

**ABLE**

Purpose : To show whether a person is a man aged 60 or over but under 65 who does not need to be available for or able to work, under 60 and long-term sick/disabled, or under 60 and available for or able to work.

Created : 22 January 1993

Database Table : ADULT

Minimum Value : 1

Maximum Value : 3

Units : Integer

Validations :

Related Variables :

Children :

Parents :

Core variable/user : **ISM** PSM (Bold indicates lead user)

Amendments : VC - 5 March 1993 - Change to categories in code 2 as were too restrictive.

: VC - 22 April 1993 - To expand definition making clear which questions have been used.

: VC - To add the category indicating a man between 60 and 65 as do not have to be available for work.

: VC - 9 February 1994 - No version 30 update needed

: VE - 21 May 1996 - Initial Version 32 update needed - INJPD

replaced by INJLONG

Issue date : 6 June 1996

**1 Definition**

This variable is coded as

- 1 Indicating a man aged 60 or over but under 65 who does not need to be available for or able to work.
- 2 Indicating an adult under the age of 60 and long-term sick or disabled.
- 3 Indicating an adult under the age of 60 and available for or able to work (includes those already working).
- 1 Not applicable to this case - adults over National Insurance Retirement Pension age.

-2 Unable to derive due to missing values.

ABLE is derived from several variables in the ADULT table which indicate whether or not a person is a man aged between 60 and 65 or if he/she is long-term sick or disabled or any others under pension age who might be available for or able to work.

The following people are to be classed as long-term sick and disabled :

Those who are under Income Support pension age and not currently working ie where WORKING = 2 (no paid work within last 7 days), JOBAWAY = 2 (where the person does not have a job which they were away from) and the reason they were not looking for or did not want to work was because they are long term sick or disabled (NOLOOK/NOWANT = 5). Any other person whose illness or disability has lasted for more than 6 months, ie where ~~injpd~~ **INJLONG**= 2.

People who are classed as being able to work include all others under Income Support pension age who have not previously been coded.

NB - The FES definition only included those known to be sick or unoccupied. The questions ~~INJPD~~ **INJLONG** and NOLOOK/NOWANT are only asked of people under NI retirement age ie 60 for women and 65 for men.

## 2 FRS Specification

For each adult

Code Condition

- |    |   |
|----|---|
| 1  | From table ADULT<br><br>If SEX = 1 and AGE >= 60 or < 65  |
| 2  | From ADULT table<br>If AGE < 60 and <del>INJPD</del> <b>INJLONG</b> = 2 or<br>If AGE < 60 WORKING = 2, JOBAWAY = 2, LIKEWK = 1 and NOLOOK = 5 or<br>If AGE < 60 WORKING = 2, JOBAWAY = 2, LIKEWK = 2 and NOWANT = 5 |
| 3  | From table ADULT<br><br>If AGE < 60 and not coded above   |
| -1 | Not applicable to this case - people over NI pension age  |

-2 Unable to derive in this case

**3 Results**

Tabulation required to show number of adults falling into each category.

**4 Test Cases**

To be added at a later date.

**CAREAB, CAREAH, CARERE, CAREFR, CARECL, CAREOT, CARECB, CARECH  
HOURAB, HOURAH, HOURRE, HOURFR, HOURCL, HOUROT, HOURCB, HOURCH**

Purpose : To provide summary variables for adult carers  
 Created : 29 April 1996  
 Database Table : ADULT  
 Minimum Value :  
 Maximum Value :  
 Units : integer  
 Validations :  
 Related Variables : CHCARE CHHOUR (variables for children doing care)  
     Children :  
     Parents :  
 Core variable/user : ASD4A  
 Amendments : VE - 21 May 1996 - No initial Version 32 update needed  
 Issue date : 5 June 1996

**1 Definition**

The aim of these derived variables is to provide summary information on adult carers within the household. For most users' needs, information is only required on *who* is being cared for and *how much time* is spent caring. Any more detailed analysis of carers and those being cared for should be done using the SAS version of the care table.

The variables recording *who* is cared for are coded as:

*For each adult*

**CAREAB** total number of adults looked after in the same benefit unit (maximum value of 1, because can only be the adult's partner, if there is one)

**CAREAH** total number of adults looked after in the same household but different benefit unit

**CARERE** relatives outside the household looked after (maximum value of 1, since questionnaire records "relative" as a single response)

**CAREFR** friends and neighbours outside the household looked after (maximum value of 1, since questionnaire records "friend/neighbour" as a single response)

**CARECL** client of voluntary organisation outside the household looked after (maximum value of 1, since questionnaire records "client of a voluntary organisation" as a single response)

**CAREOT** others outside the household looked after (maximum value of 1, since questionnaire records "other non household" as a single response)

**CARECB** total number of children looked after in the same benefit unit

**CARECH** total number of children looked after in the same household but different benefit unit

0 For all variables - not applicable to this case - adult does not look after anybody in same benefit unit/household/outside household etc.

-2 For all variables - unable to derive due to missing values.

It follows that, to calculate the total number of individuals within the household cared for by an individual, add together CAREAB, CAREAH, CARECB and CARECH. To assess whether someone carers for others outside the household, look at CAREERE, CAREFR, CARECL and CAREOT.

The variables recording *how much* caring is done are coded as:

*For each adult*

<b>HOURAB</b>	total number of hours spent caring for adults in the same benefit unit
<b>HOURAH</b>	total number of hours spent caring for adults in the same household but different benefit unit
<b>HOURRE</b>	total number of hours spent caring for relatives outside the household
<b>HOURFR</b>	total number of hours spent caring for friends and neighbours outside the household
<b>HOURCL</b>	total number of hours spent caring for clients of voluntary organisation outside the household
<b>HOUROT</b>	total number of hours spent caring for others outside the household
<b>HOURCB</b>	total number of hours spent caring for children in the same benefit unit
<b>HOURCH</b>	total number of hours spent caring for children in the same household but different benefit unit

0 **For all variables** - not applicable to this case - adult does not look after anybody in same benefit unit/household/outside household etc.

-2 **For all variables** - unable to derive due to missing values.

It follows that to calculate the total number of hours spent caring for someone within the household add together HOURAB, HOURAH, HOURCB and HOURCH. To calculate the total number of hours spent caring, add together all variables.

## 2 FRS Specification

Set all variables equal to zero.

From CARE record, for each person needing care (NEEDPER), process WHOLOO(x) for all adults in the household looking after that person (x=1-9 - up to 9 adult household members).

If WHOLOO(x) is an adult in the **same BU** as the person needing care then

$$\text{CAREAB}(x) = \text{CAREAB}(x) + 1$$

$$\text{HOURAB}(x) = \text{HOURAB}(x) + \text{HOUR}(x)$$

If WHOLOO(x) is an adult in the **same HH but different BU** to the person needing care then

$$\text{CAREAH}(x) = \text{CAREAH}(x) + 1$$

$$\text{HOURAH}(x) = \text{HOURAH}(x) + \text{HOUR}(x)$$

If WHOLOO(x) is an adult looking after a **relative outside the household** (NEEDPER=21) then

$$\text{CAREER}(x) = \text{CAREER}(x) + 1$$

$$\text{HOURRE}(x) = \text{HOURRE}(x) + \text{HOUR}(x)$$

If WHOLOO(x) is an adult looking after a **friend/neighbour outside the household** (NEEDPER=22) then

$$\text{CAREFR}(x) = \text{CAREFR}(x) + 1$$

$$\text{HOURFR}(x) = \text{HOURFR}(x) + \text{HOUR}(x)$$

If WHOLOO(x) is an adult looking after a **client of a voluntary organisation** (NEEDPER=23) then

$$\text{CARECL}(x) = \text{CARECL}(x) + 1$$

$$\text{HOURCL}(x) = \text{HOURCL}(x) + \text{HOUR}(x)$$

If WHOLOO(x) is an adult looking after **others outside the household** (NEEDPER=24) then

$$\text{CAREOT}(x) = \text{CAREOT}(x) + 1$$

$$\text{HOUROT}(x) = \text{HOUROT}(x) + \text{HOUR}(x)$$

If WHOLOO(x) is an adult looking after a **child in the same BU** then

$$\text{CARECB}(x) = \text{CARECB}(x) + 1$$

$$\text{HOURCB}(x) = \text{HOURCB}(x) + \text{HOUR}(x)$$

If WHOLOO(x) is an adult looking after a **child in the same HH but different BU** then

$$\text{CARECH}(x) = \text{CARECH}(x) + 1$$

$$\text{HOURCH}(x) = \text{HOURCH}(x) + \text{HOUR}(x)$$

-2 If any variables are missing

**CURACT\*, POACCT\*, TESSCT\*, OTBSCT\*, OTBNCT\*, OTACCT\*, GILTSCT\*, UNTRCT\*, STSHCT\*, NSBOCT\*, SAYECT\*, PRBOCT\*, PEPST\***

**\* = I (individual); B (Benefit Unit); H (Household)**

Purpose : To show the different types of account held by individuals to allow checking of publication tables on assets

Created :

Database Table : ADULT, BENUNIT, HOUSEHOL

Minimum Value : 0

Maximum Value :

Units : Integer

Validations :

Related Variables :

    Children :

    Parents :

Core variable/user : ASD4A

Amendments : VE - 24 May 1996 - Initial update for V32 - ACCOUNT=19 no longer used  
: VE - 19 August 1996 - To update for V32  
: VE - 25 February 1997 - To include new DVs for PEPs and to add ACCOUNT=19 back in

Issue date :21 May, 2003

## 1 Definition

These variables are coded as

CURACTI etc

- 1 Not applicable to this case (no cases should exist - either individuals, benefit units or households have or don't have accounts).
- 2 Unable to derive due to missing values (not applicable to this specification: variables are a simple count of records).

The 1994/95 publication looks at assets in terms of different types of account held in the ACCOUNTS table. The mapping of ACCOUNTS to these assets is shown in the attached table (file ~~acctable.xls~~ ~~spreads.xls~~). To allow checking of the tables in SAS, variables need to be set up at an individual, benefit unit and household level which count the different types of account held. So, for example, if a person held adult NS capital bonds and index-linked certificates, NSBOCTI would equal 2. If their partner held some of the same, NSBOCTB would equal 4. If someone else in the household also held the same, NSBOCTH would equal 6.

Note the variables do not count the number of different accounts a person has of a certain type. This can only be gathered for those entering the assets block, questions in the accounts block relate to total interest from particular sources.

To check publication tables, an individual, benefit unit or household will hold a particular type of account if the relevant variable is greater than zero. For example, for National Savings Bonds if NSBOCTI (individual), NSBOCTB (benefit unit) or NSBOCTH (household) is greater than zero

## 2 FRS Specification

Code Condition

## Process ACCOUNTS table via PERSON (Adults and children)

Set CURACTI, POACCTI, TESSCTI, OTBSCTI, OTBNCTI, OTACCTI, GILTSCTI, UNTRCTI, STSHCTI, NSBOCTI, SAYECTI, PRBOCTI, **PEPSCTI**, CURACTB, POACICTB, TESSCTB, OTBSCTB, OTBNCTB, OTACCTB, GILTSCTB, UNTRCTB, STSHCTB, NSBOCTB, SAYECTB, PRBOCTB, **PEPSCTB**, CURACTH, POACICTH, TESSCTH, OTBSCTH, OTBNCTH, OTACCTH, GILTSCTH, UNTRCTH, STSHCTH, NSBOCTH, SAYECTH, PRBOCTH, **PEPSCTH to zero**

CURACTI	If ACCOUNT=1 then CURACTI=1
POACCTI	If ACCOUNT=2 then POACCTI=POACTTI+1 If ACCOUNT=3 then POACCTI=POACCTI+1 If ACCOUNT=24 then POACCTI=POACCTI+1 If ACCOUNT=25 then POACCTI=POACCTI+1
TESSCTI	If ACCOUNT=4 then TESSCTI=1
OTBSCTI	If ACCOUNT=5 then OTBSCTI=OTBSCTI+1 <del>If ACCOUNT=6 then OTBSCTI=OTBSCTI+1</del> If ACCOUNT=26 then OTBSCTI=OTBSCTI+1
OTBNCTI	If ACCOUNT= <del>7</del> <b>6</b> then OTBNCTI=OTBNCTI+1 <del>If ACCOUNT=8 then OTBNCTI=OTBNCTI+1</del> If ACCOUNT=27 then OTBNCTI=OTBNCTI+1
OTACCTI	If ACCOUNT= <del>9</del> <b>7</b> then OTACCTI=OTACCTI+1 <del>If ACCOUNT=10 then OTACCTI=OTACCTI+1</del>
GILTSCTI	If ACCOUNT= <del>44</del> <b>8</b> then GILTSCTI=GILTSCTI+1 <del>If ACCOUNT=12 then GILTSCTI=GILTSCTI+1</del> If ACCOUNT=28 then GILTSCTI=GILTSCTI+1
UNTRCTI	If ACCOUNT= <del>43</del> <b>9</b> then UNTRCTI=UNTRCTI+1 If ACCOUNT=29 then UNTRCTI=UNTRCTI+1
STSHCTI	If ACCOUNT= <del>44</del> <b>10</b> then STSHCTI=STSHCTI+1 <del>If ACCOUNT=15 then STSHCTI=STSHCTI+1</del> If ACCOUNT=30 then STSHCTI=STSHCTI+1
NSBOCTI	If ACCOUNT=16 then NSBOCTI=NSBOCTI+1 If ACCOUNT=17 then NSBOCTI=NSBOCTI+1 If ACCOUNT=18 then NSBOCTI=NSBOCTI+1 If ACCOUNT=19 then NSBOCTI=NSBOCTI+1 If ACCOUNT=22 then NSBOCTI=NSBOCTI+1 If ACCOUNT=23 then NSBOCTI=NSBOCTI+1 If ACCOUNT=31 then NSBOCTI=NSBOCTI+1



If ACCOUNT=32 then NSBOCTI=NSBOCTI+1  
If ACCOUNT=33 then NSBOCTI=NSBOCTI+1  
If ACCOUNT=34 then NSBOCTI=NSBOCTI+1  
If ACCOUNT=37 then NSBOCTI=NSBOCTI+1  
If ACCOUNT=38 then NSBOCTI=NSBOCTI+1

SAYECTI      If ACCOUNT=20 then SAYECTI=SAYECTI+1

PRBOCTI      If ACCOUNT=21 then PRBOCTI=PRBOCTI+1  
                 If ACCOUNT=36 then PRBOCTI=PRBOCTI+1

**PEPSCTI      If ACCOUNT=11 then PEPSCTI=PEPSCTI+1**

**For each benefit unit**

CURACTB equals all occurrences of CURACTI  
POACCTB equals all occurrences of POACCTI  
TESSCTB equals all occurrences of TESSCTI  
OTBSCTB equals all occurrences of OTBSCTI  
OTBNCTB equals all occurrences of OTBNCTI  
OTACCTB equals all occurrences of OTACCTI  
GILTSCTB equals all occurrences of GILTSCTI  
UNTRCTB equals all occurrences of UNTRCTI  
STSHCTB equals all occurrences of STSCCTI  
NSBOCTB equals all occurrences of NSBOCTI  
SAYECTB equals all occurrences of SAYECTI  
PRBOCTB equals all occurrences of PRBOCTI  
**PEPSCTB equals all occurrences of PEPSCTI**

**For each household**

CURACTH equals all occurrences of CURACTB  
POACCTH equals all occurrences of POACCTB  
TESSCTH equals all occurrences of TESSCTB  
OTBSCTH equals all occurrences of OTBSCTB  
OTBNCTH equals all occurrences of OTBNCTB  
OTACCTH equals all occurrences of OTACCTB  
GILTSCTH equals all occurrences of GILTSCTB  
UNTRCTH equals all occurrences of UNTRCTB  
STSHCTH equals all occurrences of STSCCTB  
NSBOCTH equals all occurrences of NSBOCTB  
SAYECTH equals all occurrences of SAYECTB  
PRBOCTH equals all occurrences of PRBOCTB  
**PEPSCTH equals all occurrences of PEPSCTB**

**AUTHTYPE**

Purpose : To indicate in which type of authority the household is situated.  
 Created : 17 March 1993  
 Database Table : HOUSEHOL  
 Minimum Value : 1  
 Maximum Value : 6  
 Units : Integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : HBM  
 Issue date : 21 May 2003  
 Amendments : VC - 22 April 1993 To make definition clearer.  
               : VC - 18 May 1993 To amend categories as initial information had shown :  
               : metropolitan districts as non-metropolitan districts and vice versa.  
               : VC - 15 June 1993 Wolverhampton moved into Metropolitan category  
               : previously shown as Non-metropolitan.  
               : VC - 9 February 1994 No version 30 change needed  
               : JS - 19 January 1996: coding for Met and Non-Met reversed to maintain consistency  
               : with HBMS data; New Towns coding removed (no cases included anyway)  
               : VE - 4 June 1996 - No initial amendments needed for V32

**1 Definition**

This variable is coded as

- 1 Non-metropolitan England
- 2 Metropolitan England
- 3 London
- 4 Wales
- 5 Scotland

- 1 Not applicable to this case
- 2 Unable to derive as variable LAC is missing

The authority type is derived from the Local Authority Code in the variable LAC which is a unique code only found in FRS. Each household will then be allocated an authority type according to the list provided by Ms Odwell. Authtype is a new variable produced only by FRS for the HBM.

NB - No New Towns are currently included in the specification as there is a problem with FRS codes for them.

**2 FRS Specification**

Code each authority type according to the attached list - **NB** the type of authority code is the fourth column in the list.

- 1 Not applicable to this case (shouldn't be any)
- 2 Unable to derive as the variable LAC is missing.

NB - These LA codes are unique to FRS.

**3 Results**

Tabulation to show the number of households in each authority type.

