier years?

You can look up earlier years' information in the equivalent documentation, but the spreadsheet *Changes Metadata* gives brief details of changes made for that year.

What about earlier releases?

Releases Documentation describes the changes between different releases of the same years data.

This result looks odd, what's going on?

The FRS Users' Database holds previous queries on odd results, generally problems with one or two variables and sometimes just with a single respondent. Look here to see if one of them could be affecting you. If you still have a problem then please contact one of the FRS team.

Guide to the guides (2)

Several documents are available to users that summarise and explain some key aspects of the Family Resources Survey. These are listed and described below.

Introductory documentation (not year specific).....	2
<i>Introduction to the FRS.....</i>	<i>2</i>
<i>Guide to the guides (1).....</i>	<i>2</i>
Documentation for users who are new to the FRS.....	2
<i>Questionnaire Instructions.doc.....</i>	<i>2</i>
<i>Imputation.....</i>	<i>2</i>
<i>Hierarchical documentation.....</i>	<i>2</i>
<i>Flatfile Mapping.....</i>	<i>3</i>
<i>Usage.....</i>	<i>3</i>
<i>Derived Variables Table</i>	<i>3</i>
<i>Derived Variable Specifications.....</i>	<i>3</i>
Documentation for users more accustomed to the FRS	4
<i>Interviewers Guide to changes.doc.....</i>	<i>4</i>
<i>Changes Metadata.....</i>	<i>4</i>
<i>Questionnaire Routing</i>	<i>4</i>
<i>Electronic Questionnaire.....</i>	<i>4</i>
<i>Releases Documentation.doc.....</i>	<i>4</i>
<i>Editing</i>	<i>4</i>
<i>Data summary 1:Missing values.....</i>	<i>4</i>
<i>Data summary 2:Missing benefit values.....</i>	<i>4</i>
<i>Data summary 3:Min/Max/Mean</i>	<i>4</i>
<i>Accounts Table.....</i>	<i>4</i>
<i>Assets Table</i>	<i>4</i>
<i>Assets Table Documentation.....</i>	<i>4</i>
<i>Benefits Table Documentation</i>	<i>4</i>
<i>Benefits.....</i>	<i>4</i>
<i>Benefits Map.....</i>	<i>5</i>
<i>Period Code Conversion.xls</i>	<i>5</i>
<i>Derived Variable Changes.doc.....</i>	<i>5</i>
Other documentation available.....	6
<i>Interviewers' Showcards.doc.....</i>	<i>6</i>
<i>Pocket Guide to Benefits.doc</i>	<i>6</i>
<i>Pocket Guide to Savings.doc.....</i>	<i>6</i>
<i>Pocket Guide to Mortgages.doc.....</i>	<i>6</i>

Introductory documentation (not year specific)

Introduction to the FRS

What it sounds like!

Guide to the guides (1)

Shows you how to use the documentation to answer various questions

Documentation for users who are new to the FRS

Questionnaire Instructions.doc

This gives the interviewers' instructions for the questionnaire, and is useful when trying to understand the context and routing of the questions. However, it should not be treated as definitive: instructions are updated manually and may therefore be prone to error. For documentation generated directly from the FRS interview program, see Routing documentation.

Imputation

This summarises variables with highest levels of missing responses and imputation methods used.

Hierarchical documentation

This spreadsheet describes the hierarchical data set, including documentation that relates the data set back to the FRS questionnaire:

- ◆ Column A – TABLE – The source table on the hierarchical dataset.
- ◆ Column B – VARIABLE – The variable name on the hierarchical dataset.
- ◆ Column C – VAR_FMT – The name of the SAS format associated with the variable.
- ◆ Column D – LABEL – The label associated with the variable.
- ◆ Column E – FRSVALUE – List of numerical values associated with the SAS format.
- ◆ Column F – FMTVALUE – The value/label of the SAS format associated with the variable.
- ◆ Column G – MINVAL – Refers to the FRS questionnaire, minimum value which the variable can take.
- ◆ Column H – MAXVAL – Refers to the FRS questionnaire, maximum value which the variable can take.
- ◆ Column I – DERIVED – Indication of whether the variable is derived or not. Takes the value of 1 if the variable is derived and 0 if it is not.
- ◆ Column J – PER_VAR – Period code variable associated with amount variable if applicable.
- ◆ Column K – BENEFIT – Benefit key associated with document Benefits. Non-benefit variables take the value zero.
- ◆ Column L – QUESTION – Refers to the FRS questionnaire, source question associated with variable.
- ◆ Column M – TYPE – Whether the variable is an integer (I), floating point (F) or date (D) type variable.
- ◆ Column N – BLOCK – Refers to the FRS questionnaire, block of questions where the variable can be found.
- ◆ Column O – VARSEQ – Refers to the FRS questionnaire and the sequence in which questions are asked.
- ◆ Column P – USAGE – Description of the type of the variable, e.g. frequency, categorical etc. Further information on this (e.g. whether an amount variable has been converted to a weekly value) is held in the document Usage.

Flatfile Mapping

This spreadsheet describes the mapping between the flatfile and hierarchical datasets. The following information is shown:

- ◆ Column A – Flatfile variable name
- ◆ Column B – Full flatfile variable names (expanded list of column A summary)
- ◆ Column C – The source table on the hierarchical dataset
- ◆ Column D – The source variable name on the hierarchical dataset
- ◆ Column E – Usage (amount, categorical etc)
- ◆ Column F – Whether a derived variable
- ◆ Column G – The original question name
- ◆ Column H – The name of the SAS format applied (if required)
- ◆ Column I – Key value (if required), e.g. key to show which benefit
- ◆ Column J – The value of the SAS format applied to key value (if required)
- ◆ Column K – The label applied

The remaining columns show the decode values of the label applied (if required)

Usage

For each variable this gives its type, e.g. frequency, categorical etc. Further information on the codes (e.g. whether an amount variable has been converted to a weekly value) is attached.

Derived Variables Table

This spreadsheet lists all the derived variables (DVs) on the FRS.

- ◆ Column A – TABLE – The source table on the hierarchical dataset.
- ◆ Column B – DERIVED VARIABLE – The variable name on the hierarchical dataset.
- ◆ Column C – DESCRIPTION
- ◆ Column D – CATEGORY NUMBER & Column E – CATEGORY – Gives the area into which the DV falls.
- ◆ Column F – SPEC – The name of the specification which gives further details of the variable
- ◆ Column G – FLAT FILE MAPPING – The name of the flatfile equivalent, which will have the usual flatfile suffixes attached where necessary.
- ◆ Column H – LEAD USER – This may be the group which required the DV, or the purpose for which the FRS team created it.
- ◆ Column I – CHANGES – flags up any changes to the DV since the previous year.

Derived Variable Specifications

This directory contains all the derived variable specifications – showing how they are derived. The *specname* for each DV can be found by using the derived variable table mentioned earlier, but in most cases it is the same as the DV name.