**4 Test Cases**

None

**FAMILY RESOURCES SURVEY**

**DERIVED VARIABLE SPECIFICATION**

STND REGION	LAC	AUTHORITY NAME		TYPE OF AUTHORITY
1	776	Gateshead	2	
1	867	Newcastle upon Tyne		2
1	868	North Tyneside	2	
1	869	South Tyneside	2	
1	870	Sunderland	2	
2	076	Barnsley	2	
2	167	Doncaster	2	
2	168	Rotherham	2	
2	169	Sheffield	2	
2	170	Bradford	2	
2	171	Calderdale	2	
2	172	Kirklees	2	
2	173	Leeds	2	
2	174	Wakefield	2	
8	462	Birmingham	2	
8	463	Coventry	2	
8	464	Dudley	2	
8	465	Sandwell	2	
8	466	Solihull	2	
8	557	Walsall	2	
8	558	Wolverhampton	2	
9	273	Bolton	2	
9	274	Bury	2	
9	275	Manchester	2	
9	276	Oldham	2	
9	367	Rochdale	2	
9	368	Salford	2	
9	369	Stockport	2	
9	370	Tameside	2	
9	371	Trafford	2	
9	372	Wigan	2	
9	567	Knowsley	2	
9	568	Liverpool	2	
9	569	St Helens	2	
9	570	Sefton	2	
9	571	Wirral	2	
<hr/>				
1	572	Allerdale	1	
1	573	Barrow-in-Furness		1
1	574	Carlisle	1	
1	575	Copeland	1	
1	576	Eden		1
1	667	South Lakeland	1	
1	668	Hartlepool		1
1	669	Langbaugh-on-Tees	Redcar and Cleveland	1
1	670	Middlesbrough	1	
1	671	Stockton-on-Tees	1	
1	672	Chester-le-Street	1	
1	673	Darlington	1	
1	674	Derwentside	1	

**FAMILY RESOURCES SURVEY**

**DERIVED VARIABLE SPECIFICATION**

---

1	675	Durham	1	
1	676	Easington	1	
1	767	Sedgefield	1	
1	768	Teesdale	1	
1	769	Wear Valley	1	
1	770	Alnwick	1	
1	771	Berwick-upon-Tweed	1	1
1	772	Blyth Valley	1	
1	773	Castle Morpeth	1	
1	774	Tynedale	1	
1	775	Wansbeck	1	
2	067	Scunthorpe	1	
2	068	Craven	1	
2	069	Hambleton	1	
2	070	Harrogate	1	
2	071	Richmondshire	1	
2	072	Ryedale	1	
2	073	Scarborough	1	
2	074	Selby	1	
2	075	York	1	
2	969	East Yorks Borough of Beverley	1	1
2	970	Boothferry	1	
2	971	Cleethorpes	1	
2	972	East Yorkshire	1	
2	973	Glanford	1	
2	974	Great Grimsby	1	
2	975	Holderness	1	
2	976	Kingston upon Hull	1	
3	559	Amber Valley	1	
3	560	Bolsover	1	
3	561	Chesterfield	1	
3	562	Derby	1	
3	563	Erewash	1	
3	564	High Peak	1	
3	565	North East Derbyshire	1	1
3	566	South Derbyshire	1	
3	657	Derbyshire Dales	1	
3	658	Blaby	1	
3	659	Charnwood	1	
3	660	Harborough	1	
3	661	Hinckley and Bosworth	1	1
3	662	Leicester	1	
3	663	Melton	1	
3	664	North West Leicestershire	1	1
3	665	Oadby and Wigston	1	1
3	666	Rutland	1	
3	757	Boston	1	
3	758	East Lindsey	1	
3	759	Lincoln	1	
3	760	North Kesteven	1	
3	761	South Holland	1	
3	762	South Kesteven	1	
3	763	West Lindsey	1	
3	764	Corby	1	
3	765	Daventry	1	
3	766	East Northamptonshire	1	1

---

**FAMILY RESOURCES SURVEY**

**DERIVED VARIABLE SPECIFICATION**

---

3	857	Kettering	1	
3	858	Northampton	1	
3	859	South Northamptonshire	1	1
3	860	Wellingborough	1	
3	861	Ashfield	1	
3	862	Bassetlaw	1	
3	863	Broxtowe	1	
3	864	Gedling	1	
3	865	Mansfield	1	
3	866	Newark and Sherwood	1	
3	967	Nottingham	1	1
3	968	Rushcliffe	1	
4	153	Cambridge	1	
4	154	East Cambridgeshire	1	1
4	155	Fenland	1	
4	156	Huntingdonshire	1	
4	247	Peterborough	1	
4	248	South Cambridgeshire	1	1
4	249	Breckland	1	
4	250	Broadland	1	
4	251	Great Yarmouth	1	
4	252	Norwich	1	
4	253	North Norfolk	1	
4	254	South Norfolk	1	
4	255	Kings Lynn and West Norfolk	1	1
4	256	Babergh	1	
4	347	Forest Heath	1	
4	348	Ipswich	1	
4	349	Mid Suffolk	1	
4	350	St Edmundsbury	1	
4	351	Suffolk Coastal	1	
4	352	Waveney	1	
6	047	Guildford	1	
6	048	Mole Valley	1	
6	049	Reigate and Banstead	1	1
6	050	Runnymede	1	
6	051	Spelthorne	1	
6	052	Surrey Heath	1	
6	053	Tandridge	1	
6	054	Waverley	1	
6	055	Woking	1	
6	056	Adur	1	
6	145	Luton	1	
6	146	Mid Bedfordshire	1	
6	147	Arun	1	
6	148	Chichester	1	
6	149	Crawley	1	
6	150	Horsham	1	
6	151	Mid Sussex	1	
6	152	Worthing	1	
6	237	North Bedfordshire	1	
6	238	South Bedfordshire	1	
6	239	Bracknell Forest	1	
6	240	Newbury	1	
6	241	Reading	1	
6	242	Slough	1	

---

6	243	Windsor and Maidenhead	1	
6	244	Wokingham	1	
6	245	Aylesbury Vale	1	
6	246	Chiltern	1	
6	337	Milton Keynes	1	
6	338	South Buckinghamshire		1
6	339	Wycombe	1	
6	340	Brighton	1	
6	341	Eastbourne	1	
6	342	Hastings	1	
6	343	Hove	1	
6	344	Lewes	1	
6	345	Rother	1	
6	346	Wealden	1	
6	437	Basildon	1	
6	438	Braintree	1	
6	439	Brentwood	1	
6	440	Castle Point	1	
6	441	Chelmsford	1	
6	442	Colchester	1	
6	443	Epping Forest	1	
6	444	Harlow	1	
6	445	Maldon	1	
6	446	Rochford	1	
6	537	Southend-on-Sea		1
6	538	Tendring	1	
6	539	Thurrock	1	
6	540	Uttlesford	1	
6	541	Basingstoke and Deane	1	
6	542	East Hampshire	1	
6	543	Eastleigh	1	
6	544	Fareham	1	
6	545	Gosport	1	
6	546	Hart	1	
6	637	Havant	1	
6	638	New Forest	1	
6	639	Portsmouth	1	
6	640	Rushmoor	1	
6	641	Southampton	1	
6	642	Test Valley	1	
6	643	Winchester	1	
6	644	Broxbourne	1	
6	645	Dacorum	1	
6	646	East Hertfordshire	1	
6	737	Hertsmere	1	
6	738	North Hertfordshire	1	
6	739	St Albans	1	
6	740	Stevenage	1	
6	741	Three Rivers	1	
6	742	Watford	1	
6	743	Welwyn Hatfield	1	
6	744	Medina	1	
6	745	South Wight	1	
6	746	Ashford	1	
6	837	Canterbury	1	
6	838	Dartford	1	

---

**FAMILY RESOURCES SURVEY**

**DERIVED VARIABLE SPECIFICATION**

---

6	839	Dover	1	
6	840	Gillingham	1	
6	841	Gravesham	1	
6	842	Maidstone	1	
6	843	Rochester upon Medway	1	
6	844	Sevenoaks	1	
6	845	Shepway	1	
6	846	Swale	1	
6	947	Thanet	1	
6	948	Tonbridge and Malling	1	1
6	949	Tunbridge Wells	1	
6	950	Cherwell	1	
6	951	Oxford	1	
6	952	South Oxfordshire	1	
6	953	Vale Of White Horse	1	1
6	954	West Oxfordshire	1	
6	955	Elmbridge	1	
6	956	Epsom and Ewell	1	
7	057	Cheltenham	1	
7	058	Cotswold	1	
7	059	Forest of Dean	1	
7	060	Gloucester	1	
7	061	Stroud	1	
7	062	Tewkesbury	1	
7	063	Mendip	1	
7	064	Sedgemoor	1	
7	065	Taunton Deane	1	
7	066	West Somerset	1	
7	158	Kennet	1	
7	159	North Wiltshire	1	
7	160	Salisbury	1	
7	161	Thamesdown	1	
7	162	West Wiltshire	1	
7	656	Bath	1	
7	747	Bristol	1	
7	748	Kingswood	1	
7	749	Northavon	1	
7	750	Wansdyke	1	
7	751	Woodspring	1	
7	752	Caradon	1	
7	753	Carrick	1	
7	754	Kerrier	1	
7	755	North Cornwall	1	
7	756	Penwith	1	
7	847	Restormel	1	
7	849	East Devon	1	
7	850	Exeter	1	
7	851	Mid Devon	1	
7	852	North Devon	1	
7	853	Plymouth	1	
7	854	South Hams	1	
7	855	Teignbridge	1	
7	856	Torbay	1	
7	957	Torridge	1	
7	958	West Devon	1	
7	959	Bournemouth	1	



**FAMILY RESOURCES SURVEY**

**DERIVED VARIABLE SPECIFICATION**

---

7	960	Christchurch	1	
7	961	North Dorset	1	
7	962	Poole	1	
7	963	Purbeck	1	
7	964	West Dorset	1	
7	965	Weymouth and Portland	1	
7	966	East Dorset (aka Wimborne)	1	1
8	157	South Somerset (aka Yeovil)	1	1
8	163	Bromsgrove	1	
8	164	Hereford	1	
8	165	Leominster	1	
8	166	Malvern Hills	1	
8	257	Redditch	1	
8	258	South Hertfordshire	1	
8	259	Worcester	1	
8	260	Wychavon	1	
8	261	Wyre Forest	1	
8	262	Bridgnorth	1	
8	263	North Shropshire	1	
8	264	Oswestry	1	
8	265	Shrewsbury and Atcham	1	
8	266	South Shropshire	1	
8	357	The Wrekin	1	
8	358	Cannock Chase	1	
8	359	East Staffordshire	1	
8	360	Lichfield	1	
8	361	Newcastle-under-Lyme	1	1
8	362	South Staffordshire	1	
8	363	Stafford	1	
8	364	Staffordshire Moorlands	1	1
8	365	Stoke-on-Trent	1	
8	366	Tamworth	1	
8	457	North Warwickshire	1	1
8	458	Nuneaton & Bedworth	1	1
8	459	Rugby	1	
8	460	Stratford-on-Avon	1	
8	461	Warwick	1	
9	175	Chester	1	
9	176	Congleton	1	
9	267	Crewe and Nantwich	1	1
9	268	Ellesmere Port and Neston	1	1
9	269	Halton	1	
9	270	Macclesfield	1	
9	271	Vale Royal	1	
9	272	Warrington	1	
9	373	Blackburn	1	
9	374	Blackpool	1	
9	375	Burnley	1	
9	376	Chorley	1	
9	467	Fylde	1	
9	468	Hyndburn	1	
9	469	Lancaster	1	
9	470	Pendle	1	
9	471	Preston	1	
9	472	Ribble Valley	1	
9	473	Rossendale	1	

**FAMILY RESOURCES SURVEY**

**DERIVED VARIABLE SPECIFICATION**

9	474	South Ribble		1
9	475	West Lancashire	1	
9	476	Wyre	1	

5	353	City of London	3	
5	354	Camden	3	
5	355	Hackney	3	
5	356	Hammersmith and Fulham	3	
5	447	Haringey	3	
5	448	Islington	3	
5	449	Kensington and Chelsea	3	
5	450	Lambeth	3	
5	451	Lewisham	3	
5	452	Newham	3	
5	453	Southwark	3	
5	454	Tower Hamlets	3	
5	455	Wandsworth	3	
5	456	Westminster, City of	3	
5	547	Barking and Dagenham	3	
5	548	Barnet	3	
5	549	Bexley	3	
5	550	Brent	3	
5	551	Bromley	3	
5	552	Croydon	3	
5	553	Ealing	3	
5	554	Enfield	3	
5	555	Greenwich	3	
5	556	Harrow	3	
5	647	Havering	3	
5	648	Hillingdon	3	
5	649	Hounslow	3	
5	650	Kingston upon Thames	3	
5	651	Merton	3	
5	652	Redbridge	3	
5	653	Richmond upon Thames	3	
5	654	Sutton	3	
5	655	Waltham Forest	3	

10	077	Torfaen	4	
10	078	Aberconwy	4	
10	079	Arfon	4	
10	080	Dwyfor	4	
10	081	Meirionydd	4	
10	082	Ynys Mon - Isle of Anglesey	4	4
10	083	Cynon Valley	4	
10	084	Merthyr Tydfil	4	
10	085	Ogwr	4	
10	086	Rhondda	4	
10	177	Rhymney Valley	4	
10	178	Taff-Ely	4	
10	179	Brecknock	4	

**FAMILY RESOURCES SURVEY**

**DERIVED VARIABLE SPECIFICATION**

10	180	Montgomeryshire		4
10	181	Radnorshire	4	
10	182	Cardiff	4	
10	183	Vale of Glamorgan		4
10	184	Port Talbot (aka Afan)		4
10	185	Lliw Valley	4	
10	186	Neath	4	
10	277	Swansea	4	
10	871	Alyn and Deeside		4
10	872	Colwyn	4	
10	873	Delyn	4	
10	874	Glyndwr	4	
10	875	Rhuddlan	4	
10	876	Wrexham Maelor		4
10	977	Carmarthen	4	
10	978	Ceredigion	4	
10	979	Dinefwr	4	
10	980	Llanelli	4	
10	981	Preseli	4	
10	982	South Pembrokeshire		4
10	983	Blaenau Gwent	4	
10	984	Islwyn	4	
10	985	Monmouth	4	
10	986	Newport	4	

11	278	Berwickshire	5	
11	279	Ettrick and Lauderdale		5
11	280	Roxburgh	5	
11	281	Tweeddale	5	
11	282	Clackmannan	5	
11	283	Falkirk	5	
11	284	Stirling	5	
11	285	Annandale and Eskdale	5	
11	286	Nithsdale	5	
11	377	Stewartry	5	
11	378	Wigtown	5	
11	379	Dunfermline	5	
11	380	Kirkcaldy	5	
11	381	North East Fife	5	
11	382	Aberdeen City	5	
11	383	Banff and Buchan		5
11	384	Gordon	5	
11	385	Kincardine and Deeside	5	
11	386	Moray	5	
11	477	Badenoch and Strathspey	5	
11	478	Caithness	5	
11	479	Inverness	5	
11	480	Lochaber	5	
11	481	Nairn	5	
11	482	Ross and Cromarty		5
11	483	Skye and Lochalsh		5
11	484	Sutherland	5	
11	485	East Lothian	5	
11	486	Edinburgh City	5	

11	577	Midlothian	5	
11	578	West Lothian	5	
11	579	Argyll and Bute	5	
11	580	Bearsden & Milngavie	5	5
11	581	Clydebank / Clydesdale	5	5
11	582	Cumbernauld and Kilsyth	5	
11	583	Cumnock and Doon Valley	5	5
11	584	Cunninghame	5	
11	585	Dumbarton	5	
11	586	East Kilbride	5	
11	677	Eastwood	5	
11	678	Glasgow City	5	
11	679	Hamilton	5	
11	680	Inverclyde	5	
11	681	Kilmarnock and Loudoun	5	
11	682	Kyle and Carrick	5	
11	683	Lanark (aka Clydesdale)	5	5
11	684	Monklands	5	
11	685	Motherwell	5	
11	686	Renfrew	5	
11	777	Strathkelvin	5	
11	778	Angus	5	
11	779	Dundee City	5	
11	780	Perth and Kinross	5	
11	781	Orkney Islands Area	5	5
11	782	Shetland Islands Area	5	5
11	783	Western Isles Islands Area	5	5

**BOARDER**

Purpose : To indicate the total weekly amount of rent paid by a boarder Benefit Unit.  
 Created : VC - 12 March 1993  
 Database Table : BENUNIT  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : **HBM**, ISM (Bold indicates lead user)  
 Amendments : VC 24 March 1993 - by Benefit Unit.  
               : VC - 9 February 1994 updated for version 30  
               : VC - 25 February 1994 To exclude any amounts using period codes 12 or :  
               13  
               : JS - 20 February 1996 to allow skipped values of CVPD (where CVPAY has  
               been imputed)  
               : VE - 23 May 1996 - No initial V32 update needed  
 Issue date : 6 June 1996

NB - This is a new variable produced by FRS and does not replace the FES in any way.

**1 Definition**

This variable is coded as

BOARDER The total weekly amount paid by a benefit unit classed as a boarder to the householder for a room and food.

-1 Not applicable to this case.

-2 Unable to derive due to missing values.

BOARDER is derived by benefit unit from the variable CONVBL (asks whether person is a boarder, lodger or neither of these). Where CONVBL = 1 which indicates that an adult in the benefit unit is a boarder the amount paid for board and lodge is to be found in CVPAY.

If CVPD (what period does this cover) = 12 or 13 (one off/lumpsum or any other) they become unable to derive as these amounts cannot be converted to a weekly amount.

Where CVPAY has been imputed, CVPD will be left as "skipped". The program needs to be amended to allow for these cases to be included

**2 FRS Specification**

For each BENUNIT record and for each adult in the benefit unit

Code            Condition

BOARDER From ADULT table

If CONVBL=1 get all variables.

If CONVBL = 1 (answered boarder), and CVPD equals -1 or 1-11 the amount of BOARDER is the amount in CVPAY (amount of rent paid).

If there is more than one adult in the benefit unit, the amount of BOARDER is the total amount paid from both adults.

-1 Not applicable to this case - where convbl = 2 or 3 (lodgers or others) or is skipped as the question has not been asked.