Some variables use constants in their derivation. More information is held in **Derived Variables Metadata**

Documentation for users more accustomed to the FRS

Interviewers Guide to changes.doc

This guide, for interviewers and analysts, highlights questionnaire changes between survey years. The document also covers changes to the wording of particular questions.

Changes Metadata

This spreadsheet shows the variable changes to the hierarchical database that have taken place since the previous survey year and links them to **Interviewers Guide to Changes**. It covers new and dropped variables, rotation of questions and category changes within variables.

Questionnaire Routing

This gives the questionnaire routing, checks and computations in more detail. It is generated from the interview itself and is more definitive than the interviewers' instructions, but is difficult to follow. Once a variable on the database has been identified, this should be mapped to the original question name (using the hierarchical or flatfile documentation). The question name can then be searched for in this document.

Electronic Questionnaire

The questionnaire itself. This is a piece of software which can be installed on your machine so you can work through an interview.

Releases Documentation.doc

Outlines the changes between releases for each survey year.

Editing

Gives details of the types of edits carried out by the DWP

Data summary 1:Missing values

This gives more detail on imputation: the number and percentage of missing values and the method of imputation used for all variables.

Data summary 2:Missing benefit values

This gives more detail on benefit editing: the number and percentage of missing values for each benefit, the number imputed, and the number left as missing

Data summary 3:Min/Max/Mean

A summary of minimum and maximum values.

Accounts Table

Assets Table

These spreadsheets outline the data base structure for the hierarchical ACCOUNTS and ASSETS linking questions asked to variable names.

Assets Table Documentation

Benefits Table Documentation

Technical notes describing these tables.

Benefits

A list of the (adult level) benefit questions with their corresponding benefit key numbers on the benefits table.

Benefits Map

This has two sheets. The first (benefit) outlines the questions asked for each benefit:

- ◆ Column A – BENEFIT KEY – This is key number used for this benefit on the benefit table
- ◆ Column B – ONS/NATCEN KEY – not required for dataset analysis
- ◆ Column C – QUESTION – The initial question asking whether they receive the benefit, as held on the adult table. 1Q2 means it is category 2 of question ben1q, and the answer is held in ben1q2.
- ◆ Column D – BENEFIT - Gives the name of the benefit
- ◆ Column E – ABBREVIATION – an abbreviation of the benefit name
- ◆ Columns F to K – BASIC INFORMATION – This lists the detailed questions which are asked for that benefit. The column headings are the benefit table variables onto which the answer is mapped.
- ◆ Columns L to Q – DOCUMENT CONSULTED INFORMATION – Shows which documents the respondent may consult for that benefit.
- ◆ Columns R to T – USUAL AMOUNT INFORMATION – Shows whether and when this is asked
- ◆ Column U – CURRENT – whether presently receiving the benefit.
- ◆ Column V – CURRENT –how many weeks they have been receiving the benefit.
- ◆ Columns W to AA – WHO GETS – Some disability benefits can be received by one person on behalf another. The benefit will be coded to the person who received it, these variables (WhoRec) show on who they are receiving it for.
- ◆ Columns AB to AF – NON STANDARD QUESTIONS – For some benefits, we wish to hold benefit-specific information. To save space, this information is mapped onto the multi-purpose variables Var<number>. Columns AB to AF show which questions are mapped onto which Var* and the second sheet gives further details...

The second sheet (var) holds details of the multi-purpose variables Var1-Var5 (though not all years require all five). These are used to hold information which is relevant to only a few benefits. For each of the questions shown on sheet 'benefits' cols AB to AF there should be a corresponding bit on this sheet giving the answer formats.

Period Code Conversion.xls

This spreadsheet shows the conversion factors used for weeklyising period variables.

Derived Variable Changes.doc

This document outlines the major changes to, additions and deletions of derived variables from the previous year.

Other documentation available

Interviewers' Showcards.doc.

The interviewers' showcards that are used in various questions when the respondent is asked to say in which of a number of categories they lie. For example, when asked to which ethnic group they belong, they are shown a card with different ethnicities on it.

Pocket Guide to Benefits.doc

The interviewers' pocket guide to benefits, as produced by the survey contractors.

Pocket Guide to Savings.doc

The interviewers' pocket guide to savings, as produced by the survey contractors.

Pocket Guide to Mortgages.doc

The interviewers' pocket guide to mortgages, as produced by the survey contractors.

Analysing the FRS

This takes you through some of the programming techniques used on the FRS. It assumes you know the basic structure of the FRS, either hierarchical or flatfile versions. To refresh your memory on this you can look at 'Introduction to the FRS'. Click on the page numbers to jump to a section

<i>Hierarchical and flatfile programming</i>	1
<i>Grossing</i>	1
<i>Procedures used</i>	1
<i>Example 1: Simple breakdown - DLA mobility recipients by employment status</i>	2
<i>Example 1a: Better breakdown - DLA mobility recipients by employment status</i>	3
<i>Example 2: Merging records</i>	5
<i>Example 3: Retain statement</i>	5
<i>Example 4: Using imputation flags</i>	7

Hierarchical and flatfile programming

Programs can be written to use either the hierarchical dataset or the flatfile. The programs below come in pairs; they solve the same problem, but one uses the hierarchical dataset, the other the flatfile. The advantage of the hierarchical dataset is that processing is a lot faster, less wasteful of space, and there are fewer variable names to comprehend. In particular two adults in a benunit have two records with the same variable names, while on the flatfile there would be one record with the variables suffixed HD for the head and SP for the spouse. The flatfile does not require a knowledge of the table structure nor require merging of datasets, and it could be argued that it is easier to understand for someone unfamiliar with the FRS.

Grossing

The variable GROSS3 can be applied to sample estimates so that analyses reflect the overall population. This involves the use of a set of adjustment factors that attempt to correct for differential non-response at the same time as they scale up sample estimates. These factors take into account demographic variables such as age, sex and marital status together with region and tenure.

Procedures used

Most FRS analysis can be produced with a combination of proc tabulates, proc freqs and merge and retain statements.

- Proc Freqs and proc tabulates produce tables of results. Proc freqs are simpler but less flexible, and the output is less clear.
- Merge statements are used with the hierarchical tables. They combine information from different tables.
- Retain statements are used to combine different records – for example to look at households with a certain age distribution you would need to compare the records for all adults in the household.

Other useful procedures are Proc format, which combines variables into classes, and ods, which outputs answers to Excel. Using keep statements will speed up processing.

Example 1: Simple breakdown - DLA mobility recipients by employment status

These programs are simple because only one table is required, and only a simple breakdown. If more than one table was required, we would have to use a merge statement in the hierarchical program, and if a more detailed output was required a proc tabulate would be better.