-2 Unable to derive due to value of convbl or cvpay missing or where cvpd = 12 or 13

### 3 Results

Tabulation is required to show the number of boarders paying an amount for board and lodge in weekly bands of

Under £25  
£25 - £50  
£50 - £75  
£75 - £100  
£100 - £125  
£125 - £150  
£150 or over

### 4 Test Cases

None as yet

**BUINC**

Purpose : To show the total amount of income received by each benefit unit for use in the FRS publication.  
 Created : 2 February 1996  
 Database Table : BENUNIT  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables : INDINC, CHINCDV  
     Children :  
     Parents :  
 Core variable/user : ASD4A  
 Amendments : VE - 23 May 1996 - No initial amendments needed for V32 update  
 Issue date : 21 May 2003

**1 Definition**

This variable is coded as

BUINC           The total amount of income received each week by all members of the benefit unit.  
 0                No income is received by the benefit unit  
 -2               Unable to derive due to missing values

This specification, also sets up the component DVs used in the publication. These are:

BUEARNS       earned income  
 BSEINC        self-employment income  
 BUINV         investment income  
 BURPINC       retirement pension plus any income support  
 BPENINC       other pension income  
 BUDISBEN      disability benefits  
 BUOTHBEN     other benefits  
 BURINC        remaining income

**2 FRS Specification**

For each benefit unit

<u>Code</u>	<u>Condition</u>
BUINC	From ADULT table get INEARNs, INSEINC, ININV, INRPINC, INPENINC, INOTHBEN, INRINC and INDINC for all adults in the benefit unit  From CHILD table get CHEARNs, CHINV, CHRINC and CHINDV for all children in the benefit unit

-2 BUINC equals total of all occurrences of INDINC and CHINDV  
where any of the components are missing

**Benefit unit level components are calculated as follows:**

BUEARNS equals total of all occurrences of INEARNs and CHEARNs

BSEINC equals total of all occurrences of INSEINC

BUINV equals total of all occurrences of ININV and CHINV

BURPINC equals total of all occurrences of INRPINC

BPENINC equals total of all occurrences of INPENINC

BUDISBEN equals total of all occurrences of INDISBEN

BUOTHBEN equals total of all occurrences of INOTHBEN

BURINC equals total of all occurrences of INRINC and CHRINC

these variables should also be set to -2 if any components are missing



**BURENT**

Purpose : To show the rent eligible for housing benefit paid by a benefit unit for accommodation. This is after taking off certain service charges but before the deduction of Housing Benefit. Note: unlike HHRENT, this variable includes rent paid by BOARDERS/LODGERS.

Created : 26 January 1993

Database Table: BENUNIT

Minimum Value : 0

Maximum Value:

Units : Real

Validations :

Related Variables : HHRENT

Children :

Parents :

Core variable/user : **HBM, TAKE-UP** (nb **bold** denotes lead user)

Issue date : 21 May 2003

Amendments : VC - 26 April 1993. To divide grsrent by benefit unit not by : household.  
 : VC - 18 May 1993. To change name from GRSRENT to BURENT  
 : VC - 9 February 1994 Version 30 amendment  
 : VC - 25 February 1994 To exclude any records where period code is : 12  
 or 13  
 : BS - 8 August 1995. Spec amended to include changes to the V31  
 questionnaire.  
 : JS - 20 February 1996 - to make amendments to allow skipped values  
 where amounts have been imputed  
 : VE - 24 April 1996 - to broaden spec to include cases where HHSTAT=2  
 and no HB is received  
 : VE - 5 June 1996 - No initial amendments needed for V32 update  
 : SG - 28 July 1997 - correction to multi benefit unit household calculations

This variable has been produced for the HBM and for TAKE-UP as they need to know the eligible rent of a benefit unit rather than by household level.

**1 Definition**

This variable is coded as

BURENT This is the rent eligible for housing benefit paid by a benefit unit for accommodation. This is after taking off certain service charges but before the deduction of Housing Benefit.

-1 Not applicable to this case - owner-occupiers (this includes shared owner/occupiers where HHSTAT = 2), rent-free and non-householders BU (but boarder/lodgers should be included).

-2 Unable to derive because of missing values.

This is the total amount of eligible rent paid by a benefit unit and is derived from three other derived variables -HHRENT, BOARDER and LODGER.

The BOARDER and LODGER variables provide the amount of (gross) rent a particular boarder/lodger benefit unit pays and should be attached to the relevant benefit unit. (See comments below on food/fuel included)

All tenure types are included when looking at variables to ensure cases where the head of household is an owner occupier but has boarders/lodgers are accounted for.

HHRENT (see separate spec) derives the total amount of eligible rent paid by a household. BURENT produces this on a benefit unit basis . Therefore, if HHSTAT = 1 (indicating a conventional household of single person, couple with other family members and/or boarders/lodgers) the total amount of eligible rent produced from HHRENT should be attached to the record of benefit unit no 1, as this is the main benefit unit responsible for paying the household rent. Boarder or lodger rents are separately identified through the BOARDER and LODGER derived variables. These should be attached to the relevant BU which will not be the 1st BU. We ought to deduct amount included for food/fuel, but as set amounts for these facilities vary so much need questions in future to identify if rent includes these and if so, what value these represent. [Check: Jo S For the moment need to add variable CHBAMT to compare with BOARDER/LODGER BURENT to make sure rent is not less than HB. ]

HHSTAT=2 cases:

Where HHSTAT = 2 (indicating that the household is shared on an equal basis and the head of household is unclear or arbitrary) the household rent needs further work to produce a proportional amount of rent for each benefit unit within the household. To do this proportions are derived and applied to HHRENT using RENT, SRENTAMT, HBENAMT (the latter for 1st BU where rent is declared after HB deducted).

NB There are a very small number of cases in 94/5 where SRENTAMT does not exist pre imputation and there will be NONE post imputation so the above will always be possible.

**2 FRS Specification**

For each household

<u>Code</u>	<u>Condition</u>
BURENT	From HOUSEHOL table, if TENURE = 1 2 or 3 or 4 or 5 (renting) get HHRENT and HHSTAT variables  From BENUNIT table get BOARDER and LODGER variables  If HHSTAT = 1 or -1, then  if BENUNIT=1, and not TENURE=1 BURENT = HHRENT if BENUNIT>1 and either BOARDER or LODGER exists BURENT for that benefit unit is the amount in BOARDER or LODGER. (Check if can use an HB receipt (CHBAMT) variable here for cross check. Eg in cases where B/L rent less than HB) (In other cases B/L rent may be greater than CHBAMT because rent includes food/fuel which is not eligible for HB. Could set rent to

HB here) For nonHB B/L rent may be greater than eligible rent but cant derive later.

If HHSTAT=2

if RENT (of 1ST BU) declared after HB deducted (HBENCHK=2) and RENTPD, HBENPD and SRENTPD<>12 or 13 then  
 $TOTRENT = (RENT + HBENAMT \text{ (for 1st BU)}) + (SRENTAMT + SCHBAMT \text{ (for all remaining BUs)})$

(for 1st BU) set BURENT =  $(RENT + HBENAMT) / TOTRENT * HHRENT$   
 (for other BUs) set BURENT =  $(SRENTAMT + SCHBAMT) / TOTRENT * HHRENT$

but if RENT (of 1ST BU) declared before HB deducted (HBENCHK=1) or no HB received by 1st BU (HBENAMT<=0) then  
 $TOTRENT = RENT + SRENTAMT + SCHBAMT$  (for all remaining BUs)

(for 1st BU) set BURENT =  $RENT / TOTRENT * HHRENT$   
 (for other BUs) set BURENT =  $(SRENTAMT + SCHBAMT) / TOTRENT * HHRENT$

Above specification assumes that all 12 and 13 period codes have been converted, additional checks need to be put in for period codes for RENT, HBENAMT and SRENTAMT

In shared households, information on service charges etc are only collected for the first benefit unit and not for second/third benefit units. Therefore, the sum of SRENTAMT and RENT/HBENAMT may not equal total HHRENT. To maintain consistency with HHRENT it is necessary to share out total rent pro-rata.

- 1 skipped (household contains no individuals paying rent)
- 2 unable to derive due to missing values or period codes 12 or 13

### 3 Results

Tabulation is required to show the total amount of eligible rent a household has to pay sorted into bands of the following weekly amounts.

Under £25  
 £25 - £50  
 £50 - £75  
 £75 - £100  
 £100 - £125  
 £125 - £150  
 £150 - £175  
 £175 - £200

£200 and over

**CHCARE, CHHOUR**

Purpose : To provide summary variables for child carers  
 Created : 29 April 1996  
 Database Table : CARE  
 Minimum Value :  
 Maximum Value :  
 Units : integer  
 Validations :  
 Related Variables : CAREAB, CAREAH, CARERE, CAREFR, CARECL, CAREOT, CARECB,  
 CARECH, HOURAB, HOURAH, HOURRE, HOURFR, HOURCL, HOUROT,  
 HOURCB, HOURCH(variables for adults caring)  
     Children :  
     Parents :  
 Core variable/user :  
 Amendments : VE - 4 June 1996 - No initial V32 update needed  
                   JS - 29 October 1996 - to reflect actual coding for V31  
 Issue date : 21 May 2003

**1 Definition**

The aim of these derived variables is to provide summary information on child carers within the household. For most users needs, information is only required on *who* is being cared for and *how much time* is spent caring. Any more detailed analysis of carers and those being cared for should be done using the SAS version of the care table. For adults, a detailed breakdown of whether those being cared for in the household are in the same or different benefit units. This is not possible for children because information is not collected at WHOLOOK for children individually. Summary variables therefore only give information on the total number of individuals ~~in the household~~ looked after by children and how much time they spend caring.

**CHCARE** Total number of individuals ~~within the household~~ cared for by child/children within the household

**CHHOUR** Total number of hours spent caring ~~for household members~~ by child/children in the household

0 **For all variables** - not applicable to this case - child/children in the household not look after anybody ~~in the household~~.

-2 **For all variables** - unable to derive due to missing values.

**2 FRS Specification**

Set all variables equal to zero.

From CARE record, for each person ~~in the household~~ needing care (~~NEEDPER=1-20~~), If WHOLOO11 (child/children doing caring)  
 CHCARE=CHCARE+1  
 CHHOUR=CHHOUR+HOUR(11)

-2 If any variables are missing

**CHINCDV**

Purpose : To show the total amount of income received by a child for use in the FRS publication (based on INCOMECH).  
 Created : 2 February 1996  
 Database Table : CHILD  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : ASD4A6E  
 Amendments : VE - 5 June 1996 - No initial amendments needed for V32 update  
 Issue date : 21 May 2003

**1 Definition**

This variable is coded as

CHINCDV The total amount of income received each week by a child. This variable includes earnings from any spare time jobs, income from trust funds, from savings accounts and other assets.

0 Child has no sources of income

-2 Unable to derive due to missing values

The variable CHINCDV is derived from the variables CHAMTERN, ACCOUNT, ACCINT (which produce the amount of income from a child's investment accounts), CHTST and CHAMTST (which produce the amount of income from trust funds). If a child possesses any or all of these incomes the variable is then the total amount received each week.

ACCINT and CHAMTTST are database variables created to hold the amounts collected in chint (block u-chint) and chamt (block s-chinc) respectively as these question names are duplicated.

The DV is also used to create components of gross income which are accumulated with adult versions to obtain BU and HH level variables. These are:

CHEARNS children's earnings (identical to CHAMTERN but set to zero if not applicable)  
 CHINV child's investment earnings  
 CHRINC remaining children's income

The variable has been extended to include any income from grants or scholarships. Any income from free school meals, prescriptions etc are counted in adult variables.

**2 FRS Specification**

For each child

<u>Code</u>	<u>Condition</u>
CHINCDV	<p>From CHILD table if CHAMTERN exists then get CHAMTERN</p> <p>set CHEARNS to zero If CHAMTERN exists and CHPDERN equal to -1 or 1-11 then CHEARNS=CHAMTERN</p> <p>From ACCOUNTS table</p> <p>set CHINV to zero</p> <p>If ACCOUNT = 24 get ACCINT and add to CHINV (NSB Post Office - ordinary) If ACCOUNT = 25 get ACCINT and add to CHINV (NSB Post Office - investment) If ACCOUNT = 26 get ACCINT and add to CHINV (Building Society) If ACCOUNT = 27 get ACCINT and add to CHINV (Bank account) If ACCOUNT = 28 get ACCINT and add to CHINV (Gilts) If ACCOUNT = 29 get ACCINT and add to CHINV (Unit Trusts) If ACCOUNT = 30 get ACCINT and add to CHINV (Other stocks/shares)</p> <p>From CHILD table</p> <p><b>set CHRINC to zero</b></p> <p>If CHTST = 1 and CHPDTST equal to -1 or 1-11 (child has an income from a trust fund) get CHAMTTST and add to CHRINC</p> <p>If TOTGNTCH exists then add TOTGNTCH/52 to CHRINC</p> <p>CHINCDV=CHEARNS+CHINV+CHRINC</p>
-2	Unable to derive if any components are missing



## CWATAMTD

Purpose: To show total amount of council water charge paid by Scottish households  
Database Table : HOUSEHOL  
Lead User : ASD4A  
Related variables:  
    Children: HHRENT  
    Parents: STDREGN  
Amendments : VE - 10 June 1996 - Amended for V32. COMMINC is no longer asked and WHYNOCT, a new question has been incorporated  
              : JS - 13 November 1996, to delete cwatpd (figures only collected from annual statement and converted to weekly amounts as part of the conversion process)  
              : VE - 20 November 1996 - to simplify specification  
              : VE - 4 February 1996 - To bring into line with HBAI by using a lower Council Tax Band if a lower valuation band has been used  
Issue date: : 21 May 2003

### 1 Definition

CWATAMTD The total weekly amount of council water charge paid by Scottish Households

-1 Not applicable to this case

-2 Unable to derive due to missing values

This variable is set up using CWATAMT for cases where council water charge is paid as part of rent and a look up table (used for the 1994/95 FES data supplied from the Scottish Office) for cases where council water charge is paid separately.

Additional checks are made in the program to pick up the cases which are not asked directly about council tax included in rent. Those who are receiving a combination of discounts or rebates still have a value imputed for CWATAMTD because water charges are not included in CTB. For more information, see minute dated 14/2/95 (file vsctw).

### 2 FRS Specification

For each Scottish household (STDREGN=11) from HOUSEHOL table

Code                      Condition

CWATAMTD From RENTER HOUSEHOL table

If ~~COMMINC equals 1 and CWATAMT exists and CWATPD equals -1 or 1-11~~  
then CWATAMTD equals CWATAMT

Else if ~~COMMINC equals 2 (rent does not include council tax)~~  
then apply look up table to calculate CWATAMTD

Else if ~~RENT=0 (COMMINC is not asked in these cases)~~  
and ~~CTBAND=0 (not valued separately) or CTEXREB=1 2 (respondent has a formal exemption from the council tax) or (CTEXREB=3 and WHYNOCT=4) (neither of these - these cases are not covered by other questions so would otherwise be set to unable to derive)~~ set CWATAMTD to zero

Else if ~~CTAMT>0 (paying council tax) or CTEXREB=2 1 (combinations of discounts) or CTEXREB=4 (bill not yet issued) CTEXREB=3 and WHYNOCT=1, 2 or 3 (CT not paid because bill not yet received and HH not previously liable, bill not yet paid and HH not previously liable or deliberate non-payment)~~ then apply look up table to calculate CWATAMTD

~~Else if CTBAND=0 (not valued separately) or CTEXREB=1 2 (respondent has a formal exemption from the council tax) or CTEXREB=3 and WHYNOCT=4 (neither of these - these cases are not covered by other questions so would otherwise be set to unable to derive) set CWATAMTD to zero (repeat for cases where RENT was missing at interview and following questions have therefore been skipped)  
Else if CTAMT>0 (paying council tax) or CTEXREB=2 1 (combinations of discounts) or CTEXREB=4 (bill not yet issued) **CTEXREB=3 and WHYNOCT=1, 2 or 3 (CT not paid because bill not yet received and HH not previously liable, bill not yet paid and HH not previously liable or deliberate non-payment)** then apply look up table to calculate CWATAMTD (repeat for cases where RENT was missing at interview and following questions have therefore been skipped)~~

~~Else if ADDINF table, TENURE equals 1 (not renting)  
then apply look up table.~~

#### **Set CTBANDA**

**If CTBAND>0 then CTBANDA=CTBAND  
If (CTLVBAND=1 and CTLVCHK=2 and CTBAND>1) then  
CTBANDA=CTBAND-1**

Else apply look up table (final mop-up of any missing CWATAMTD). to calculate CWATAMTD from LAC and CTBANDA

- 1 Not applicable (see above)
- 2 Unable to derive due to missing values (this should not occur: COMMINC contains no missing values, all cases where COMMINC equals 2 are updated using a look up table, even where TENURE was missing and has been set to owner occupier, values updated using this method)

**DEDUCTS**

Purpose : To show the total amount of deductions from pay all employment, other than :  
Income Tax, National Insurance contributions and superannuation/pension :  
deductions.

Created : 20 January 1993

Database Table : ADULT

Minimum Value : 0

Maximum Value :

Units : Real

Validations :

Related Variables :

Children :

Parents :

Core variable/user : ~~ASD6A~~ ~~ASD4A~~ ~~ASD6E~~

Issue date : 21 May 2003

Amendments : VC - 11 February 1993 change to multi responses  
: VC - 8 March 1993 changed to show for main employment only  
: VC - 23 April 1993 to expand definition to show meaning of :  
questions/database variables.  
: VC - 22 July 1993 Coding highlighted a problem with the charity section as :  
the variable AMTTAXF is not asked after CHARITY = 1 but after CHRTAXF : =  
1 indicating that the deduction has been made under the tax free scheme.  
: VC 11 February 1994 Changes to reflect version 30 changes  
: VC - 25 February 1994 To exclude any deductions which would be converted  
to a weekly amount using the pay period  
: JS - 20 February 1996 to allow for skipped values where variables have been  
imputed  
: VE - 24 May 1996 - Initial amendments for V32 - New category of  
OTHDED (OTHDED=7 - repayment of loan from employer)  
: VE - 6 November 1996 - Charity part of spec made clearer  
: VE - 27 November 1996 - To amend spec so that if OTHDED =1 but  
DEDUC=-1 (due to editing/imputation), DEDUCTS is not set to -2

NB - This does not include any deductions for superannuation, pension schemes or additional voluntary contributions as these are dealt with in a separate variable called SUPERAN.

**1 Definition**

This variable is coded as

DEDUCTS The total amount of any deductions from pay from all employment excluding Income Tax, National Insurance Contributions and deductions for superannuation/pension schemes or AVCs.