EXAMPLE HIERARCHICAL PROGRAM

```
/* *****  
*** DLA mobility by employment status ***  
*** EP 8 March 1999 ***  
*** EH 20/08/02 Updated for 0001 ***  
***** */  
  
/* This program uses the hierarchical FRS dataset - ASD14 */  
  
/* To find the employment status of adults in receipt of DLA mobility */  
  
rsubmit;  
libname frs0001 '/data1/latest/frs0001';  
endrsubmit;  
  
rsubmit;  
  
/* All the information is on the adult table, and it is a breakdown of  
numbers of adults, so no merging is required */  
  
proc freq data= frs0001.adult (keep = sernum benunit ben2q2 empstatb  
gross3);  
table ben2q2 *empstatb;  
weight gross3; /* To give estimate for the GB population - remove  
this line to get the distribution within the sample */  
run;  
endrsubmit;
```

EXAMPLE FLATFILE PROGRAM

```
/* *****  
*** DLA mobility by employment status ***  
*** EP 8 March 1999 ***  
*** EH 20/08/02 Updated for 0001 ***  
***** */  
  
/* This program uses the FRS flatfile - ASD14 */  
  
/* To find the employment status adults in receipt of DLA mobility */  
  
rsubmit;  
libname frs01 '/data1/latest/frs0001';  
endrsubmit;  
  
rsubmit;  
data testhd; /*This picks up data for the head of the BU*/  
set frs01.frs0001 (keep=sernum benunit qdlacahd qdlamohd empbhd  
gross3);  
flag=0;  
if qdlamohd=1 then flag=1;  
empa=empbhd; /*these create a variable which is the same for both  
heads and spouses, so the proc freq can combine them*/  
run;  
data testsp; /*This picks up data for any spouse in the BU*/  
set frs01.frs0001 (keep=sernum benunit qdlacasp qdlamosp empbsp  
gross3);  
flag=0;  
if qdlamosp=1 then flag=1;  
empa=empbsp;  
run;  
data test;  
set testsp testhd;  
run; /*combines the two so the proc freq looks at all adults*/  
  
proc freq data=test;  
table flag*empb;  
weight gross3; /* To give estimate for the GB population - remove  
this line to get the distribution within the sample */  
run;  
endrsubmit;
```

These could then be repeated for other years. Care must be taken to ensure the variables used are consistent, for example in some years there are 14 categories for employment status, in some years 13.

If you only required information on the head of the benefit unit, you could use a where statement (uperson=1) in the hierarchical program, or you could look only at qdlamohd and empbhd in the flatfile program. In this case you could feed the flatfile directly into the proc freq as we have done in the hierarchical program.

Example 1a: Better breakdown - DLA mobility recipients by employment status

There are several ways you can improve the above code. These apply equally to both programs, but the hierarchical program is used for the examples.

First of all, replace proc freq with a proc tabulate to make the output neater:

```
Proc tabulate data=frs0001.adult;
  Class ben2q2 empstatb;
  freq gross3;
  Table empstatb all, ben2q2 all;
Run;
```

The 'all' keyword produces a 'total' row and column. The variable before the comma defines the rows, after the comma the columns.

We do not need all the splits. Use a proc format to combine the two 'no' options on ben2q2, and another to label empstatb.

```
Proc format;
Value yn 1='yes' 2, 3='no';

Value empfmt 1='Self-Employed'
2='Full Time Employee'
3='Part Time Employee'
4='FT Employee temporarily Sick'
5='PT Employee temporarily Sick'
6='Industrial Action'
7='Unemployed'
8='Work related Govt training'
9='Retired'
10='Unoccupied under retirement age'
11='temporarily sick'
12='Long term sick'
13='Students in non advanced FE'
14='Unpaid Family Workers';
```

Apply this to ben2q2 and empstatb in the proc tabulate. Also add a format to the whole table to get rid of the excess decimal points (format=10.0 gives 10 characters, no decimal points per cell)

```
Proc tabulate data= frs0001.adult format=10.0;
  Class ben2q2 empstatb;
  freq gross3;
  format ben2q2 yn. Empstab empfmt.;/*note full stops after format
names*/
  Table empstatb all, ben2q2 all;
Run;
```

We only have class variables, which restricts the options available to proc tabulate. To get round this, we can either create a dummy variable which is 1 for every case, use another variable as a dummy variable (with great caution!) or use gross3 as a variable rather than a weight.

I shall use sernum as a dummy variable. I know it is on every single record, so it should not create problems. We can now use a weight rather than a freq statement, and can use sumwgt to give us the weighted number of cases. I can add a further column definition after the comma, all*n, which will give us the sample size. This is needed to warn us if the sample size is too small to draw conclusions from, and can be compared with the total number on the

original file (e.g. there are 41842 adults on the 2000-01 file, so if we are looking at all adults we should expect to see this figure as the total sample size).

I have added a keep statement to speed up processing by dropping unnecessary variables.

```
Proc tabulate data=frs0001.adult (keep=sernum ben2q2 empstatb gross3)
format=10.0;
  Class ben2q2 empstatb;
  Var sernum;
  weight gross3;
  format ben2q2 yn. empstatb empfmt.;
  Table empstatb all, (ben2q2 all)*sernum*sumwgt all*n;
Run;
```

Finally, I add the ODS command to output the results to Excel. Part one goes before the proc tabulate (replace horsfale with your username):

```
ods html path='/usr/users/horsfale'
      body='xxx.xls'
      style=minimal;
```

and part 2 goes after

```
proc download   infile='/usr/users/horsfale/xxx.xls'
                outfile='\\acn\home\horsfale\output.xls';
run;
ods html close;
```

The final program would be as follows:

EXAMPLE HIERARCHICAL PROGRAM

```
rsubmit;
libname frs0001 '/data1/latest/frs0001';
endrsubmit;

rsubmit;
Proc format;
Value yn 1='yes' 2, 3='no';

Value empfmt 1='Self-Employed'
2='Full Time Employee'
3='Part Time Employee'
4='FT Employee temporarily Sick'
5='PT Employee temporarily Sick'
6='Industrial Action'
7='Unemployed'
8='Work related Govt training'
9='Retired'
10='Unoccupied under retirement age'
11='temporarily sick'
12='Long term sick'
13='Students in non advanced FE'
14='Unpaid Family Workers';

ods html path='/usr/users/horsfale'
      body='xxx.xls'
      style=minimal;

Proc tabulate data= frs0001.adult (keep=sernum ben2q2 empstatb gross3)
format=10.0;
  Class ben2q2 empstatb;
  Var sernum;
  weight gross3;
  format ben2q2 yn. empstatb empfmt.;
  Table empstatb all, (ben2q2 all)*sernum*sumwgt all*n;
Run;

proc download   infile='/usr/users/horsfale/xxx.xls'
                outfile='\\acn\home\horsfale\output.xls';
run;
ods html close;
```

Example 2: Merging records

Suppose we want to know DLA mobility recipients by household tenure. Tenure information is on the household table, so for the hierarchical dataset we have to merge datasets.

Sernum is enough to uniquely identify the household we wish to attach to the adult record, so we use this to match records:

```
Data adult1;
merge frs0001.adult (keep=sernum ben2q2 gross3)
      frs0001.househol (keep=sernum tenure);
by sernum;
run;
```

Note the keep statements – the less information we read over the quicker processing will be.

The resulting dataset is at adult level, as that is the lowest level of any of the input datasets. We insert this into the program we used above:

EXAMPLE HIERARCHICAL PROGRAM

```
rsubmit;
libname frs0001 '/data1/latest/frs0001';
endrsubmit;

rsubmit;
Proc format;
Value yn 1='yes' 2, 3='no';

Value tenure
1='Rented from Council'
2='Rented from Housing Association'
3='Rented privately unfurnished'
4='Rented privately furnished'
5='Owned with mortgage'
6='Owned outright';

ods html path='/usr/users/horsfale'
      body='xxx.xls'
      style=minimal;

Data adult1;
merge frs0001.adult (keep=sernum ben2q2 gross3)
      frs0001.househol (keep=sernum tenure);
by sernum;
run;

Proc tabulate data=adult1 format=10.0;
Class ben2q2 tenure;
Var sernum;
weight gross3;
format ben2q2 yn. tenure tenure.;
Table tenure all, (ben2q2 all)*sernum*sumwgt all*n;

Run;

proc download      infile='/usr/users/horsfale/xxx.xls'
                   outfile='\\acn\home\horsfale\output.xls';
run;
ods html close;
```

Tenure information is at benunit level on the flatfile so we do not need to merge. We use the same program as before, using set statements to move from benefit unit level to adult level.

Example 3: Retain statement

Now suppose we want to know how many households contain a DLA mobility recipient, split by tenure type. Our DLA information is kept at adult level (hierarchical) or benunit level (flatfile) so we use a retain statement to move to a higher level.

EXAMPLE HIERARCHICAL PROGRAM

```
rsubmit;
libname frs0001 '/data1/latest/frs0001';
endrsubmit;

rsubmit;
Proc format;
```

```

Value yn 1='yes' 2, 3='no';

Value tenure
1='Rented from Council'
2='Rented from Housing Association'
3='Rented privately unfurnished'
4='Rented privately furnished'
5='Owned with mortgage'
6='Owned outright';

ods html path='/usr/users/horsfale'
      body='xxx.xls'
      style=minimal;

data house1; /*output is at household level*/
  set frs0001.adult (keep=sernum ben2q2 gross3); /*Input at adult level*/
  by sernum;
  retain dlaflag;
  if first.sernum then do; /*When you are at the first record for a new
household, set the flag to 0*/
    dlaflag=0;
  end;
  if ben2q2=1 then dlaflag=1; /*Check each record*/
  else dlaflag=0;
  if last.sernum then output; /*When you reach the last record for that
household, output it*/
run;

Data house2;
merge house1
      frs0001.househol (keep=sernum tenure);
by sernum;
run;

Proc tabulate data=house2 format=10.0;
  Class ben2q2 tenure;
  Var sernum;
  weight gross3;
  format ben2q2 yn. tenure tenure.;
  Table tenure all, (ben2q2 all)*sernum*sumwgt all*n;

Run;

proc download   infile='/usr/users/horsfale/xxx.xls'
                outfile='\\acn\home\horsfale\output.xls';
run;
ods html close;

```

Flatfile:

Our previous program used three data steps to get down from benunit to adult level. Now we are trying to go up to household level, so we can replace these with a similar retain statement to the hierarchical programs with an 'if' statement which picks up heads and spouses simultaneously.

EXAMPLE FLATFILE PROGRAM

```

rsubmit;
libname frs01 '/data1/latest/frs0001';
endrsubmit;

rsubmit;
Proc format;
Value yn 1='yes' 2, 3='no';

Value tenure 1='Rented from Council'
2='Rented from Housing Association'
3='Rented privately unfurnished'
4='Rented privately furnished'
5='Owned with mortgage'
6='Owned outright';

rsubmit;

ods html path='/usr/users/horsfale'
      body='xxx.xls'
      style=minimal;
data house;
  set frs01.frs0001 (keep=sernum benunit qdlacahd qdlamohd gross3);
  flag=0;

```



```

        if qdlamohd=1 or qdlamosp=1 then flag=1;
run;

Proc tabulate data=house2 format=10.0;
  Class ben2q2 tenure;
  Var sernum;
  weight gross3;
  format ben2q2 yn. tenure tenure.;
  Table tenure all, (ben2q2 all)*sernum*sumwgt all*n;
Run;

proc download   infile='/usr/users/horsfale/xxx.xls'
                outfile='f:\output.xls';
run;
ods html close;

```

Example 4: Using imputation flags

Imputation is the process whereby missing values, for chosen variables, are edited to valid values. The main objective of imputation is to maximise the information available to users for analysis. Furthermore, the imputation carried out simplifies the analysis for users and helps to secure the uniformity of analysis created from the FRS datasets.

The editing process is carried out on cases where a value has been given but following checking is believed to be incorrect, for example where a respondent has given a benefit amount which is not valid.

When imputation and editing are undertaken, transactions are applied to a copy of the dataset to ensure that the original dataset can still be accessed in its unaltered state. Transactions change the old value in the copied dataset to a new value, but they do not make any changes to the original dataset. Transactions allow the user to follow the audit trail and see how the values have been changed. The transactions are saved in a transaction dataset, which records the table, variable, serial number and related key variables, the old value and the new value of the record which has been changed.

For example if the variable ABSPAR (whether receiving any allowance from an absent partner) was changed from a 'don't know' response to a valid response of 'no' then the transaction would read:

FRSTABLE	FRSVAR	SERNUM	KEY1	KEY2	KEY3	KEY4	OLD_VAL	NEW_VAL
Adult	abspar	1234	1	2	.	.	-9	2

Where:

Key1=benunit

Key2=person

Old_val=original value in dataset (-8='refused', -9='don't know')

New_val=the new imputed/edited value which will be recorded in the release 1 dataset (2='no').

Using the transaction dataset it is possible to create your own imputation flags, as and when necessary, for whichever variable required. The transaction dataset can be merged with the FRS dataset to examine whether an individual case has been edited. The SAS program below is an example of how the transaction dataset can be accessed and imputation flags created.

EXAMPLE HIERARCHICAL PROGRAM

```

/*****
*** Program: p:\frs\shared\frs35\imputation\imputation flag.sas ***
*** By: Julie Stanborough ***
*** Date: August 1999 ***
*****/

/* run this code on release 1 on the hierarchical dataset */

```

```

/* change job and profit1 for different tables and variables as necessary */
rsubmit;
libname frs0001 '/data1/latest/frs0001';
endrsubmit;
rsubmit;

/* from the transaction dataset get all the cases where a transaction has
been */
/* created for profit1 on the job table. Rename key1-3 depending on */
/* the keys in the table */

data trans;
    set frs0001.transact (keep=sernum key1 key2 key3 frstable frsvar
old val new val
    rename=(key1=benunit key2=person key3=jobtype)
    where=(frstable='JOB' and frsvar='PROFIT1'));
run;

/* sort the transaction data by the keys */
proc sort data=trans;
    by sernum benunit person jobtype;
run;

/* merge the trans dataset with the job dataset */
/*keep only the cases with a profit1>0 and a trans entry*/
data proftran;
    merge frs0001.job (where=(profit1>0) in=j) trans(in=t);
    by sernum benunit person jobtype;
    if j and t;
run;

/* print all profit1 cases which have been imputed or edited */
/* to just examine imputed values add 'if old_val in(-8,-9)' in the proc
print step */

proc print data=proftran ;
run;

endrsubmit;

```

It should be noted that transactions are applied to the hierarchical dataset and therefore transactions can only be accessed via the hierarchical dataset and not the flatfile.