-1 Not applicable to this case

-2 Unable to derive variable

DEDUCTS is derived from the variables AMTTAXF, AMTOTH, OTHDED3 - OTHDED67, DEDUC3 - DEDUC56 and DEDOTH. AMTTAXF and AMTOTH deal with deductions paid to charities and

OTHDED3 - OTHDED67 indicate whether payments are made to Trade Unions, Friendly Societies Sports and Social clubs, **to repay a loan from employer** or any others not previously mentioned . DEDUC3 - DEDUC56 and DEDOTH store the amounts paid.

OTHDED3 - OTHDED67 are database variables produced from OTHDED, which when = 1 indicate whether that person has a particular deduction. DEDUC3 - DEDUC56 and DEDOTH hold the amount for each deduction respectively and are also database variables produced from DEDUC and DEDOTH.

NB - For information OTHDED1 = 1 would represent pension/superannuation deductions and OTHDED2 = 1 represents AVCs.

The original version of the spec used the question/variable CHARITY. However, a problem emerged during coding which showed that the amount held in AMTTAXF was dependent on another variable - CHRTAXF which asks additionally if the deduction is through the tax free scheme and the amount collected refers to these deductions only. If analysts need information about all charitable deductions the questionnaire will have to be changed.

If PAYPD =12 or 13 the record must be rejected as unable to derive. This variable is used by the database conversion programme to convert the amount of deduction into a weekly rate. 12 or 13 refer to one-off/lumpsum deductions or any other period. If the variables cannot be converted the record must be rejected.

Where variables PAYAMT, CHRTAXF and OTHDED1-6 have been imputed, questions which follow will have been skipped and as a result DEDUCTS will be set to "unable to derive". To overcome this problem, the coding has been altered to:

- i allow skipped PAYPD
- ii impute AMTTAXF to over all mean for 1994/95 ~~(this will need to be updated annually)~~  
**CHRMEAN in constants table (h:\asd4a\frs\dvars\docs\spreads.xls)**
- iii assume where PAYAMT has been imputed (and CHRTAXF and OTHDED1-6 skipped) no deductions are made

## 2 FRS Specification

For each ADULT

<u>Code</u>	<u>Condition</u>
DEDUCTS	<p>From JOB table, for all jobs a person has</p> <p>If PAYPD equals -1 or 1 to 11 do the following -</p> <p><del>If CHRTAXF = 1 (any deductions made from wage for charities under tax free scheme? DEDUCTS=AMTTAXF (how much deducted) else if AMTTAXF=skipped DEDUCTS=DEDUCTS+1.27 and if CHROTH = 1 (any other charitable deductions) get AMTOTH (amount of any other deduction)</del>  <del>else if CHRTAXF= 1 DEDUCTS=DEDUCTS (ie no change)</del></p> <p><b>If CHARITY = 1 (deductions for charities) and CHRTAXF = 1 (deductions made under tax free scheme) and AMTAXF exists then DEDUCTS = AMTAXF (amount deducted)</b></p>

Else if CHARITY = 1 (deductions for charities) and AMTAXF = -1 then DEDUCTS = DEDUCTS + CHRMEAN (from constants table)

If CHROTH = 1 (other deductions for charities) then DEDUCTS = DEDUCTS + AMTOTH

If OTHDED3 = 1 and DEDUC3 >= 0, DEDUCTS = DEDUCTS + DEDUC3 (amount of union fees) else if OTHDED3 = -1 DEDUCTS = DEDUCTS

If OTHDED4 = 1 and DEDUC4 >= 0, DEDUCTS = DEDUCTS + DEDUC4 (amount for friendly societies) else if OTHDED4 = -1 DEDUCTS = DEDUCTS

If OTHDED5 = 1 and DEDUC5 >= 0, DEDUCTS = DEDUCTS + DEDUC5 (amount for sports clubs or specialised pastimes) else if OTHDED5 = -1 DEDUCTS = DEDUCTS

If OTHDED6 = 1 and DEDUC6 >= 0, DEDUCTS = DEDUCTS + DEDUC6 (amount of any other deductions not included above excluding tax/NI amount for repayment of a loan from employer) else if OTHDED6 = -2 -1 DEDUCTS = DEDUCTS

If OTHDED7 = 1 and DEDOTH >= 0, DEDUCTS = DEDUCTS + DEDOTH (amount of any other deductions not included above) else if OTHDED7 = -2 DEDUCTS = DEDUCTS

- 1 Not applicable to this case.
- 2 Unable to derive as any of the above variables are missing or PAYPD = 12 or 13 (one-off/lumpsum or other)

### 3 Results

Tabulation is required to show the number of adults with deductions from pay by the weekly amount of deduction sorted into bands of, for example

Under £5.00  
 £5.00 - £10.00  
 £10.00 - £15.00  
 £15.00 - £20.00  
 £20.00 - £25.00  
 Over £25.00

**DEPDEDS**

Purpose : To indicate the class of non-dependency applicable to each benefit unit.  
 Created : 14 January 1993  
 Database Table : BENUNIT  
 Minimum Value : 1  
 Maximum Value : 9  
 Units : Integer  
 Validations :  
 Related Variables :  
 Children :  
 Parents : TOTHOURS  
 Core variable/user : ISM HBM PSM  
 Amendments : VC - 28 January 1993,  
 : VC - 3 February 1993 (NB2), VC - 9 February 1993 : change to multi :  
 response.  
 : VC - 8 March 1993, To clarify which tables certain variables come from.  
 : VC - 16 March 1993, see Ms Odwell 's minute of 10 March 1993 for :  
 comments and changes to categories.  
 : VC - 24 March 1993. To add new categories of boarder and lodger and to :  
 remove head of household benefit unit and whether blind/in receipt of AA/DLA  
 : VC - 25 June 1993. To split category 5 into 18 - 24 receiving IS and 25 and :  
 over and receiving IS.  
 : VC - 11 February 1994 Amended to reflect changes to version 30  
 : JS - 20 February 1996: amended to reflect changes to benefits for V31  
 : VE - 4 June 1996 - Initial V32 changes - to amend TRAIN category for YTS  
 Issue date : 21 May 2003

NB This specification has been changed from the original FES specification as the way in which non-dependent deductions are assessed is now done on a different basis. Mr Shiels of ASD4D provided the new categories after consultation with the relevant policy section.

NB2 The specification was amended as Ms Odwell suggested an easier grouping with the models themselves dealing with the earnings bands so that the FRS specification does not have to be changed each year.

NB Boarders/lodgers are always non-family members.

**1 Definition**

This variable is coded as follows, if anybody in the benefit unit meets one of the criteria. If more than one of the criteria are met, the higher code takes precedence (ie 1 is highest). Boarders or lodgers are to be coded first as they are separate conditions and only any remaining non-dependents in the household should be coded as 3 - 9.

- 1 Boarder
- 2 Lodger
- 3 Aged 18 or over and working more than 16 hours a week
- 4 Aged 18 or over and on YTS
- 5 Aged 18 to 24 and in receipt of Income Support
- 6 Aged 25 and over and in receipt of Income Support

- 7 Students  
 8 Any others aged over 18  
 9 Aged 16 - 17
- 1 Not applicable to this case  
 -2 Unable to derive

This variable indicates to which class of non-dependency a benefit unit belongs. Depdededs will be derived from several variables from the BENUNIT, ADULT and BENEFITS tables.

## 2 FRS Specification

For each BENEFIT UNIT from BENUNIT record where BENUNIT > 1 (indicating that the BU is not the head of household BU) and for each adult in the benefit unit.

### Code   Condition

- 1 From ADULT table, if R01 = 17 (indicating that the relationship to person number 1 (head of household) is a non family member) and CONVBL = 1 (asked if boarder, lodger or other - 1 indicates a boarder).
- 2 From ADULT table, if R01 = 17 and CONVBL = 2 (as above but indicating a lodger).
- 3 From ADULT table, if AGE >= 18, WORKING = 1 (currently working) and TOTHOOURS >= 16 (TOTHOOURS is a derived variable indicating the total number of hours worked by an adult) or  
 If AGE >= 18, WORKING = 2 (not currently working), JOBAWAY = 1 (away from normal job) and TOTHOOURS >= 16.
- 4 From ADULT table, if AGE >= 18 and TRAIN = 21 (TRAIN asks if adult was on any form of govt training scheme, 21 = Youth Training).
- 5 From ADULT table, if AGE >= 18 and <= 24 and from BENEFITS table, if BENEFIT = 19
- 6 From ADULT table, if AGE >= 25 and from BENEFITS table, if BENEFIT = 19
- 7 From ADULT table, if AGE >= 18, FTED = 1 (in full-time education) and TYPEED = 7 (indicates university, polytechnic or any other college of higher education).
- 8 From ADULT table, if AGE >= 18 and not previously coded.
- 9 From ADULT table, if AGE = 16 or 17 (NB - this will pick out non-dependent adults aged 16 or 17, dependants who are aged 16 or 17 will be in the child table and are dealt with separately)
- 1 Not applicable to this case  
 -2 Unable to derive due to missing values.

## 3 Results

Tabulation to show total number of non-dependent benefit units and numbers falling into each category

**4 Test Cases**

- A Household with only head of household benefit unit
- B Household with one non-dependent, registered blind
- C Household with one non-dependent, aged 16 and not on YTS
- D Household with one non-dependent, aged 16 on YTS
- E Household with one non-dependent, aged 19, full-time student
- F Household with one non-dependent, aged 19 receiving Income Support
- G Household with one non-dependent, aged 20 earning £75 per week
- H Household with one non-dependent, aged 20 earning £105 per week
- I Household with one non-dependent, aged 20 earning £150 per week
- J Household with one non-dependent, aged 30 receiving IS
- K Household with one non-dependent, aged 30 not on IS



**DISINDHB**

Purpose : To indicate whether one or both adults in a benefit unit are blind or disabled.  
 Created : 13 January 1993  
 Database Table : BENUNIT  
 Minimum Value : 0  
 Maximum Value : 6  
 Units : Integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : HBM **ISM** PSM OTHERS (Bold indicates lead user)  
 Amendments : VC - 28 January 1993, VC - 9 February 1993 change to multi response.  
               : VC - Up to date benefit code received change to AA and DLA codes.  
               : VC - 29 March 1993, simplified FRS specification inserted to make coding :  
               easier.  
               : VC - 23 April 1993 To expand definition to show meaning of :  
               questions/database variables.  
               : VC - 25 June 1993 To change spec to include all benefit units and :  
               increase number of categories to show 1 blind, 2 blind, 1 disabled, 2 disabled :  
               or neither.  
               : VC - 11 February 1994 Amended to reflect version 30 changes  
               : JS - 20 February 1996 amended to reflect v31 changes  
               : VE - 13 May 1996 Amended to clarify the situation when an individual  
               is both blind and disabled  
               : VE - 4 June 1996 - Amended to reflect initial V32 changes  
 Issue date : 21 May 2003

**1 Definition**

This variable is coded as

- 1 1 person in benefit unit blind.
- 2 2 people in benefit unit blind.
- 3 1 person in benefit unit disabled.
- 4 2 people in benefit unit disabled.
- 5 1 blind person and 1 disabled person in benefit unit.
- 6 No person in benefit unit blind or disabled.
- 2 Unable to derive variable due to missing values

The variables used to produce DISIND are to be found in the ADULT table and are produced for all benefit units.

The first category is fulfilled if only one person in a benefit unit is registered blind (SPCREG1 = 1) and any other member of the same benefit unit is neither blind nor disabled. SPCREG1 is a database variable which is created from the question SPCREG and indicates that the person is registered blind. However, if there are two members of the benefit unit who are blind category 2 is appropriate, in this case SPCREG1 = 1 applies to both adults.

Categories 3 and 4 are used in a similar way if there are one or two members of the benefit unit classed as disabled. This classification is fulfilled if a person is receiving the care component of Disability Living Allowance (BEN1Q1 = 1), BEN2Q01=1 receiving Attendance Allowance (BEN2Q10 = 1) BEN2Q03=1 or where Attendance Allowance has been awarded AA to start at a later date (FUTATT = 1) B2QFUT3=1.

The fifth category is used where there are two members of the benefit unit and one is blind and the other is disabled.

The sixth category is where no adult in that benefit unit fulfils any of the above categories.

**An adult who appears to be classed as both blind and disabled is classified as disabled. This gives rise to the following coding system:**

- A**      **Neither blind nor disabled**
- B**      **Disabled**
- C**      **Blind**
- D**      **Both blind and disabled**

Person 1	A	B	C	D
Person 2				
A	6	3	1	3
B	3	4	5	4
C	1	5	2	5
D	3	4	5	4

**2 FRS Specification**

For the each BENUNIT record in each household

From ADULT table for each adult in the benefit unit

Count total number of adults in benefit unit where SPCREG1 = 1 = blind (temporary variable used for DISIND only).

Count total number of adults in benefit unit where BEN1Q1 = 1 BEN2Q01=1 or BEN2Q10 = 1 BEN2Q03=1 or FUTATT B2QFUT3 = 1 = dis (temporary variable)

(preset temporary variables to 0)

<u>Code</u>	<u>Condition</u>
1	If blind = 1 and dis = 0
2	If blind = 2 and dis = 0
3	If blind = 0 and dis = 1
4	If blind = 0 and dis = 2
5	If blind = 1 and dis = 1
6	Any other benefit unit not previously coded (where blind = 0 and dis = 0)
-2	Unable to derive because any of the above variables have missing values.

### **3 Results**

Tabulation needed to show number of benefit units falling into each category.

### **4 Test Cases**

To be added at a later date.

**ECOTYPBU**

Purpose	: To derive an HBAI type economic status indicator for each benefit unit.
Created	:
Database Table	: BENUNIT, ADULT
Minimum Value	: 1
Maximum Value	: 8
Units	: Integer
Validations	:
Related Variables	:
Children	:
Parents :	:
Core variable/user	: ASD6A <b>HBAI</b> ASD4A ( <b>Bold</b> denotes lead user)
Amendments	: VE - 21 May 1996 - Amended for initial Version 32 changes and to change full-time hours cut-off to greater than or equal to 30 (not greater than 29)
	: VE - 20 June 1996 - To change lead user to ASD6A
Issued	: 6 June 1996

**1 Definition**

This variable is coded by first creating an economic status variable for each adult and then from this deriving a variable for each benefit unit. As far as possible the variable has been coded to be consistent with the FES economic status variable detailed in FES Appendix 46.

The individual economic status variable EMPSTATC is coded as follows (EMPSTATC should also be added to the data base and flat file):

- 1 = FT SE
- 2 = FT Emp
- 3 = PT Emp or PT SE
- 4 = Unemployed
- 5 = Not working for any other reason.

EMPSTATC is derived from several variables from the adult and job records. Part time work is defined as employment or self employment for less than 30 hours.

An individual is self-employed if they record they are self-employed EMPSTAT=2 and work 30 hours or more in their main job (QHRSELF >= **29 30**) and if they are not on a Govt Training Prog (these should all be set as unemployed, except if they're on Enterprise Allowance **Business Start-up schemes**).

An individual is employed if they record that they are an employee (EMPSTAT=1), and are either not away from work or away from work by on full pay (this is set negatively by ABSPAY <> 2,3 - ie not away from work on reduced pay) and they work 30 or more hours a week in their main job (QHRS+EMPOVT >= **29 30**) and they are not on a Govt Training Scheme.

Part Time work, either employed or self employed is set as for the above two conditions but with the hours condition set at less than 30 hours.

EMPSTATC is set as unemployed by the following conditions:

where an individual is in receipt of UB (BEN3Q1=1);  
 or where an individual has looked for work in the last four weeks and can start work in the next two (LOOK4=1 and START=1);  
 or where an individual has looked for a Govt Training scheme in the last 4 weeks and can start work in the next two and is under 60 (LKYT4=1 and AGE<60 START=1); or  
 where an individual is waiting to start a job, either WAIT=1 or JOBAWAY=3  
 or where an individual is on a Govt Training Scheme, except ~~Enterprise Allowance~~ **Business Start-up schemes.**

Anyone not coded as the above is coded as not working for any other reason.

ECOTYPBU is coded as follows:

- 1 = Self Employed
- 2 = Single or Couple all in full-time work
- 3 = Couple, one in ft work, one in pt work
- 4 = Couple, one in ft work, one not working
- 5 = One or more in pt work
- 6 = Head or Spouse aged 60+
- 7 = Head or Spouse Unemployed
- 8 = Other

The coding for this is quite straightforward once EMPSTATC is coded.

## 2 FRS Specification

EMPSTATC

For each adult get all values from adult table:

Tests are applied sequentially with the coding determined by the last test to be satisfied:

Code

- 5 All cases first coded to not working for any other reason and then overwritten when one of the conditions apply
- 4 (from benefit record BEN3Q1=1) or  
(LOOK4=1 and START=1) or  
(LKYT4=1 and AGE<60 and START=1) or  
WAIT=1 or  
JOBAWAY=3 or  
or TRAIN = 1,2,4,5,6 )
- 3 (TRAIN <> 1,2,4,5,6) and (EMPSTAT=1 or EMPSTAT=2) and  
(ABSPAY<>2,3 and, from first job record, QHRS+EMPOVT<30) or (from first job record  
QHRSELF<30)
- 2 (TRAIN <> 1,2,4,5,6) and EMPSTAT=1 and ABSPAY<>2,3 and, from first job record,  
QHRS+EMPOVT>= ~~29~~ **30**

1 (TRAIN <> 1,2,4,5,6) and EMPSTAT =2 and, from first job record, QHRSELF>= 29 30

ECOTYPBU

For each Benefit Unit:

Tests are applied sequentially with the coding determined by the last test to be satisfied:

Code

- 8 All cases first coded to other and then overwritten when one of the conditions apply
- 7 EMPSTATC=4 for either the first or the second adult (if there is one)
- 6 AGE>59 for either the first or second adult (if one exists).
- 5 EMPSTATC=3 for either first or second adult (if one exists).
- 4 EMPSTATC=2 for first adult and EMPSTATC=4 or 5 for second adult, or vice versa.
- 3 EMPSTATC=2 for first adult and EMPSTATC=3 for second adult, or vice versa.
- 2 EMPSTATC=2 for first adult and EMPSTATC=2 for second adult (if one exists)
- 1 EMPSTATC=1 for either first or second adult (if one exists).

**ECSTATBU**

Purpose : To derive an HBAI consistent variable for economic status for use in the publication.