An alternative way to examine whether imputation has been carried out on a particular case is by setting up an imputation flag manually and comparing the pre and post imputation files. Using this method will not provide as much information but can be used on both the hierarchical and flatfile. The SAS program below is an example of how the imputation flags can be created using the flatfile.

EXAMPLE FLATFILE PROGRAM

```

/* SAS program to create imputation flags on the flatfile */
/* setting up a flag to show imputed cases of Attendance Allowance */

libname preimp '/data1/frs/frs978r0';
libname postimp '/data1/latest/frs9798';

data impflag;
    merge preimp.frs9798 (keep=sernum benunit qaahd)
    postimp.frs9798 (keep=sernum benunit qaahd)
    rename=(qaahd=postqaa);
    by sernum benunit;
    imp=0;
    if postqaa=1 and qaahd ne 1 then imp=1;
run;

proc freq data=impflag;
table imp;
run;

```

15 November 2002

Coverage: Scotland and Great Britain
Theme: Social and Welfare

Grossing factors for National Statistics from DWP Family Resources Survey

Department for Work and Pensions statisticians, in consultation with other departments, have been reviewing the grossing methodology for the DWP Family Resources Survey (FRS). The FRS is used as a source for National Statistics on income distribution and on the take-up of income-related benefits.

The current FRS grossing regime ensures that FRS-based estimates for Great Britain match population estimates derived from other National Statistics sources. However the regime has not sought to ensure that estimates for individual countries or regions match population estimates for those countries or regions. In practice estimates have not always matched. As a consequence, it has not been possible to produce reliable estimates of the **number** of individuals in households below various income levels for countries or regions.

As a result of the working group discussions, DWP statisticians envisage – subject to consultation – adopting a grossing regime which will:

- take account of the revision of population and related estimates in the light of the 2001 Census, once these revisions are made;
- take account of new evidence, expected next year, from comparisons of Census data and FRS data; these comparisons will be conducted by the Office for National Statistics;
- and, if possible, match FRS-based estimates for all countries and regions to official population estimates.

This cannot be implemented in full before next year. Given the demand for more robust low income estimates for Scotland, an interim grossing regime has been developed which matches to both GB and Scottish population estimates. The new interim regime will allow estimates of **numbers** of individuals below various income thresholds, in Scotland, to be published. The effects on estimates of the **percentage** of individuals below various income thresholds are small. (See note 4 in Notes for Editors.)

The new interim regime yields estimates, for low income counts in Great Britain, which are identical or very close to previously published estimates. (See note 2 in Notes for Editors.) DWP statisticians intend to revise Great Britain low income estimates in due course and will consult users of low income statistics as to the timing of this revision. In the meantime the estimates published in the last edition of Households Below Average Income, and the database underlying those estimates, remain the recommended source of information on patterns of low income for Great Britain. The Scottish Executive are today publishing estimates for Scotland, consistent with the new interim regime. Thereafter, all analysis specific to Scotland will be based on the new interim regime.

DWP statisticians will consult the users of other FRS-based National Statistics published by the Department on the application of a new grossing regime to those statistics.

Issued by:

Information and Analysis
Directorate
Department for Work and
Pensions

Telephone : 020 7962 8222
Press Office : 020 7238 0766

DWP website :
www.dwp.gov.uk

Statistician:

Raj Majumdar
Information and Analysis
Directorate
Department for Work and
Pensions
Room 452
The Adelphi
1-11 John Adam Street
LONDON
WC2N 6HT

Telephone : 020 7962 8222
Facimile : 020 7712 2001
email :
Rajib.Majumdar@dwp.gsi.gov.uk

Notes for Editors

1. The Family Resources Survey (FRS) collects information on the incomes and circumstances of private households in Great Britain. The survey covered around 24,000 households in 2000-01. The FRS grossing regime allows FRS data to be scaled-up to provide estimates of the numbers of households, in Great Britain, with various characteristics – e.g. with income below the national average. The process of grossing involves scaling-up the FRS to estimates of population, households and to a number of other key totals which come from a variety of National Statistics sources. Several Government Statistical Service surveys adopt this approach to weighting survey data.
2. The following table shows what would be the impact, of adopting the new interim grossing regime, on key low income estimates for Great Britain published in Households Below Average Income. Figures are presented under the new interim grossing methodology and changes from published estimates are marked in superscript.

Table 1: Great Britain – Percentage and Number of individuals in different groups below 60 per cent of median household income

	Below 60% contemporary median				Below 60% 1996/7 median (fixed in real terms)			
	Percentage		Number (millions)		Percentage		Number (millions)	
	BHC	AHC	BHC	AHC	BHC	AHC	BHC	AHC
All individuals								
1994/5	18	24	9.8	13.2 ^{0.1}	20	26	11	14.5 ^{0.1}
1995/6	17	23	9.4 ^{0.1}	13 ^{0.1}	19	26	10.7 ^{0.1}	14.5 ^{0.2}
1996/7	18	25	10.3 ^{-0.1}	13.9	18	25	10.3 ^{-0.1}	13.9
1997/8	18	24	10.3	13.5	18	23	9.9	13.1
1998/9	18	24	10.2	13.4	16	22	9.3 ^{0.1}	12.6 ^{0.1}
1999/00	18	23	10	13.3	15	20	8.3	11.4 ^{0.1}
2000/01	17	23	9.7	12.9	13	17	7.4 ^{0.1}	9.9 ^{0.1}
Children								
1994/5	23	32 ¹	2.9	4	26	34	3.3	4.4 ^{0.1}
1995/6	21	32 ¹	2.7	4	24	35 ¹	3.1	4.5 ^{0.1}
1996/7	25 ⁻¹	34	3.2 ^{-0.1}	4.3 ^{-0.1}	25 ⁻¹	34	3.2 ^{-0.1}	4.3 ^{-0.1}
1997/8	25	33	3.2	4.2 ^{-0.1}	24	32	3.1	4.2
1998/9	24	33	3.1	4.2	22	31	2.8	4
1999/00	23	32	3	4.1	19	28	2.4	3.6
2000/01	21	31	2.7	3.9	16	24 ¹	2	3
Working-age Adults								
1994/5	14 ⁻¹	20	4.9	6.6	16	21	5.4	7.2 ^{0.1}
1995/6	13 ⁻¹	19	4.5	6.5	15	21	5.1	7.1
1996/7	15	21 ¹	5	7 ^{0.1}	15	21 ¹	5	7 ^{0.1}
1997/8	15	20	5	6.7	14	19	4.8	6.5
1998/9	14	19	4.9	6.6	13	18	4.5	6.3 ^{0.1}
1999/00	14 ⁻¹	20	5	6.8	12	17	4.2	6 ^{0.1}
2000/01	14	19	4.9	6.6	11	16	3.9 ^{0.1}	5.4
Pensioners								
1994/5	21	27 ¹	2	2.6 ^{0.1}	25 ¹	31 ¹	2.3	3 ^{0.1}
1995/6	22	26 ¹	2.1	2.4	25	31 ¹	2.4	3 ^{0.1}
1996/7	21	27	2	2.6 ^{0.1}	21	27	2	2.6 ^{0.1}
1997/8	22	27	2.1	2.6	21	26 ¹	2	2.4
1998/9	23	27	2.2	2.6	21 ¹	24	2 ^{0.1}	2.3
1999/00	22	26	2	2.4	17	19	1.7 ^{0.1}	1.8
2000/01	22	25	2	2.3	16	15	1.5	1.4

Note: Numbers in superscript indicate the difference in the revised estimate presented *minus* the published estimate

3. Following consultation on the new interim grossing regime, DWP statisticians expect to publish revised estimates for all the HBAI Supplementary Tables. The effects of revisions are expected to be very small.
4. The following tables show the impact of adopting the new interim grossing regime on estimates, previously published in Households Below Average Income, of the percentage of various groups in Scotland below 60% of median income.

ESTIMATES WITH NEW INTERIM GROSSING REGIME

Table 2: Proportion of Individuals in Relative Low Income Households, below 60% of GB Median Income, Scotland

Year	Children		Working Age Adults		Pensioners	
	Before housing Costs	After Housing Costs	Before housing costs	After housing costs	Before housing Costs	After housing costs
	%	%	%	%	%	%
1994/95	26	28	16	18	23	28
1995/96	25	32	14	17	23	25
1996/97	30	34	16	20	24	28
1997/98	29	31	15	18	19	26
1998/99	27	31	15	19	21	27
1999/00	24	28	15	19	22	26
2000/01	25	30	17	22	20	24

Notes:

Small changes in estimates from year to year may not be significant in view of data uncertainties. Results are presented separately for income before housing costs and income after housing costs. For all results, Median Income refers to the median income of the whole Great Britain population.

Table 3: Proportion of Individuals in Absolute Low Income Households, below 60% of 1996/97GB Median Income, Scotland

Year	Children		Working Age Adults		Pensioners	
	Before housing Costs	After housing Costs	Before housing costs	After housing costs	Before housing Costs	After Housing Costs
	%	%	%	%	%	%
1994/95	28	30	18	20	26	33
1995/96	29	34	16	19	27	33
1996/97	30	34	16	20	24	28
1997/98	27	30	14	17	19	25
1998/99	26	29	14	18	17	23
1999/00	20	24	13	17	18	21
2000/01	18	21	14	17	13	14

Notes:

Small changes in estimates from year to year may not be significant in view of data uncertainties. Results are presented separately for income before housing costs and income after housing costs. For all results, Median Income refers to the median income of the whole Great Britain population.

PREVIOUS ESTIMATES WITH CONVENTIONAL GROSSING REGIME

Table 4: Proportion of Individuals in Relative Low Income Households, below 60% of GB Median Income, Scotland

Year	Children		Working Age Adults		Pensioners	
	Before housing Costs	After Housing Costs	Before housing costs	After housing costs	Before housing costs	After housing costs
	%	%	%	%	%	%
1994/95	25	27	17	18	23	28
1995/96	25	32	14	18	23	25
1996/97	29	34	16	20	24	28
1997/98	28	30	15	17	19	25
1998/99	26	30	15	19	20	26
1999/00	24	29	16	20	22	26
2000/01	25	30	17	22	20	23

Notes:

Small changes in estimates from year to year may not be significant in view of data uncertainties. Results are presented separately for income before housing costs and income after housing costs. For all results, Median Income refers to the median income of the whole Great Britain population.

Table 5: Proportion of Individuals in Absolute Low Income Households, below 60% of 1996/97GB Median Income, Scotland

Year	Children		Working Age Adults		Pensioners	
	Before housing Costs	After Housing Costs	Before housing costs	After housing costs	Before housing costs	After housing costs
	%	%	%	%	%	%
1994/95	27	30	18	20	26	33
1995/96	29	34	16	19	27	33
1996/97	29	34	16	20	24	28
1997/98	27	29	14	17	18	25
1998/99	25	28	14	18	18	23
1999/00	21	25	13	17	18	20
2000/01	17	21	14	17	13	13

Notes:

Small changes in estimates from year to year may not be significant in view of data uncertainties. Results are presented separately for income before housing costs and income after housing costs. For all results, Median Income refers to the median income of the whole Great Britain population.

- Statistical questions on revised low income estimates for Scotland should be addressed to Julie Goodall, Scottish Executive (0131 244 3004). Statistical questions on low income estimates for Great Britain should be addressed to Colin Wilkie-Jones, Department for Work and Pensions (020 7962 8232). Statistical questions on the FRS grossing regime should be addressed to Rajib Majumdar, Department for Work and Pensions (020 7962 8222).

FRS GROSSING REVIEW: BACKGROUND NOTE

1. Grossing regimes for sample surveys scale up the survey figures to allow estimates to be presented for the total number of people with a given characteristic; and to correct, as far as possible, for any biases due to some groups being less ready to take part in the survey than others are. After the launch of the Family Resources Survey (FRS) in 1992, DSS statisticians devised a grossing regime that seeks to correct for response biases in relation to age, gender, lone parents/all families with children, housing tenure and Council Tax Band - these last 2 aspects being a proxy for income. It does this by scaling up the sample counts to match independent 'control totals.'
2. An FRS grossing review was launched last year – to run in parallel with National Statistics quality review of Households Below Average Income (HBAI) statistics and Pensioners' Incomes (PI) statistics. This was to allow us to synchronise timings, to avoid having to change HBAI/PI results one year in response to HBAI/PI review recommendations and again, shortly thereafter, due to a revised FRS grossing regime. A formal Working Group - with members from DWP, Office for National Statistics, Scottish Executive and Inland Revenue - was set up earlier this year.
3. The Working Group identified an urgent need in relation to Scotland. Plausible estimates of the number of children in households with income below eg 60% of the median income – in Scotland, could not be prepared by the Scottish Executive (SE) using DWP's Households Below Average Income (HBAI) dataset. The figures jumped around from one year to the next, because HBAI's estimates of the *total* number of children in Scotland fluctuate implausibly. This is because the FRS grossing regime uses control totals for Great Britain, not for individual countries or regions; so it does not correct for year-on-year variations in response rates at country/region level. As a consequence, SE statisticians had to refrain from releasing results for the **number** of children in "poverty" in Scotland; and rely only on results for the **percentage**, which are published in HBAI. This issue had been raised in the Scottish Parliament, and SE Ministers had indicated that the problem would be addressed in time to present numbers in this year's Social Justice Annual Report, due for publication on 18th November.
4. The Working Group have considered:
 - the Scottish Executive's urgent needs; and longer term needs for sub-national results, given growing interest in results for English regions and Wales;
 - census-linked revisions to control totals – each of the departments producing control totals, used for FRS grossing, will be revising these in the light of results from the 2001 Census;
 - the vital role foreseen for the Census comparison study - our first opportunity to secure definitive estimates of FRS response biases, using 2001 Census data for non-respondents.