Created : 8 February 1996

Database Table : BENUNIT

Minimum Value : 1

Maximum Value : 9

Units : Integer

Validations :

Related Variables :

    Children :

    Parents :

Core variable/user : Publication

Amendments : VE - 21 May 1996 - Amended for initial Version 32 changes and to change full-time hours cut-off to greater than or equal to 30 (not greater than 29).

Issued : 21 May 2003

**1 Definition**

ECSTATBU is an extended version (ie including disability category) of the current HBAI economic status breakdown. The additional category for sick and disabled is calculated on the basis of answers to questions HEALTH, HPROB, RSTRCT, LAREG and JCREG.

The variable uses the existing coding for ECOTYPBU, but includes an initial test for disability which is then overwritten with the existing codes based on EMPSTATC tests.

ECOTYPBU is coded by first creating an economic status variable for each adult and then from this deriving a variable for each benefit unit. As far as possible the variable has been coded to be consistent with the FES economic status variable detailed in FES Appendix 46.

The individual economic status variable EMPSTATC is coded as follows. EMPSTATC should also be added to the flat file:

1 = FT SE  
 2 = FT Emp  
 3 = PT Emp or PT SE  
 4 = Unemployed  
 5 = Not working for any other reason.

EMPSTATC is derived from several variables from the adult and job records. Part time work is defined as employment or self employment for less than 30 hours.

An individual is self-employed if they record they are self-employed EMPSTAT=2 and work 30 hours or more in their main job (QHRSELF >= 29 30) and if they are not on a Govt Training Prog (these should all be set as unemployed, except if they're on Enterprise Allowance **Business Start-up schemes**).

An individual is employed if they record that they are an employee (EMPSTAT=1), and are either not away from work or away from work by on full pay (this is set negatively by ABSPAY <> 2,3 - ie not away

from work on reduced pay) and they work 30 or more hours a week in their main job (QHRS+EMPOVT>= 29 30) and they are not on a Govt Training Scheme.

Part Time work, either employed or self employed is set as for the above two conditions but with the hours condition set at less than 30 hours.

EMPSTATC is set as unemployed by the following conditions:

where an individual is in receipt of UB (BEN3Q1=1);  
 or where an individual has looked for work in the last four weeks and can start work in the next two (LOOK4=1 and START=1);  
 or where an individual has looked for a Govt Training scheme in the last 4 weeks and can start work in the next two and is under 60 (Lkyt4=1 and AGE<60 START=1); or  
 where an individual is waiting to start a job, either WAIT=1 or JOBAWAY=3  
 or where an individual is on a Govt Training Scheme, except Enterprise Allowance **Business Start-up schemes.**

Anyone not coded as the above is coded as not working for any other reason.

ECSTATBU is coded as:

- 1 Self employed (benefit units where at least one adult usually works self-employed for 30 or more hours a week)
- 2 Single or couple, all in full time work *as an employee* (30 or more hours a week)
- 3 Couple, one in full time work, one in part-time work (*including part-time self employed*)
- 4 Couple, one in full time work (*as an employee*), one not working
- 5 Single or couple, one or more in part-time work (*self-employed or employee*)
- 6 Single or couple, head or spouse aged 60 or over
- 7 Single or couple, head or spouse unemployed
- 8 Single or couple, head or spouse sick or disabled (under pension age)
- 9 Others

Note: The pensioner category **is not** consistent with the HBAI family status variable or HHCOMP, using a 60+ cut off for pensioners. Moreover, working pensioners will be classified as self employed/working full or part time **before** they are classified as pensioners. The disability category may also be different from HHCOMP (although the selection criteria are the same) because of the hierarchical classification.

The full-time/part-time split for ECSTATBU is also different from the EMPSTATB classification which uses the 16 hour rule. Total hours worked are based on QHRSELF (hours worked as self-employed - only asked of first job) and total of QHRS+EMVOVT (contracted hours plus usual overtime) for each job as an employee. This is different from EMPSTATB (although in practice, figures will be very similar) which uses the TOTHOURS DV to distinguish full/part time work. TOTHOURS looks at hours worked from **all** jobs, ie including any hours worked as employees at second and third jobs with self-employed.



**2 FRS Specification**

EMPSTATC is coded in the same way as ECOTYPBU

ECSTATBU then has an additional check for disability which is included as the first test.

For each Benefit Unit:

Tests are applied sequentially with the coding determined by the last test to be satisfied:

Code

- 9 All cases first coded to other and then overwritten when one of the conditions apply
- 8 If (((SEX=1 and AGE<65) or (SEX=2 and AGE<60)) and ((HEALTH=1 and HPROB=1) or (RSTRCT=1 or RSTRCT=2) or (LAREG=1) or (JCREG=1))) for either the first adult or the second adult (if there is one) in the benefit unit
- 7 EMPSTATC=4 for either the first or the second adult (if there is one)
- 6 AGE>59 for either the first or second adult (if one exists).
- 5 EMPSTATC=3 for either first or second adult (if one exists).
- 4 EMPSTATC=2 for first adult and EMPSTATC=4 or 5 for second adult, or vice versa.
- 3 EMPSTATC=2 for first adult and EMPSTATC=3 for second adult, or vice versa.
- 2 EMPSTATC=2 for first adult and EMPSTATC=2 for second adult (if one exists)
- 1 EMPSTATC=1 for either first or second adult (if one exists).

**EMPISM (IN259)**

Purpose : To give FES equivalent employment status for use in the ISM  
 Created : May 1995 as part of parallel run exercise  
 Database Table : ADULT  
 Minimum Value : 0  
 Maximum Value : 8  
 Units : Integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents : EMPSTATB  
 Core variable/user : ISM  
 Amendments : VE - 23 May 1996 - No amendments needed for initial V32 update  
 Issued : 21 May 2003

**1 Definition**

This variable is coded as

0	not recorded
1	employee
2	self employed
3	unemployed seeking work
4	unemployed about to work
5	sick
6	retired
7	unoccupied
8	government training scheme

**2 FRS Specification**

For each adult record

<u>Code</u>	<u>Condition</u>
0	EMPSTATB=-1,-2,13
1	EMPSTATB=2,3,4,5,6
2	EMPSTATB=1
3	From ADULT table where AGE<65 and SEX=1 or AGE<60 and SEX=2 If WORKING=2 and JOBAWAY=3 and LOOKWK=1,2,3 If WORKING=2, JOBAWAY=2 and LOOK4=1 If WORKING=2, JOBAWAY=2 and LKYT4=1 if WORKING=2, JOBAWAY=2, LIKEWK=1 and NOLOOK=1

- 4 From ADULT table where AGE<65 and SEX=1 or AGE<60 and SEX=2  
If WORKING=2 and JOBAWAY=2 and WAIT=1
- 5 EMPSTATB=11,12
- 6 EMPSTATB=9
- 7 EMPSTATB=10
- 8 EMPSTATB=8

**EMPSTATB**

Purpose : To indicate the employment status of each adult.  
 Created : 8 January 1993  
 Database Table : ADULT  
 Minimum Value : 1  
 Maximum Value : 10  
 Units : Integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : ~~ASD6A~~ **ASD4A**  
 Issue date : 21 May 2003  
 Amendments : VC - 13 January 1993  
               : VC - 20 April 1993 - includes additional categories which were incorrectly :  
               : thrown out as not derivable.  
               : VC - 23 April 1993 To expand definition to show meaning of :  
               : questions/database variables.  
               : VC - 12 May 1993 To add a category to include students and people taking :  
               : industrial action.  
               : VC 14 June 1993 To expand categories as more adults unclassified than :  
               : necessary.  
               : JS 19 January 1996 to do a belt and braces job on students  
               : JS 8 March 1996 - to amend codes for reasons for absence from work  
               : VE - 24 May 1996 - No initial updates needed for V32

**1 Definition**

This variable is coded as

- 1 Self-employed
- 2 Full-time employee at work
- 3 Part-time employee at work
- 4 Full-time employee temporarily not working
- 5 Part-time employee temporarily not working
- 6 Industrial action
- 7 Unemployed
- 8 Work-related government training programme
- 9 Retired - unoccupied minimum NI age
- 10 Unoccupied - under minimum NI age
- 11 Sick - temporarily sick for less than 28 weeks
- 12 Sick - long-term sick/disabled for more than 28 weeks
- 13 Students and adults in non-advanced full-time education
- 1 Not applicable to this case
- 2 Unable to derive

The employment status variable is derived in the main from the ADULT table using a variety of variables to indicate the current employee status of all adults. Part-time and full-time work have been separated using the Income Support definition of full-time remunerative work - 16 hours a week or more.

NB - People who are at home looking after children are included in the unoccupied category (10).

## 2 FRS Specification

For each adult

Code   Condition

- 1      From ADULT table, if WORKING = 1 and EMPSTAT = 2 or  
If WORKING = 2, JOBAWAY = 1 and EMPSTAT = 2.  
(Includes those not working within the last 7 days but do have a job to return to.)
- 2      From ADULT table, if EMPSTAT = 1 and TOTHOURS is greater than or equal to 16 and  
WORKING = 1 or JOBAWAY = 1 and  
TDAYWRK = 1 or {TDAYWRK = 2 or 3 and ABSWK = 2 or ABSWHY = 1 or 3}  
(Includes those not working within last 7 days but are working today.)
- 3      From ADULT table, if EMPSTAT = 1 and TOTHOURS is less than 16 and  
WORKING = 1 or JOBAWAY = 1 and  
TDAYWRK = 1 or {TDAYWRK = 2 or 3 and ABSWK = 2 or ABSWHY = 1 or 3}  
(Includes those not working within last 7 days but are working today.)
- 4      From ADULT table, if EMPSTAT = 1 and TOTHOURS is greater than or equal to 16 and  
WORKING = 1 or JOBAWAY = 1 and  
INJLONG = 1 or  
TDAYWRK = 2 or 3 and {ABSWK = 2 or ABSWHY = 2 5 6, 7 or 8}
- 5      From ADULT table, if EMPSTAT = 1 and TOTHOURS is less than 16 and  
WORKING = 1 or JOBAWAY = 1 and  
INJLONG = 1 or  
TDAYWRK = 2 or 3 and {ABSWK = 2 or ABSWHY = 2 5 6, 7 or 8}
- 6      From ADULT table, if EMPSTAT = 1 and  
WORKING = 1 or JOBAWAY = 1 and  
TDAYWRK = 2 or 3 and ABSWHY = 4.
- 7      Code 7 for the following - for all adults who are below pension age  
From ADULT table, where AGE lt 65 and SEX = 1 and where AGE lt 60 and SEX = 2  
If WORKING = 2 and JOBAWAY = 3 and LOOKWK = 1 2 or 3  
If WORKING = 2, JOBAWAY = 2 and LOOK4 = 1  
If WORKING = 2, JOBAWAY = 2 and LKYT4 = 1  
If WORKING = 2, JOBAWAY = 2 and WAIT = 1  
If WORKING = 2, JOBAWAY = 2, LIKEWK = 1 and NOLOOK = 1
- 8      From ADULT table, if TRAIN = 1,2,4, 5 or 6
- 9      Code 9 for the following  
From ADULT table, where AGE ge 65 and SEX = 1 and where AGE ge 60 and SEX = 2
- 10     Code 10 for the following - for all adults who are below pension age  
From ADULT table, as above  
If WORKING = 2, JOBAWAY = 2, LIKEWK = 1 and NOLOOK = 3 or 6 or 7 or 8

If WORKING = 2, JOBAWAY = 2, LIKEWK = 2 and NOWANT = 3 or 6 or 7 or 8  
 If WORKING = 2, JOBAWAY = 2 and Nolk0002 = 1. (Includes those not working as looking after children)

- 11 Code 11 for the following - for all adults below pension age  
 From ADULT table,  
 If WORKING = 2, JOBAWAY = 2, LIKEWK = 1, NOLOOK = 4 or  
 If WORKING = 2, JOBAWAY = 2, LIKEWK = 2, NOWANT = 4 or  
 If INJLONG = 1.
- 12 As above from ADULT table,  
 If WORKING = 2, JOBAWAY = 2, LIKEWK = 1, NOLOOK = 5 or  
 If WORKING = 2, JOBAWAY = 2, LIKEWK = 2, NOWANT = 5 or  
 If INJLONG = 2
- 13 From ADULT table, if FTED = 1 and TYPEED = 4,5 or 7 or  
 If TEA = 96 and TYPEED = 4 or 5 or 7  
 If WORKING = 2 and JOBAWAY = 2 and NOWANT = 2 or NOLOOK = 2
- 1 Not applicable to this case

Key to above coding

- Working Any paid work in last 7 days? 1 = yes, 2 = no.
- Jobaway If not doing paid work, do you have a job that away from last 7 days. 1 = yes, 2 = no, 3 = waiting take up new job.
- Empstat1 = employed  
 2 = self-employed (including Enterprise Allowance)
- Tdaywrk Are you going to work today. 1 = yes, 2 = no although normal working day, 3 = no not normal working day.
- Tothours Number of hours worked by an employee - a derived variable in itself.
- Abswk Have you been away from work for more than 3 working days. 1 = yes, 2 = no.
- Abswhy Reason for absence. 1 = pattern of shifts, 2 = illness/accident, 3 = holiday, 4 = strike, 5 = Laid off, 6 = maternity leave, 7 = compassionate leave, 8=other.
- Injlong How long been unable to work due to illness/injury. 1 = 6 months or less, 2 = more than 6 months.
- Look4 Looking for work in last 4 weeks. 1 = yes, 2 = no.
- Lkyt4 Looking for place on government scheme in last 4 weeks. 1 = yes, 2 = no.
- Wait Were you waiting to take up a job already obtained. 1 = yes, 2 = no
- Likewk Would you like to have a regular paid job. 1 = yes, 2 = no.

Nolook Why not look. 1 = waiting results job application, 2 = student, 3 = looking after family/home, 4 = temporarily sick, 5 = long-term sick, 6 = believes no jobs available, 7 = retired, 8 = any other reason.

Nowant Why not want. 1 = waiting results of job application, 2 = student, 3 = looking after family or home, 4 = temporarily sick/injured, 5 = long-term sick or disabled, 6 = doesn't need employment, 7 = retired from paid work, 8 = any other reason.

fted Still in full-time education (only asked of 16 to 18 year olds). 1 = yes, 2 = no.

TEA At what age did person leave full-time education (only asked if aged over 18). If answer is 96 the person is still receiving full-time education.

Typed What type of school or college does person attend. 3 = state run special school, 4 = secondary school, 5 = non-advanced further education/6th form/tertiary/further education college. (1 and 2 are nursery and junior schools)

### 3 Results

Tabulation is required to show the number of adults falling into each category.

**EMPSTATI**

Purpose : To indicate a person's employment status using the ILO definition.  
 Created : VC - 17 May 1993  
 Database Table : ADULT  
 Minimum Value : 1  
 Maximum Value : 6  
 Units : Integer  
 Validations :  
 Related Variables : EMPSTATB (Indicates employment status by BU)  
     Children :  
     Parents :  
 Core variable/user : ASD6A ASD4A  
 Issue Date : 21 May 2003  
 Amendments : VC 9 June 1993 To put people who are on holiday from their normal place :  
                   of work into category 1  
                   : VE - 24 May 1996 - Initial updates for V32 - amendments of  
                   TRAIN for new category definitions

**1 Definition**

This variable is coded as

- 1 Employee (includes people not at work due to pattern of shifts)
- 2 Self-employed
- 3 Employee temporarily not at work (including those less than 28 weeks sick, laid off, on strike or on maternity leave)
- 4 Work-related Government Training
- 5 Unemployed (ILO definition)
- 6 Inactive
- 1 Not applicable to this case
- 2 Unable to derive due to missing values

EMPSTATI is derived from several variables in the ADULT table of the FRS database.

An employee is derived from the variables WORKING = 1 (has paid work in last 7 days) or where WORKING = 2 (no work in last 7 days) but JOBAWAY = 1 (has a job to return to). Once it has been established that the person is working, EMPSTAT = 1 then indicates that the person is an employee. All employees are then asked whether they are going to work today, if the answer is yes (TDAYWRK = 1) the person can be automatically classed as an employee. However, if the person answers no (TDAYWRK = 2 or 3) further investigation must be conducted to find out whether the absence is for less than 3 working days (ABSWK = 2) they are also classed as employees. If ABSWK = 1 the reason for the absence must be found from ABSWHY. If ABSWHY = 1 the absence is due to normal pattern of shifts or ABSWHY = 3 on holiday, the person may be classed an employee).

Self-employed people are derived where WORKING = 1 or WORKING = 2 and JOBAWAY = 1 (as above) and EMPSTAT = 2 (self-employed). The question TDAYWRK is not asked of people who declare themselves self-employed.

There are several ways to indicate that an employee is temporarily absent from their normal employment also using the ABSWHY question. Where ABSWHY = 2 the person is off sick and the period of sickness



must be checked in ABS1PD (if it is less than 28 weeks the person falls into code 2 if not they are classed as inactive). If ABSWHY = 4 they are on strike, ABSWHY = 5 they are temporarily laid off or if ABSWHY = 6 they are on maternity leave.

The variable TRAIN indicates whether or not a person is on a work-related government training scheme and codes 1 to 5 6 indicate a government scheme (~~1 = ET, 2 = Youth Training, 3 = Voluntary Projects Programme, 4 = Community Industry, 5 = Other government programme~~ **1 - Youth Training, 2 - Training for Work, 3 - Business Start-up Scheme, 4 - Career Development Loans/Youth Credits, 5 - Community Action, 6 - Other training scheme**). **People on Business Start-up Schemes are treated as 'Working' and self-employed.**

The ILO definition of unemployment is available for and actively seeking work. So where WORKING = 2 (not working) and JOBAWAY = 2 (no job to return to) we would check to see if the person is available for and looking for work. LOOK4 = 1 indicates that the person is looking for work in the last 4 weeks, LKYT4 = 1 indicates that he/she has been looking for a government scheme, WAIT = 1 indicates that he/she is waiting to start a new job, LIKEWK = 1 and NOLOOK = 1 indicates that the person is not looking for employment as he/she is waiting to start a new job.

Any other person will be caught by the inactive group which acts as a catch all.