- other known deficiencies in the current grossing regime, including the need for accurate and consistent counts of (families with) children in place of the Child Benefit series currently in use.¹
5. The second component has become even more significant than officials expected in the summer, in the light of the significant changes to population estimates, stemming from the 2001 Census, announced by ONS a few weeks ago. The third component is particularly important because evidence emerging this year suggests that the FRS overstates the proportion of people in workless households.² This is likely to affect HBAI's low income estimates for any one year (though the effect on estimates of changes over time should be much smaller). At present we cannot quantify this overstatement, and therefore we cannot correct for it. We should be able to, when we have the results of comparisons of FRS data with Census data – the latter will include households who declined to give FRS interviews, and so will allow us to quantify non-response biases in the FRS.
 6. The sophistication of the FRS grossing regime, with its multiplicity of controls, stands us in good stead in many ways. But we now stand to pay a price for it:
 - The various control totals will all be updated in the light of new data from the 2001 Census.
 - But different control totals will be updated, by various Departments/teams, at different times over the next few years.

So if we took new information on board as soon as each tranche become available, given that we have just revised (to address Scottish Executive needs) we would be faced with revising HBAI and possibly other FRS-based statistics in each of three or four successive years.

7. The desirability of using the best data available at any one time has to be weighed against the undesirability of a succession of revisions to FRS-based statistics. The latter would be resource-intensive. They could also undermine confidence in the statistics - especially as we could not always guarantee that the revisions would take us closer to the truth, if we define this as the results that will emerge when all the revisions currently contemplated have been implemented.

¹ Child Benefit data is used as a proxy for the number of children. In this context it has a discontinuity, as more recent data understates the number of very young children

² This is discussed in a report on differences between the FRS and LFS, in estimates of the proportion of children in workless households, which can be found at:
<http://www.dwp.gsi.gov.uk/asd/frs/reports/index.html>

8. The Working Group considered a wide range of options and proposed the following strategy for the FRS grossing regime:

Stage A No more than modest revisions in the 2002 grossing round, feeding into the 2001/2 income statistics to be released next year, and into Scottish Executive results (to 2000/1) to be released in November.

Stage B. Major revisions in the 2003 grossing round, incorporating new (2001 Census-based) population estimates and the results of the FRS/Census comparisons. These revisions cannot be introduced sooner, because the data will not be available. Conversely, delaying them would delay our remedying of known deficiencies in FRS-based results; this would be difficult to justify.

Stage C. Final Census-based revisions in the 2005 grossing round.

7. For Stage A, DWP statisticians have devised a regime that considerably reduces spurious volatility in grossed population estimates for Scotland. It also adopts a consistent time series for the number of families with children, in Scotland and in Great Britain. The effects on results for Great Britain are very small.³

³ In relation to the PSA target on child poverty, which is monitored via HBAI estimates of the number of children in households with income below 60% of the median income in Great Britain, the new grossing regime leaves unchanged the published figures for 1998/9 – the baseline year – and for the latest available year, 2000/1.

Changes to FRS Grossing Regime: Great Britain

Background

Since 2000, Department for Work and Pensions statisticians, in consultation with other departments, have been reviewing the grossing methodology for the DWP Family Resources Survey (FRS).

As part of this review an interim grossing regime was introduced in December 2002 with the release of the 2001-02 FRS. This was a modest revision primarily targeted at improving low income estimates for Scotland.

With additional review work now completed, a new grossing regime for the survey is being issued with the 2003-04 FRS data. This new regime consists of both an enhanced set of control totals and incorporates data on a post-Census basis.

This review has focussed on the Great Britain grossing regime with the regime employed for Northern Ireland grossing not affected by these changes.

A new grossing regime

The new grossing regime follows a number of methodological investigations, which include comparisons of FRS sample data with the Census (the Census Non-Response work), as well as investigation of the need for and effect of certain controls within the interim system, and potential alternatives. Worklessness was considered within the Census Non-Response investigation, but the conclusion was the available data did not allow us to make robust adjustments for any potential worklessness bias on the FRS.

Two key changes have been made in moving to the new regime. The first has been to replace the controls on numbers of family units (singles and couples) in broad age categories, by controls on numbers of individuals by gender and much finer age categories. The second major change is that we have considered population controls at a Government Office Region level, and the new regime uses population controls at this level rather than only separating out Scotland from the Great Britain totals.

The interim regime for Great Britain contained control totals as follows:

- At a GB level:
 - male and female lone parents
 - households by CTB band
 - households by tenure
 - households in London, Scotland and the rest of GB (ie. three controls)
- For England and Wales (combined) and Scotland separately:
 - Singles males aged below 35
 - Single males aged 35-59
 - Single males aged 60 or more
 - Single females aged below 35
 - Single females aged 35-64
 - Single females aged 65 or more

- Couples where the male is aged 65 or more
- Couples where the male is below age 65
- Families with children
- Total number of children

The new “regional” regime contains the same GB level controls, and retains the control on “families with children” in England and Wales (combined) and Scotland, but replaces the Singles and Couples categories, and the total number of children, by regional population figures for the population by age and sex, using age bands (for males) 0-9, 10-19 dependents, 16-24 independents, 25-29, 30-34, 35-39, 40-44, 45-49, 50-59, 60-64, 65-74, 75-79, 80 plus, and (for females) the same but using 60-69 and 70-74.

The Family Resources Survey series

Two grossing factors are being supplied on the 2003-04 FRS release

GROSS2 – this is the previous, interim methodology, using pre-Census control totals to produce the grossing factors, and will be consistent with previous values of this variable on earlier releases.

GROSS3 - this is the new grossing regime, and also uses control total information on a post-Census basis (as far as is currently possible).

The new regime, GROSS3, has been back cast over the FRS series from 1994-95.

Family Resources Survey

Description of files provided with the ASCII export format

General

All files are provided in comma separated value (csv) format.

All files include a comma separated list of column names as the first row

For each **table** in the FRS hierarchical data structure, two files are provided - **table.dat** & **table.dic**.

One other file is provided - vallabs.fmt

Table.dat

The first columns contain the key variables in order of key, these are followed by base survey variables in alphabetical order, and then by additional derived variables in alphabetical order.