## 2 FRS Specification

For each adult get all variables from ADULT table

### Code   Condition

- |   |  |
|---|--|
| 1 | If WORKING = 1 and EMPSTAT = 1 and TDAYWRK = 1 or<br>If WORKING = 1 and EMPSTAT = 1 and TDAYWRK = 2 or 3 and ABSWK = 2 or<br>If WORKING = 1 and EMPSTAT = 1 and TDAYWRK = 2 or 3 and ABSWHY = 1 or 3 (shifts or holiday) or<br>If JOBAWAY = 1 and EMPSTAT = 1 and TDAYWRK = 2 or 3 and ABSWK = 2 or<br>If JOBAWAY = 1 and EMPSTAT = 1 and TDAYWRK = 2 or 3 and ABSWHY = 1 or 3 (shifts or holiday)                             |
| 2 | If WORKING = 1 and EMPSTAT = 2 or<br>If JOBAWAY = 1 and EMPSTAT = 2 <b>or</b><br><b>If TRAIN=3</b>   |
| 3 | If WORKING = 1 and EMPSTAT = 1 and TDAYWRK = 2 or 3 and ABSWHY = 2 and ABS1PD <= 28 or (not working today, absent more than 3 days & reason illness/accident) or<br>If WORKING = 1 and EMPSTAT = 1 and TDAYWRK = 2 or 3 and ABSWHY = 4 or 5 or 6 or<br>If JOBAWAY = 1 and EMPSTAT = 1 and TDAYWRK = 2 or 3 and ABSWHY = 2 and ABS1PD <= 28 or<br>If JOBAWAY = 1 and EMPSTAT = 1 and TDAYWRK = 2 or 3 and ABSWHY = 4 or 5 or 6. |
| 4 | If TRAIN = 1, 2, 3, 4, <del>5</del> <b>or 6</b>  |
| 5 | If WORKING = 2, JOBAWAY = 2 and<br>If LOOK4 = 1 or (looking for work)<br>If LKYT4 = 1 or (looking for govt scheme)<br>If WAIT = 1 or (waiting start job)   |

If LIKEWK = 1 and NOLOOK = 1 (would like regular work and reason not looking = waiting start new job/business)

6 Anyone else not previously coded.

-1 Not applicable (should not be relevant to EMPSTATI)

-2 Unable to derive as any of the above variables are missing

### **3 Results**

Tabulation required to show the number of adults falling into each category and a breakdown of the cases caught in code 6 (inactive).

### **4 Test Cases**

None as yet.

**EQAHCBU**

Purpose : Equivalence scale after housing costs for benefit unit which will be used to calculate the equivalised income after housing costs for a benefit unit.  
 Created : VC - 24 September 1993  
 Database Table : BENUNIT  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables :  
     Children :  
     Parents : SPCALC  
 Core variable/user : HBAI  
 Amendments : VC - 11 October 1993 To identify equivalence scales used.  
               : VC - 11 February 1994 Amendments to reflect version 30 changes.  
               : VE - 4 June 1996 - No initial amendments needed for V32 update  
               : VE - 1 July 1996 - Amended for constants being held in a separate table  
 Issued : 21 May 2003

**1 Definition**

This variable is coded as

EQAHCBU This is a scale calculated for the benefit unit depending on the number of adults and children using the McClement's scale in the same way as FES. **The constants are held in h:\frs\dvars\docs\spreads.xls as V32Constants.** The relevant values from the following table are accumulated to provide the benefit unit scale.

1st adult (head of household)	0.55	<b>HDBU1A</b>
spouse of head	0.45	<b>SPBU1A</b>
heads of subsequent units	0.45	<b>HDBUXA</b>
spouse of any subsequent units	0.45	<b>SPBUXA</b>
each dependent aged		
0 - 1	0.07	<b>D01A</b>
2 - 4	0.18	<b>D24A</b>
5 - 7	0.21	<b>D57A</b>
8 - 10	0.23	<b>D810A</b>
11 - 12	0.26	<b>D1112A</b>
13 - 15	0.28	<b>D1315A</b>
16 +	0.38	<b>D16+A</b>

-2           Unable to derive due to missing values

The resultant scale will be used to calculate a further income variable for the benefit unit. The individual groups for the above list are identified using the benefit unit number (BENUNIT) and the variables HEAD

and SPOUSE which are calculated using the SPCALC sub-programme plus the age of every child (AGE from CHILD record).

The head of the benefit unit is defined as the first adult in that benefit unit and could, therefore, be either male or female depending on the order the interviewer used during the interview. ~~Is this OK it can be changed if you need anything different.~~

NB - There shouldn't be any not applicable cases as all households have at least one adult.

**2 FRS Specification**

For each BENUNIT record, set EQAHCBU to zero

<u>Code</u>	<u>Condition</u>		
EQAHCBU	Process each ADULT record and accumulate the following for each adult in the benefit unit.		
	If BENUNIT = 1 and HEAD = 1 (head of household) add	<del>0.55</del>	<b>HDBU1A</b>
	If BENUNIT = 1 and SPOUSE = 1 (spouse/partner) add	<del>0.45</del>	<b>SPBU1A</b>
	If BENUNIT >= 2 and HEAD = 1 add	<del>0.45</del>	<b>HDBUXA</b>
	If BENUNIT >= 2 and SPOUSE = 1 add	<del>0.45</del>	<b>SPBUXA</b>
	Process each CHILD record and accumulate the following for each child in the benefit unit.		
	If AGE <= 1 add	<del>0.07</del>	<b>D01A</b>
	If AGE >= 2 and <= 4 add	<del>0.18</del>	<b>D24A</b>
	If AGE >= 5 and <= 7 add	<del>0.24</del>	<b>D57A</b>
	If AGE >= 8 and <= 10 add	<del>0.23</del>	<b>D810A</b>
	If AGE >= 11 and <= 12 add	<del>0.26</del>	<b>D1112A</b>
	If AGE >= 13 and <= 15 add	<del>0.28</del>	<b>D1315A</b>
	If AGE >= 16 add	<del>0.38</del>	<b>D16+A</b>
-2	If any of above values are missing		

**3 Results**

None required as will be used as a component of other variables.

**EQBHCBU**

Purpose : Equivalence scale before housing costs for benefit unit which will be used to calculate the equivalised income before housing costs for a benefit unit.  
 Created : VC - 6 October 1993  
 Database Table : BENUNIT  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables :  
     Children :  
     Parents : SPCALC  
 Core variable/user : HBAI  
 Amendments : VC - 11 October 1993 To identify the equivalence scale used.  
               : VC - 11 February 1994 Amendments to reflect verion 30 changes  
               : VE - 4 June 1996 - No initial amendments needed for V32  
               : VE - 1 July 1996 - Amended for constants being held in a separate table  
 Issued : 21 May 2003

**1 Definition**

This variable is coded as

EQBHCBU This is a scale calculated for the benefit unit depending on the number of adults and children using the McClements table of scales in the same way as FES. **The constants are held in h:\frs\dvars\docs\spreads.xls as V32Constants.** The relevant values from the following table are accumulated to provide the benefit unit scale.

1st adult (head of household)	0.64	<b>HDBU1B</b>
spouse of head	0.39	<b>SPBU1B</b>
heads of subsequent units	0.46	<b>HDBUXB</b>
spouse of any subsequent units	0.42	<b>SPBUXB</b>
each dependent aged		
0 - 1	0.09	<b>D01B</b>
2 - 4	0.18	<b>D24B</b>
5 - 7	0.21	<b>D57B</b>
8 - 10	0.23	<b>D810B</b>
11 - 12	0.25	<b>D1112B</b>
13 - 15	0.27	<b>D1315B</b>
16 +	0.36	<b>D16+B</b>

-2           Unable to derive due to missing values

The resultant scale will be used to calculate a further income variable for the benefit unit. The individual groups for the above list are identified using the benefit unit number (BENUNIT) and the variables HEAD

and SPOUSE which are calculated using the SPCALC sub-programme plus the age of every child (AGE from CHILD record).

The head of the benefit unit is defined as the first adult in that benefit unit and could, therefore, be either male or female depending on the order the interviewer used during the interview. ~~Is this OK it can be changed if you need anything different.~~

NB - There shouldn't be any not applicable cases as all households have at least one adult.

**2 FRS Specification**

For each BENUNIT record, set EQBHCBU to zero

<u>Code</u>	<u>Condition</u>		
EQBHCBU	Process each ADULT record and accumulate the following for each adult in the benefit unit.		
	If BENUNIT = 1 and HEAD = 1 (head of household) add	<del>0.64</del>	<b>HDBU1B</b>
	If BENUNIT = 1 and SPOUSE = 1 (spouse/partner) add	<del>0.39</del>	<b>SPBU1B</b>
	If BENUNIT >= 2 and HEAD = 1 add	0.45	<b>HDBUXB</b>
	If BENUNIT >= 2 and SPOUSE = 1 add	0.42	<b>SPBUXB</b>
	Process each CHILD record and accumulate the following for each child in the benefit unit.		
	If AGE <= 1 add	<del>0.09</del>	<b>D01B</b>
	If AGE >= 2 and <= 4 add	<del>0.18</del>	<b>D24B</b>
	If AGE >= 5 and <= 7 add	<del>0.24</del>	<b>D57B</b>
	If AGE >= 8 and <= 10 add	<del>0.23</del>	<b>D810B</b>
	If AGE >= 11 and <= 12 add	<del>0.25</del>	<b>D1112B</b>
	If AGE >= 13 and <= 15 add	<del>0.27</del>	<b>D1315B</b>
	If AGE >= 16 add	<del>0.36</del>	<b>D16+B</b>
-2	If any of above values are missing		

**3 Results**

None required as will be used as a component of other variables.

**EQUIVAHC**

Purpose : Household after housing costs equivalence scale which will be used to calculate the household equivalised income after housing costs. :

Created : VC - 23 September 1993

Database Table : HOUSEHOL

Minimum Value : 0

Maximum Value :

Units : Real

Validations :

Related Variables :

    Children : EQVAHCHH

    Parents :

Core variable/user : HBAI

Amendments : VC - 11 October 1993 To identify the equivalence scale used.  
 : VE - 22 May 1996 - No initial amendments needed for V32  
 : VE - 1 July 1996 - Amended for constants being held in a separate table  
 : VE - 23 September 1996 - To clarify spouse of head as being in benunit 1

Issued : 21 May 2003

**1 Definition**

This variable is coded as

EQUIVAHC This is a scale calculated for the household depending on the number of adults and children using the McClements table of scales in the same way as FES. **The constants are held in h:\frs\dvars\docs\spreads.xls as V32Constants.** The relevant values from the following table are accumulated to provide the household scale.

1st adult (head of household)	0.55	<b>AD1A</b>
spouse of head	0.45	<b>AD2A</b>
or other 2nd adult	0.45	<b>AD2A</b>
3rd adult	0.45	<b>AD3A</b>
or any subsequent adult	0.40	<b>SUBSADA</b>
each dependent aged		
0 - 1	0.07	<b>DEP01A</b>
2 - 4	0.18	<b>DEP24A</b>
5 - 7	0.21	<b>DEP57A</b>
8 - 10	0.23	<b>DEP810A</b>
11 - 12	0.26	<b>DEP1112A</b>
13 - 15	0.28	<b>DEP1315A</b>
16 +	0.38	<b>DEP16+A</b>

-2           Unable to derive due to missing values

The resultant scale will be used to calculate a further income variable for the household. The individual groups for the above list are identified using the person number (PERSON) and the marital status of each adult (MS) and the age of every child (AGE from CHILD record).

NB - There shouldn't be any not applicable cases as all households have at least one adult.

**2 FRS Specification**

For each HOUSEHOLD, set EQUIVAHC to zero

<u>Code</u>	<u>Condition</u>		
EQUIVAHC	Process each ADULT record and accumulate the following for each adult in the household		
	If PERSON = 1 (head of household) add	<del>0.55</del>	<b>AD1A</b>
	If PERSON = 2 and <b>BENUNIT=1</b> and MS = 1 or 3 (spouse/partner) add	<del>0.45</del>	
			<b>AD2A</b>
	<b>Else if PERSON = 2</b> add	<del>0.45</del>	<b>AD2A</b>
	If PERSON = 3 add	<del>0.45</del>	<b>AD3A</b>
	If PERSON > 3 add for each subsequent adult	<del>0.40</del>	<b>SUBSADA</b>
	Process each CHILD record and accumulate the following for each child in the household		
	If AGE <= 1 add	<del>0.07</del>	<b>DEP01A</b>
	If AGE >= 2 and <= 4 add	<del>0.18</del>	<b>DEP24A</b>
	If AGE >= 5 and <= 7 add	<del>0.24</del>	<b>DEP57A</b>
	If AGE >= 8 and <= 10 add	<del>0.23</del>	<b>DEP810A</b>
	If AGE >= 11 and <= 12 add	<del>0.26</del>	<b>DEP1112A</b>
	If AGE >= 13 and <= 15 add	<del>0.28</del>	<b>DEP1315A</b>
	If AGE >= 16 add	<del>0.38</del>	<b>DEP16+A</b>
-2	If any of above values are missing		

**3 Results**

None required as will be used as a component of other variables.



**EQUIVBHC**

Purpose : Household before housing costs equivalence scale which will be used to calculate the household equivalised income before housing costs.

Created : VC - 23 September 1993

Database Table : HOUSEHOL

Minimum Value : 0

Maximum Value :

Units : Real

Validations :

Related Variables :

    Children : EQVBHCHH

    Parents :

Core variable/user : HBAI

Amendments : VC - 11 October 1993 To identify the equivalence scale used.  
 : VE - 22 May 1996 - No initial amendments needed for V32  
 : VE - 1 July 1996 - Amended for constants being held in a separate table  
 : VE - 23 September 1996 - To clarify spouse of head as being PERSON=2 and BENUNIT=1

Issued : 21 May 2003

**1 Definition**

This variable is coded as

EQUIVBHC This is a scale calculated for the household depending on the number of adults and children using the McClements table of scales in the same way as FES. **The constants are held in h:\frs\dvars\docs\spreads.xls as V32Constants.** The relevant values from the following table are accumulated to provide the household scale.

1st adult (head of household)	0.64	<b>AD1B</b>
spouse of head	0.39	<b>SPHDB</b>
or other 2nd adult	0.46	<b>AD2B</b>
3rd adult	0.42	<b>AD3B</b>
or any subsequent adult	0.36	<b>SUBSADB</b>
each dependent aged		
0 - 1	0.09	<b>DEP01B</b>
2 - 4	0.18	<b>DEP24B</b>
5 - 7	0.21	<b>DEP57B</b>
8 - 10	0.23	<b>DEP810B</b>
11 - 12	0.25	<b>DEP1112B</b>
13 - 15	0.27	<b>DEP1315B</b>
16 +	0.36	<b>DEP16+B</b>

-2           Unable to derive due to missing values

The resultant scale will be used to calculate a further income variable for the household. The individual groups for the above list are identified using the person number (PERSON) and the marital status of each adult (MS) and the age of every child (AGE from CHILD record).

NB - There shouldn't be any not applicable cases as all households have at least one adult.

**2 FRS Specification**

For each HOUSEHOLD, set EQUIVBHC to zero

<u>Code</u>	<u>Condition</u>		
EQUIVBHC	Process each ADULT record and accumulate the following for each adult in the household		
	If PERSON = 1 (head of household) add	0.64	<b>AD1B</b>
	If PERSON = 2 <b>and BENUNIT =1</b> and MS = 1 or 3 (spouse/partner) add	0.39	
			<b>SPHDB</b>
	<b>Else</b> if PERSON = 2	0.46	<b>AD2B</b>
	If PERSON = 3 add	0.42	<b>AD3B</b>
	If PERSON > 3 add for each subsequent adult	0.36	<b>SUBSADB</b>
	Process each CHILD record and accumulate the following for each child in the household		
	If AGE <= 1 add	0.09	<b>DEP01B</b>
	If AGE >= 2 and <= 4 add	0.18	<b>DEP24B</b>
	If AGE >= 5 and <= 7 add	0.24	<b>DEP57B</b>
	If AGE >= 8 and <= 10 add	0.23	<b>DEP810B</b>
	If AGE >= 11 and <= 12 add	0.25	<b>DEP1112B</b>
	If AGE >= 13 and <= 15 add	0.27	<b>DEP1315B</b>
	If AGE >= 16 add	0.36	<b>DEP16+B</b>
-2	If any of above values are missing		

**3 Results**

None required as will be used as a component of other variables.

**FAMTHBAI**

Purpose : This is the family type used for HBAI purposes for each benefit unit.  
 Created : VC - 9 September 1993  
 Database Table : BENUNIT  
 Minimum Value : 1  
 Maximum Value : 6  
 Units : Integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : HBAI  
 Amendments : to fit in with HBAI definitions: FAMTHBAI =1 when head of benefit unit over pensionable age; state pension definition  
                   : JS - 13 March 1996 - to tighten up definitions and amend spec in line with ASD6A changes  
                   : VE - 4 June 1996 - No initial amendments needed for V32  
 Issued : 21 May 2003

**1 Definition**

This variable is coded as

FAMTHBAI This is the definition of family type used by HBAI.

1	Pensioner couple
2	Pensioner single
3	Couple with children
4	Couple without children
5	Lone parent
6	Single without children.

FAMTHBAI is now derived independently from FAMTYPE, but needs to have checks on whether a second adult is present for couples to ensure correct definitions are set up (see FAMTYPE for further information).

According to the HBAI publication, pensioner/non-pensioner singles/couples are where they are headed by a someone over/under state pension age. However, for the FES, the head of the benefit unit is always the man whereas for the FRS female heads are possible.

This causes problems for cases where MS=2 (married, spouse not in household) and the head (partner who is in the household) is female as cases would be set to unable to derive. The female is still taken as the head in these cases. NOTE: HBAI delete these cases from their file.

**2 FRS Specification**

Codes are hierarchical, ie if have children but head is over pension age, BUs fall in to code 1 and not 3, or code 2 and not 5 (similarly not codes 4 and 6 if no children).

ie the variable is coded sequentially from top down (1 to 6) and is set according to the first condition fulfilled

<u>Code</u>	<u>Condition</u>
1	If MS=1,3,4,5 6 or 7 and (second adult in the benefit unit exists) and man in the benefit unit is over 65 [ie (UPERSON=1 and SEX=1 and AGE>=65) or (UPERSON=2 and SEX=1 and AGE>=65)]  or MS=2 and (UPERSON=1 and SEX=1 and AGE>=65) or (UPERSON=1 and SEX=2 and AGE>=60) man or woman over pensionable age where MS=married, spouse not in household
2	If MS=1,3,4,5,6 or 7 and (only one adult in the benefit unit) and person is over pension age [ie (UPERSON=1 and SEX=1 and AGE>=65) or (UPERSON=1 and SEX=2 and AGE>=60)]
3	If MS= 1,3,4,5,6 or 7 and (second adult in the benefit unit exists) and depchildb > 0 or MS=2 and depchildb>0
4	If MS=1,3,4,5,6 or 7 and (second adult in the benefit unit exists) and depchildb=0 or MS=2 and depchildb=0
5	If MS=1,3,4,5,6 or 7 and only one adult in the benefit unit and depchildb>0
6	If MS=1,3,4,5,6,7 and only one adult in the benefit unit and depchildb=0

**3 Results**

Tabulation will be required to show the number and percentage of benefit units falling into each category.

**FAMTYPBU**

Purpose : This is the family type used for publication purposes for each benefit unit. It is consistent with the HBAI variable FAMTHBAI except that pensioner benefit units are defined on the basis of the head of the benefit unit, be it male or female

Created : VC - 9 September 1993

Database Table : BENUNIT

Minimum Value : 1

Maximum Value : 6

Units : Integer

Validations :

Related Variables :

    Children :

    Parents :

Core variable/user : HBAI **ASD4A**

Amendments : to fit in with HBAI definitions: FAMTYPBU =1 when head of benefit unit over pensionable age; state pension definition  
: JS - 13 March 1996 - to tighten up definitions and amend spec in line with ASD6A changes  
: VE - 4 June 1996 - No initial amendments needed for V32  
: VE - 20 June 1996 - To change lead user to ASD4A in response to request from HBAI

Issued : 21 May 2003

**1 Definition**

This variable is coded as

FAMTYPBU This is the definition of family type used for the publication.

1	Pensioner couple
2	Pensioner single
3	Couple with children
4	Couple without children
5	Lone parent
6	Single without children.

FAMTYPBU is now derived independently from FAMTYPE, but needs to have checks on whether a second adult is present for couples to ensure correct definitions are set up (see FAMTYPE for further information).

According to the HBAI publication, pensioner/non-pensioner singles/couples are where they are headed by a someone over/under state pension age. However, for the FES, the head of the benefit unit is always the man whereas for the FRS female heads are possible. Since for the publication, tables have been produced which show the age of the head of benefit unit, for consistency, pensioner FAMTYPBU cases are based on the age of the head, regardless of the sex.

This causes problems for cases where MS=2 (married, spouse not in household) and the head (partner who is in the household) is female as cases would be set to unable to derive. The female is still taken as the head in these cases. NOTE: HBAI delete these cases from their file.

## 2 FRS Specification

Codes are hierarchical, ie if have children but head is over pension age, BUs fall in to code 1 and not 3, or code 2 and not 5 (similarly not codes 4 and 6 if no children).

ie the variable is coded sequentially from top down (1 to 6) and is set according to the first condition fulfilled

<u>Code</u>	<u>Condition</u>
1	If MS=1,3,4,5 6 or 7 and (second adult in the benefit unit exists) and the head of the benefit unit is over pension age [ie (UPERSON=1 and SEX=1 and AGE>=65) or (UPERSON=1 and SEX=2 and AGE>=60)]  or MS=2 and (UPERSON=1 and SEX=1 and AGE>=65) or (UPERSON=1 and SEX=2 and AGE>=60) man or woman over pensionable age where MS=married, spouse not in household
2	If MS=1,3,4,5,6 or 7 and (only one adult in the benefit unit) and person is over pension age [ie (UPERSON=1 and SEX=1 and AGE>=65) or (UPERSON=1 and SEX=2 and AGE>=60)]
3	If MS= 1,3,4,5,6 or 7 and (second adult in the benefit unit exists) and depchildb > 0 or MS=2 and depchildb>0
4	If MS=1,3,4,5,6 or 7 and (second adult in the benefit unit exists) and depchildb=0 or MS=2 and depchildb=0
5	If MS=1,3,4,5,6 or 7 and only one adult in the benefit unit and depchildb>0
6	If MS=1,3,4,5,6,7 and only one adult in the benefit unit and depchildb=0

## 3 Results

Tabulation will be required to show the number and percentage of benefit units falling into each category.

**FAMTYPE**

Purpose : Family Type Indicator for each Benefit Unit  
 Created : AJG 1 December 1992  
 Database Table : BENUNIT  
 Minimum Value : 1  
 Maximum Value : 8  
 Units : Integer  
 Validations :  
 Related D Variables :  
     Children :  
     Parents :  
 Core variable/user : ISM FCM **PSM** (**Bold** indicates lead user)  
 Issue date : 21 May 2003  
 Amendments : VC - 22 January 1993  
               : VC - 23 March 1993 To change the variables used in the coding to make the :  
               : derivation much simpler.  
               : VC - 7 May 1993 To use income-related benefits definition of pension age ie :  
               : 60 and over  
               : VC - 11 February 1994 Amendments to reflect version 30 changes.  
               : JS - 13 March 1996, to tighten up definition  
               : VE - 4 June 1996 - No initial amendments needed for V32

**1 Definition**

This variable is coded as

1	Under pension age	with dependants	couple
2			single
3	Under pension age	without dependants	couple
4			single
5	Over pension age	with dependants	couple
6			single
7	Over pension age	without dependants	couple
8			single
-2	Unable to derive		

For the purpose of this variable:

Pension age is taken to be 60 and over (ie the income-related benefits pension age).

Couples and single people are identified by the variable marital status in the ADULT table. Couples are to include MS = 1 (married spouse in household), MS = 2 (married spouse not in household), MS = 3 (cohabiting). Single people will be the rest - MS = 4 (single, never married), MS = 5 (widowed), MS = 6 (separated) and MS = 7 (divorced) It is intended that in future, difficulties arising from not knowing the status of the absentee partner involved in MS = 2, will be resolved and these cases will cover only those who should be treated as a couple for benefit purposes.

The variable DEPCHLDB, which is produced by the questionnaire programme itself and is not related to any particular question, indicates the total number of dependent children in a benefit

unit. Dependants include those aged 15 and under, and those aged 16,17 and 18 in full time non-advanced education.

Following investigation of V30, it appears that some benefit units, marital status does not match up with benefit unit details (eg, people coded as married but where a spouse is not present). Instead of identifying singles/couples on the basis of their marital status, a check is added to see if a second adult (spouse) does/does not exist. However, for MS=2 (married spouse not in household), the absence of a partner is legitimate (typical example - husband working abroad on 9 month contract thereby failing the 6 month residency rule for coding members of the household) and these cases have to be allowed for under the relevant couples sections.

## 2 FRS Specification

For each BENUNIT record

Adult under or over pension age is determined as

If AGE >= 60 = over pension age  
If AGE < 60 = under pension age

Benefit Unit under or over is determined as any adult in the Benefit Unit over pension age = Benefit Unit over pension age.

Therefore FAMTYPE is determined as

### CODE CONDITION

- |    |   |
|----|---|
| 1  | If under pension age and DEPCHILDB > 0 and ((second adult record in BU exists) or MS=2 where single adult record) |
| 2  | If under pension age and DEPCHILDB > 0 and only one adult record in BU and MS<>2                                  |
| 3  | If under pension age and DEPCHILDB = 0 and ((second adult record in BU exists) or MS=2 where single adult record) |
| 4  | If under pension age and DEPCHILDB = 0 and only one adult record in BU and MS<>2                                  |
| 5  | If over pension age and DEPCHILDB > 0 ((and second adult record in BU exists) or MS=2 where single adult record)  |
| 6  | If over pension age and DEPCHILDB > 0 and only one adult record in BU and MS<>2                                   |
| 7  | If over pension age and DEPCHILDB = 0 and ((second adult record in BU exists) or MS=2 where single adult)         |
| 8  | If over pension age and DEPCHILDB = 0 and only one adult record in BU and MS<>2                                   |
| -2 | If variable cannot be derived ie if any of DEPCHILDB AGE, SEX or MS are missing                                   |

## 3 Results

Tabulate numbers in each category.



**FOSTERBU**

Purpose : To indicate the total number of foster children in a benefit unit.  
 Created : VC - 3 March 1993  
 Database Table : BENUNIT  
 Minimum Value : 0  
 Maximum Value :  
 Units : INTEGER  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : ~~HBAI~~ **ASD4A**  
 Amendments : VC - 11 February 1994 Amendments to reflect version 30 changes  
               : VE - 6 June 1996 - No amendments needed for initial V32 update  
               : VE - 20 June 1996 - To change lead user to ASD4A in response to request  
               : from HBAI  
 Issue date : 21 May 2003

**1 Definition**

This variable is coded as

FOSTERBU The total number of foster children in a benefit unit.

-1 Not applicable to this case.

-2 Unable to derive due to missing values.

The variable FOSTERBU is derived from the variable foster which is attached to every child record. FOSTERBU is then the count of each child in the benefit unit where foster = 1. If foster = 2 the child is not a foster child and is not applicable to this case. The variable foster merely asks whether the child is a foster child and is attached to each child record for children up to the age of 15. Dependent children aged 16 or over are not asked the question as they are no longer recognised as being foster children. As a result this group will probably be pulled out as -1 (not applicable).

**2 FRS Specification**

For each BENEFIT UNIT (BENUNIT record)

<u>Code</u>	<u>Condition</u>
FOSTERBU	From CHILD table, for each child in benefit unit If foster = 1, compute count of foster children for that benefit unit.
-1	Not applicable to this case - where FOSTER = 2
-2	Unable to derive due to missing value of FOSTER - probably those dependant children aged 16 - 18.

**3 Results**

Tabulation is required to show the number of foster children by benefit unit in the following categories - None, 1, 2, 3, 4, 5 and over.

**4 Test Cases**

None as yet

**FSMVAL, FSMBU & FSMHH**

Purpose	: This specification produces three variables which calculate the value of free school meals for each dependent, benefit unit and household
Created	: VC - 2 September 1993
Database Table	: CHILD, BENUNIT & HOUSEHOL
Minimum Value	: 0
Maximum Value	:
Units	: Real
Validations	:
Related Variables	: FSMLKVAL, FSMLKBU, FSMLKHH, FWMLKVAL, FWMLKBU & FWMLKHH
Children	:
Parents :	:
Core variable/user	: HBAI
Amendments	: VC - 12 October 1993 To set an amount for the cost of free school meals as : in FES appendix 66
	: VC - 2 November 1993 To take out reference to -1 (not applicable as should : be referred to as 0 in HBAI.
	: VC - 11 February 1994 Amendments to reflect version 30 changes
	: JS - 1 April 1996 to reflect version 31 changes
	: VE - 22 May 1996 - Amendments to reflect initial V32 changes - SCHMEAL replaced by FREEITEM
	: VE - 27 June 1996 - To set amount for the cost of free school meals to April 1995 level
	: VE - 1 July 1996 - Amended for constants being held in a separate table
	: VE - 29 October 1996 - FREEITEM in Blaise code renamed SCHMEAL etc.
Issued	: 21 May, 2003

**1 Definition**

This variable is coded as

FSMVAL	This is the value of any free school meals received by a dependent.
FSMBU	This is the total value of any free school meals received by each benefit unit.
FSMHH	This is the total value of any free school meals received by the household.
-1	Not applicable to this case - applies to all of above variables.
-2	Unable to derive due to missing values - applies to all of above variables.

The value of free school meals received by each dependent is derived from the SCHMEAL ~~FREEITEM~~ and SMLIT variable from the CHILD record. Where SCHMEAL = 1 ~~FREEITEM=4~~ (has some free school meals) the number of free meals is obtained from SMLIT. This amount is then multiplied by the cost of a school meal, which is to be found in the Tax Benefit Model ~~for 1994~~ to produce the total amount spent each week.

~~The total amount of free school meals for a week (from the TBM) is £4.04 £4.30 per week. This amount is then multiplied by 52/40 to give an average throughout the school year and is then divided by 5 to get~~

a daily amount = ~~£1.0504~~ **£1.118** per day. The daily cost of a free school meal is given by the value **COSTM** from the **V32Constants** table held in **h:\frs\dvars\docs\spreads.xls**.

Once the cost of free school meals has been produced for each dependent it must be accumulated for the benefit unit and the household.

## 2 FRS Specification

<u>Code</u>	<u>Condition</u>
FSMVAL	<p>For each CHILD from CHILD table,</p> <p>Set COST (of free school meals) to <del>£1.0504</del> <b>£1.118 COSTM</b> (Supplied to FES from Tax Benefit Model for 1994)</p> <p>If SCHMEAL = 1 <del>FREEITEM=4</del>, calculate the value of free school meals</p> <p>calculate FSMVAL = SMLIT * COST</p> <p><del>If FREEITEM not eq 4 set FSMVAL=0</del></p> <p>For each BENEFIT UNIT</p> <p>sum each occurrence of FSMVAL for each child in benefit unit</p> <p>For each HOUSEHOLD</p> <p>sum each occurrence of FSMVAL for each child in household</p>

NB - Child is FRS version of child ie 15 and under or aged 16 to 19 and in full time education.

0	Not applicable - where case has no children (NUMCHIL/DEPCHILD = 0) or no free school meals.
-2	Unable to derive due to any of above values being missing.

## 3 Results

Tabulation will be required to show the number of children, benefit units and households by the value of their free school meals sorted into the following bands

Under £2.50  
 £2.50 - £5.00  
 £5.00 - £7.50  
 £7.50 - £10.00  
 £10.00 - £12.50  
 £12.50 - £15.00  
 £15.00 - £17.50  
 £17.50 and over

**FSMLKVAL, FSMLKBU & FSMLKHH**

Purpose : This specification produces three variables which calculates the value of free school milk for each child, benefit unit and household

Created : VC - 3 September 1993

Database Table : CHILD, BENUNIT & HOUSEHOL

Minimum Value : 0

Maximum Value :

Units : Real

Validations :

Related Variables : FSMVAL, FSMBU, FSMHH, FWMLKVAL, FWMLKBU & FWMLKHH

Children :

Parents :

Core variable/user : HBAI

Amendments : VC - 2 November 1993 To take out reference to -1 not applicable and replace with 0 .

: JS - 1 April 1996 to update for V31

: VE - 22 May 1996 - Initial amendments for V32 - SCHMILK replaced by FREEITEM

: VE - 14 June 1996 - To update the cost of a pint of milk to 1995-96 rates

: VE - 30 October 1996 - FREEITEM replaced by SCHMILK

Issued : 21 May 2003

**1 Definition**

This variable is coded as

FSMLKVAL This is the total value of any free school milk received by a child.

FSMLKBU This is the total value of any free school milk received by any child in the benefit unit.

FSMLKHH This is the total value of any free school milk received by any child in the household.

0 Not applicable to this case - applies to all of above variables no children or no free milk

-2 Unable to derive due to missing values - applies to all of above variables

The value of free school milk received by each person is derived from the SCHMILK ~~FREEITEM~~ and SMKIT variables from the CHILD record on the database. Where SCHMILK = 1 ~~FREEITEM=3~~ (has some free school milk) the number of pints of milk is obtained from SMKIT. This amount is then multiplied by the cost of each bottle of free school milk to produce the total amount spent each for that child.

The cost of a bottle of free school milk is calculated using the same method that CSO uses for FES. It may need updating each year so check with FES first before running. CSO use the cost of a third of a pint/bottle of milk at ~~34p~~ **36p** per bottle/carton. This gives a cost of a third of a pint to be ~~£0.1133~~ **£0.12**.

Once the cost of free school milk has been produced for each child it must be accumulated for each child in the benefit unit and then the household.

**2 FRS Specification**

<u>Code</u>	<u>Condition</u>
FSMLKVAL	For each CHILD from CHILD record.  If SCHMILK = 1 <del>FREEITEM=3</del> , calculate the value of free school milk calculate FSMLKVAL = SMKIT x cost of free school milk ( <del>0.1133</del> <b>0.12</b> )  If SCHMILK = 2 <del>FREEITEM not eq 3</del> , calculate FSMLKVAL = 0.
FSMLKBU	For each BENEFIT UNIT (for BENUNIT record)  sum each occurrence of FSMLKVAL for each child in benefit unit.
FSMLKHH	For each HOUSEHOLD (for HOUSEHOL record)  sum each occurrence of FSMLKVAL for each child in household.
0	Not applicable - no school milk/no dependents
-2	Unable to derive as any of above variables are missing.

### 3 Results

Tabulation will be required to show the number of children, benefit units and households by the value of free school milk received sorted into the following bands

Under £1.00  
 £1.00 - £2.00  
 £2.00 - £3.00  
 £3.00 - £4.00  
 £4.00 - £5.00  
 £5.00 - £6.00  
 £6.00 or over

### 4 Test Cases

None produced as yet.

**FWMLKVAL, FWMLKBU & FWMLKHH**

Purpose	: This specification produces three variables which calculates the value of free welfare milk for each person (adult & child), benefit unit and household
Created	: VC - 3 September 1993
Database Table	: ADULT, CHILD, BENUNIT & HOUSEHOL
Minimum Value	: 0
Maximum Value	:
Units	: Real
Validations	:
Related Variables	: FSMVAL, FSMBU, FSMHH, FSMLKVAL, FSMLKBU & FSMLKHH
Children	:
Parents :	:
Core variable/user	: HBAI
Amendments	: VC - 2 November 1993 To remove references to -1 not applicable and replaced by 0
	: JS - 1 April 1996 - to update for V31.
	: VE - 22 May 1996 - Amendments to reflect initial V32 changes - WELFMILK replaced by FREEITEM
	: VE - 14 June 1996 - To update cost of a pint of milk to 1995-96 rates
	: VE - 1 July 1996 - Amended for constants being held in a separate table
Issued	: 21 May 2003

**1 Definition**

This variable is coded as

FWMLKVAL	This is the total value of any free welfare milk received by a person (adult or child).
FWMLKBU	This is the total value of any free welfare milk received by any person in the benefit unit.
FWMLKHH	This is the total value of any free welfare milk received by any person in the household.
0	Not applicable to this case - applies to all of above variables
-2	Unable to derive due to missing values - applies to all of above variables

The value of free welfare milk received by each person is derived from the WELFMILK ~~FREEITEM~~ and WMKIT variables from the ADULT and CHILD records on the database. Where WELFMILK = 1 ~~FREEITEM=2~~ (has some free welfare milk) the number of pints of milk is obtained from WMKIT. This amount is then multiplied by the cost of each pint of free welfare milk (to be supplied by the Tax Benefit Model) to produce the total amount spent each for that person.