SAS Missing values are represented as follows :

- A** - Skipped
- B** - Don't know
- C** - Refused to answer
- D** - Unable to derive - (Applies to derived variables only)

Dates are represented in DD/MM/YYYY format.

transact.dat (special case)

Editing transactions are stored in a SAS dataset. This is also exported in ASCII format with the other survey data files. However, it should be noted that the representation of the data differs slightly from the other data files as follows:

- It contains some alpha-numeric variables (denoted by type=(**A**) in the transact.dic file)
- Dates are represented in DDMMYY:HH:MM:SS format.
- Skipped variables are represented by a dot for numeric variables, and a space for alpha-numeric variables
- In the variables **old_val** & **new_val**, representing the values before and after the transaction is applied, the skipped values **A**, **B**, **C** & **D** are represented by the negative numbers **-1**, **-9**, **-8** & **-2** respectively. (This is because transactions are stored initially in an Oracle database, where character representations for missing numeric variables are not allowed)
- Some transactions exist solely to create or delete benefits records. In these, the words **INSERT** or **DELETE** will appear in the **frsvar** variable, and dots will appear in the **old_val** & **new_val** variables.

Table.dic

These files contain selected data dictionary items for each table - (one row for each variable)..

The variables (rows) are listed in the same sequence as their column sequences in the data files.

The dictionary columns are listed and described below :

Table -	(Just confirms which table the information relates to)
Variable -	Variable name
Type -	(I), (I2) = Integer, (F), (F1), (F2) = Floating point, (D) = Date, (A) = Alpha-numeric
Key -	If a key variable, it's position in the key sequence (sernum=1, benunit=2 etc.) Otherwise undefined (represented by a dot)
Minval -	Minimum value - (based on original survey data - not always present)
Maxval -	Maximum value - (based on original survey data - not always present)
Width -	The field 'width' of the original survey data variable, before conversion to our hierarchical database structure - (It might be useful for some purposes)
Var_fmt -	The name of the SAS Format (i.e. set of value labels) as listed in the file vallabs.fmt (see below) This provides value labels for most categorical variables
Label -	Variable label or description - where available

Vallabs.fmt

In the FRS database, a SAS 'format' refers to a named set of value labels, which can be associated with one or more variables. In the vallabs.fmt file described below, each value, within a named format, occupies a single row.

The file contains no table or variable information. SAS 'Formats' are associated with variables by format name only. This association can be made by relating *fmtname* in the vallabs.fmt file to *var_fmt* in the *table.dic* files.

The formats meta-data may appear to be listed approximately in alphabetical order of the tables with which the formats are associated. But this effect is coincidental. The strict sequence is by alphabetical order of format name.

<i>Fmtname</i> -	Format name - can be associated with <i>var_fmt</i> in the <i>table.dic</i> files described above
<i>Start</i> -	Minimum value of range associated with label
<i>End</i> -	Maximum value of range associated with label
	(In practise, most of our value labels are for categorical variables, and so <i>Start</i> & <i>End</i> are the same)
<i>Label</i> -	Label describing the value, (or range of values)

Family Resources Survey

Procedure for creating SAS system files from SAS transport files

Background

Our policy is to transfer SAS data as SAS transport files, using PROC COPY with the SAS XPORT engine. This makes it possible for recipients to import the data into a variety of SAS versions, on a variety of platforms.

Points to note:

1. The formats catalogue is also exported as a SAS transport file, and will need to be converted back into a formats catalogue. (see below)
2. The SAS transport files are binary files, and it is important that you use binary mode when transferring them across platforms.
3. These files are created from SAS 6.12 running on DEC Alpha system running Digital Unix V4.0

The export process

The procedures used to create the dataset and formats catalogue exports are outlined below:

NB. The occurrences of & below, indicate where a specific name should be substituted, or a macro variable set.

***** Create XPORT transport files of datasets *****;

```
libname source "&Source_dir";
libname export XPORT "&path/&dataset.exp";
proc copy in=source out=export;
  select &dataset;
run;
```

***** Create formats XPORT file *****;

```
libname temp "&Temp_dir";
libname library "&Library_dir";
libname format XPORT "&path/formats.exp";
proc format library=library
  cntlout=temp.fmts;
proc copy in=temp out=format;
  select fmts;
run;
```

The import process

The import process is basically the reverse of the export process, using PROC COPY with the XPORT engine. The following code provides an example of how you could carry out this process.

***** Import SAS datasets from transport (XPORT) dataset files *****;

```
libname dest "&Dest_dir";
libname export XPORT "&path/&dataset.exp";
proc copy in=export out=dest;
run;
```

***** Import formats catalogue from ftransport (XPORT) formats file *****;

```
libname temp "&Temp_dir";
libname library "&Library_dir";
libname format XPORT "&path/formats.exp";
proc copy in=format out=temp;
proc format library=library
  cntlin=temp.fmts;
run;
```

Family Resources Survey

Procedure for creating SPSS system files from SAS export files

The procedure outlined below may be used to create SPSS system files from SAS export files. The GET SAS DATA command in SPSS provides the basis for this procedure. However, certain tweaks have also been added in order to produce useable SPSS files.

Reasons for needing special procedures:

Software

Certain versions of SPSS (e.g. SPSS for UNIX version 5.0) do not properly support the 'formats option' within the GET SAS DATA command. SPSS won't necessarily give an error when you try to use the formats option in such circumstances, but you will find that the option doesn't actually do anything. 'Formats' adds value labels and output formats to variables in the SPSS system file.

One user found that they had an 'incapable' version of SPSS operating on their UNIX platform (where the files were eventually to be stored), but used a more recent version of SPSS (Version 6.1.3 for Windows, operating on a desktop PC) to create useable SPSS system files. These files were then posted back to their UNIX platform using a file transfer program.

Users who find it necessary to use a desktop PC should be aware that this computer will need a sufficient amount of storage space. To convert a file, the computer will need enough space to hold at least - each **.spx** SAS export file, a **Formats.spx** formats file (if required), and the resultant **.sys** SPSS system file

Missing values

GET SAS DATA cannot convert SAS user-missing values into SPSS user-missing values. Instead, all SAS missing codes are converted to system missing in SPSS. The SAS export files are therefore provided by the DSS with any missing values set to real, negative numbers as follows:

Skipped	-1
Refused	-8
Don't Know	-9
DV - Missing data	-2

These values are then specified as SPSS user-missing values during the conversion procedure.

General procedure to be used for all files.

1. If the computer which is being used to create the SPSS system files cannot read files directly from the CD-ROM/Floppy disk, use a file transfer program to copy the file entitled **Formats.spx** (if required) and the **.spx** file(s) you wish to convert onto the hard disk of this computer. NB. The **.spx** files should be transferred as binary files.

2. In SPSS, run the following command syntax to create each SPSS system file:

```
get sas data='c:\filename.spx'  
  /formats='c:\formats.spx' .  
missing values all (-9 thru -1) .  
save outfile='c:\filename.sys' .
```