From 1994/95, it has been agreed that costs should be based on CSO figures for a pint of milk (ie consistent with free school milk calculation) rather than the TBM. The cost of a pint of milk for ~~1994/95~~ **1995-96 is 34p-36p. is given by the value COSTWMK from the V32Constants table held in h:\frs\dvars\docs\spreads.xls.**

Once the cost of free welfare milk has been produced for each person it must be accumulated for each person in the benefit unit and then the household.

**2 FRS Specification**

<u>Code</u>	<u>Condition</u>
FWMLKVAL	<p>For each ADULT from ADULT record.</p> <p>If WELFMILK = 1 <del>FREEITEM=2</del>, calculate the value of free welfare milk</p> <p>calculate FWMLKVAL = WMKIT x <del>cost of free welfare milk (.34 for 1994/95 0.36 for 1995-96)</del> <b>COSTWMK</b></p> <p>If WELFMILK = 2 <del>FREEITEM not eq 2</del>, calculate FWMLKVAL = 0.</p> <p>For each CHILD from CHILD record.</p> <p>If WELFMILK = 1 <del>FREEITEM=2</del>, calculate the value of free welfare milk</p> <p>calculate FWMLKVAL = WMKIT x <del>cost of free welfare milk (.34 for 1994/95 0.36 for 1995-96)</del> <b>COSTWMK</b></p> <p>If WELFMILK = 2 <del>FREEITEM not eq 2</del>, calculate FWMLKVAL = 0.</p>
FWMLKBU	<p>For each BENEFIT UNIT (for BENUNIT record)</p> <p>sum each occurrence of FWMLKVAL for each adult and child in benefit unit.</p>
FWMLKHH	<p>For each HOUSEHOLD (for HOUSEHOL record)</p> <p>sum each occurrence of FWMLKVAL for each adult and child in household.</p>
0	Not applicable - no welfare milk
-2	Unable to derive as any of above variables are missing.

**3 Results**

Tabulation will be required to show the number of adults/children, benefit units and households by the value of free welfare milk received sorted into the following bands

Under £1.00  
 £1.00 - £2.00  
 £2.00 - £3.00  
 £3.00 - £4.00  
 £4.00 - £5.00  
 £5.00 - £6.00  
 £6.00 or over



**GROSS**

Purpose	: This variable contains the FRS grossing factors. They allow analysts to weight FRS statistics so that they are representative of the national population.
Created	:
Database Table	: BENUNIT, HOUSEHOL
Minimum Value	:
Maximum Value	:
Units	:
Validations	:
Related Variables	:
Children	:
Parents	:
Core variable/user	: ASD3A
Amendments	: VE - 3 June 1997 - To correct a mistake in the control total table

**1 Definition**

This variable contains the FRS grossing factors. They allow analysts to weight FRS statistics so that they are representative of the national population. In doing so they attempt to correct for some of the differential rates of non-response by different types of household. Each household has a different grossing factor and each benefit unit within the household has the same grossing factor. However, **the grossing system can be used for analyses at both levels.**

**Examples**

To find the number of households with income greater than £x:

- (i) select out the head of households' benefit unit from the flat file
- (ii) count up how many of them have values for the household income variable HHINC greater than £x
- (iii) weight by GROSS956

To find the number of benefit units with income greater than £x:

- (i) count how many have values for the benefit unit income variable BUINC greater than £x
- (ii) weight by GROSS956

**Units**

The grossing factors on the FES were in thousands **but the FRS grossing factors will give actual population numbers.** The reason for changing this is to avoid confusion between grossed and ungrossed statistics in outputs from computer runs. For example, a sample count of 100 might gross up to about 90,000 on the FRS and if this latter is output as "90" it could be mistaken for a sample count. The smaller FES sample did not cause such problems.

~~In the interests of accuracy, the grossing factors have not been rounded to whole numbers. Hence, grossed counts will not necessarily be integers so that, for example, a count of 2,546.677 households would need to be rounded by the analyst to 2,547. They have been rounded this year as no real gain in accuracy from having a non-rounded figure and therefore it will be less confusing in this form.~~

**Derivation**

The grossing system represented by GROSS956 is just one of the many tested by ASD3A. Alternatives have not been added to the flatfile, to avoid confusion and promote consistency. However, it may be that for some specific tasks, analysts prefer to generate grossing factors which, for example, control to a different combination of variables. ASD3A can then advise on the use of software and on the statistical considerations in choosing a grossing regime.

To derive GROSS956, the sample is divided into different groups and the grossing factors are the ratio of population estimates to sample counts for those groups. The groups are designed to reflect differences in response rates among different types of households. They have also been chosen with the aims of DSS analyses in mind. The population estimates are based on control variables, with values derived from external data sources.

The control variables and their sources are listed below. The grossing system controls for variables at both household level and benefit unit level. A grossed count of the number of owner occupying households would thus tie in with the DoE figure, whilst the grossed number of single men under 35 would be consistent with the OPCS estimate. Some adjustments have been made to the original control total data sources so that definitions match those in the FRS, eg an adjustment has been made to the demographic data to exclude people not resident in private households.

<b>Control variables used to generate grossing factors</b>		
Variable	Groupings	Source of data
Age/sex/marital status	Single men: <35, 35-59, 60+ Single women: <35, 35-64, 65+ Couples: <65, 65+	OPCS, GAD
Lone parents	Male, female	ASD4E estimates
Families	No. of couples with children	ASD1 Child benefit data
Tenure type	LA renters, private renters, owner occupiers	DoE estimates
Council Tax Band	A, B, C-D, E-H	DoE estimates
Region	London, other	DoE estimates

In order to reconcile control variables at different levels and estimate their joint population software provided by the French national statistics institute INSEE has been used. This program, CALMAR, is a SAS macro and it works by iterating towards a solution. Options within CALMAR that give the solution which minimises the range of grossing factors have been used. This should maximise the potential precision of grossed estimates; if a few cases are associated with very small or very large grossing factors, grossed estimates will have relatively wide confidence intervals.

Careful consideration has been given to the combination of control totals and the way age ranges, CT Bands and so on have been grouped together. The aim has been to strike a balance so that the grossing system will provide, where possible, accurate estimates in different dimensions without significantly increasing variances. Further details are provided in Andrew Ray's Analytical Note 5, copies of which are held in ASD3.

### Control totals

The controls below relate to individuals, benefit units or households, depending on the category.

<b>Control Totals for GROSS</b>	
Single men <35	4,218,000
Single men 35-59	1,890,000
Single men 60+	1,142,000
Single women <35	3,164,000
Single women 35-64	2,321,000
Single women 65+	2,954,000
Couples <65	11,128,000
Couples 65+	2,643,000
Male lone parents	121,000
Female lone parents	1,527,000
Number of families with children	5,348,000
LA renters	<del>4,479,000</del> <b>4,470,000</b>
Private renters	3,302,000
Owner occupiers	15,665,000
CT Band A	6,348,000
CT Band B	4,719,000
CT Band C-D	8,219,000
CT Band E-H	4,151,000
Households in London	2,980,000
Households in other regions	20,201,000

**GROSSPAY**

Purpose : To show the total amount of earnings received by an adult from main and subsidiary jobs as an employee, excluding any income from odd jobs. This is the gross amount before any deductions for Income Tax, National Insurance, Trade Union dues etc.

Created : VC - 10 May 1993

Database Table : ADULT

Minimum Value : 0

Maximum Value :

Units : Real

Validations :

Related Variables :

Children :

Parents :

Core variable/user : ISM HBM **PSM** FCM (Bold indicates lead user)

Issue date : 21 May 2003

Amendments : VC - 24 June 1993 To remove references to odd jobs, baby sitting and mail order agents which are to be included in a net income variable.

: VC - 13 September 1993 Amended using guidelines provided by Ms Honey.

: VC - 15 February 1994 Amended to reflect version 30 changes

: VC - 28 February 1994 To exclude any records with a period code of 12 or 13

: JS - 20 February 1996 to allow skipped values where variables have been imputed

: JS - 21 March 1996 to correctly consider JOBTYP=1

: VE - 24 May 1996 - Initial updates for V32 - to amend for new category definitions for DEDUC

: SG - 3 October 1997 - If all jobs for adult are self-employed then GROSSPAY should be skipped

**1 Definition**

This variable is coded as

GROSSPAY The total gross earnings before deductions for Income Tax, NI etc from all jobs an adult may have as an employed earner, excluding any income from odd jobs.

-1 Not applicable where an adult does not have any jobs, **or all jobs are self-employed.**

-2 Unable to derive where any variables missing.

The variable GROSSPAY is derived from a variety of variables held in the ADULT and JOB tables and for each job held by that adult.

GROSSPAY is derived from the variable GRWAGE which holds the person's gross earnings before tax, NI etc but only where the payslip has been consulted (where PAYSLIP = 1).

If the payslip has not been consulted (where PAYSLIP = 2) the amount of net pay is obtained from PAYAMT. This variable holds the total amount of net pay after all deductions have been taken off and these deductions must be added back to PAYAMT to find GROSSPAY.

The amount of income tax deducted is found in PAYE and National Insurance in NATINS these must be found in all cases. Other deductions for example trade union fees, payments to charities etc are also to be added back to PAYAMT but may not be relevant to every case. Therefore, if CHRTAXF = 1 (indicating that the person has a deduction for charities) (see deducts for reason using this not charity variable) the amount held in AMTTAXF must be added to PAYAMT and if CHROTH = 1 (indicates that there is another deduction for a charity) the amount held in AMTOTH must be added to PAYAMT.

If any of OTHDED1 to OTHDED67 = 1 there will be a deduction for pension/superannuation, union fees, friendly societies, sports social clubs, **repayment of a loan from employer** and any other deductions with the amount held in the relevant DEDUC variable. DEDUC1 holds the amount for pension/superannuation, DEDUC2 holds the amount additional voluntary contributions, DEDUC3 holds the amount of union fees, DEDUC4 holds amount for friendly societies, DEDUC5 holds amount for sports clubs, **DEDUC6 holds the amount for the repayment of a loan from employer** and DEDOTH holds the amount for any other deduction not already mentioned and any occurrence of these must be added to PAYAMT.

The variables OTHDED1 to OTHDED67, DEDUC1 to DEDUC56 and DEDOTH are database variables collected from the questions OTHDED in the e-main block which asks were there any other deductions from your wage or salary and DEDUC which holds the amount of deduction in each case.

To get a person's gross earnings a check must be made to see if an income tax refund was included in PAYAMT. Therefore, if TAXINC = 1 the amount held in TAXAMT has to be deducted from PAYAMT to get a true amount of gross earnings.

However, if the period code for the benefit is 12 or 13 (lumpsum/one-off or other period) the record must be set to unable to derive as it has not been possible to convert the amount of benefit into a weekly amount during the database conversion process. Therefore, if PAYPD = 12 or 13 GROSSPAY is set to -2.

Where PAYSLIP has been imputed to yes, GRSWAGE will have been skipped. Similarly where PAYAMT is imputed, PAYPD and other variables may also be skipped. This has to be catered for in the DV specification

GROSSPAY is then the total amount of earnings from all of these sources for all jobs.

## 2 FRS Specification

For each ADULT, set GROSSPAY to zero.

GROSSPAY If WORKING=1 or JOBAWAY=1

From JOB table for every job a person has If EMPEE=1

If PAYPD equals -1 or 1-11 and

If PAYSLIP = 1, and GRSWAGE exists calculate GROSSPAY = amount in GRSWAGE.

Or if PAYSLIP = 2 or (PAYSLIP=1 and GRSWAGE=-1), calculate GROSSPAY as follows

GROSSPAY = sum of GROSSPAY, PAYAMT,

If PAYE exists add PAYE to GROSSPAY

If NATINS exists add NATINS to GROSSPAY

If AMTTAXF exists add AMTTAXF to GROSSPAY

If AMTOTH exists add AMTOTH to GROSSPAY

If DEDUC1 exists add DEDUC1 to GROSSPAY

If DEDUC2 exists add DEDUC2 to GROSSPAY

If DEDUC3 exists add DEDUC3 to GROSSPAY

If DEDUC4 exists add DEDUC4 to GROSSPAY

If DEDUC5 exists add DEDUC5 to GROSSPAY

**If DEDUC6 exists add DEDUC6 to GROSSPAY**

If DEDOTH exists add DEDOTH to GROSSPAY

Then if JOBTYP=1 and TAXAMT exists subtract TAXAMT from total GROSSPAY

GROSSPAY is then the total of all of the above for every job a person has.

(coding should now be the same as UGRSPAY, except does not include usual earnings)

-2If any of the above variables are missing or PAYPD = 12 or 13

### **3Results**

Tabulation is required to show the number of adults falling into the following bands of weekly earnings:

Under £50  
£50 - £100  
£100 - £150  
£150 - £200  
£200 - £250  
£250 - £300  
£300 - £350  
£350 - £400  
£400 - £450  
£450 - £500  
£500 and over

**GVTREGN**

Purpose : To indicate in which Government Office Region the interviewee lives.  
 Created : 27 August 1996  
 Database Table :  
 Minimum Value : 1  
 Maximum Value : 13  
 Units : Integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : ASD4A  
 Amendments : VE - 23 September 1996 - To include the change of name of the Government Office for Yorkshire and Humberside from 1 October 1995 to the Government Office for Yorkshire and Humber  
               : VE - 3 March 1997 - To include separate categories for the North West and for Merseyside in line with Regional Trends publications and to correct the order of GORs to make consistent with Regional Trends  
 Issue date : 21 May, 2003

**1 Definition**

This variable is coded as

1 North East  
~~2 North West and Merseyside~~  
**34** Yorkshire and ~~the~~ Humberside  
**45** East Midlands  
**56** West Midlands  
~~6~~**10** South West  
 7 Eastern  
 8 London  
 9 South East  
~~40~~**11** Wales  
~~44~~**12** Scotland  
**42** North West  
**43** Merseyside  
  
 -1 Not applicable to this case  
 -2 Unable to derive as variable LAC is missing

The government office region is derived from the Local Authority Code in the variable LAC. It should be noted that these regions are not exactly the same as the standard regions (STDREGN). Cumbria, which is included with the North in STDREGN, is in government office region 2 (North West and Merseyside). LACs which are in Bedfordshire, Essex and Hertfordshire are included with the Eastern LACs (GVTREGN=7), whereas they are included with the South East in STDREGN.

**2 FRS Specification**

Code each region according to the attached list.

-1 Not applicable to this case (shouldn't be any)

-2 Unable to derive as the variable LAC is missing.

NB - These LA codes are unique to FRS.

**3 Results**

Tabulation to show the number of households in each standard region

**4 Test Cases**

None



FAMILY RESOURCES SURVEY

DERIVED VARIABLE SPECIFICATION

STANDARD LAC LOCAL  
REGION CODE AUTHORITY

1	668	Hartlepool
1	772	Blyth Valley
1	771	Berwick-upon-Tweed
1	776	Gateshead
1	867	Newcastle upon Tyne
1	669	Langbaugh-on-Tees <b>Redcar and Cleveland</b>
1	674	Derwentside
1	672	Chester-le-Street
1	773	Castle Morpeth
1	673	Darlington
1	675	Durham
1	868	North Tyneside
1	676	Easington
1	670	Middlesbrough
1	768	Teesdale
1	774	Tynedale
1	870	Sunderland
1	671	Stockton-on-Tees
1	869	South Tyneside
1	767	Sedgefield
1	775	Wansbeck
1	770	Alnwick
1	769	Wear Valley

<del>2</del>	<del>573</del>	<del>Barrow-in-Furness</del>
<del>2</del>	<del>576</del>	<del>Eden</del>
<del>2</del>	<del>574</del>	<del>Carlisle</del>
<del>2</del>	<del>575</del>	<del>Copeland</del>
<del>2</del>	<del>667</del>	<del>South Lakeland</del>
<del>2</del>	<del>572</del>	<del>Allerdale</del>
<del>2</del>	<del>267</del>	<del>Crewe and Nantwich</del>
<del>2</del>	<del>474</del>	<del>South Ribble</del>
<del>2</del>	<del>373</del>	<del>Blackburn</del>
<del>2</del>	<del>467</del>	<del>Fylde</del>
<del>2</del>	<del>374</del>	<del>Blackpool</del>
<del>2</del>	<del>369</del>	<del>Stockport</del>
<del>2</del>	<del>475</del>	<del>West Lancashire</del>
<del>2</del>	<del>569</del>	<del>St Helens</del>
<del>2</del>	<del>476</del>	<del>Wyre</del>
<del>2</del>	<del>276</del>	<del>Oldham</del>
<del>2</del>	<del>567</del>	<del>Knowsley</del>
<del>2</del>	<del>469</del>	<del>Lancaster</del>
<del>2</del>	<del>272</del>	<del>Warrington</del>
<del>2</del>	<del>468</del>	<del>Hyndburn</del>
<del>2</del>	<del>470</del>	<del>Pendle</del>
<del>2</del>	<del>176</del>	<del>Congleton</del>
<del>2</del>	<del>269</del>	<del>Halton</del>
<del>2</del>	<del>370</del>	<del>Tameside</del>
<del>2</del>	<del>568</del>	<del>Liverpool</del>

**FAMILY RESOURCES SURVEY****DERIVED VARIABLE SPECIFICATION**

---

2	372	Wigan
2	368	Salford
2	274	Bury
2	375	Burnley
2	571	Wirral
2	473	Rossendale
2	471	Preston
2	268	Ellesmere Port and Neston
2	273	Bolton
2	376	Chorley
2	367	Rochdale
2	371	Trafford
2	175	Chester
2	270	Macclesfield
2	472	Ribble Valley
2	271	Vale Royal
2	275	Manchester
2	570	Sefton

---

34	972	East Yorkshire
34	969	East Yorks Borough of Beverley
34	971	Cleethorpes
34	174	Wakefield
34	167	Doncaster
34	068	Craven
34	973	Glanford
34	171	Calderdale
34	173	Leeds
34	067	Scunthorpe
34	073	Scarborough
34	071	Richmondshire
34	072	Ryedale
34	172	Kirklees
34	074	Selby
34	069	Hambleton
34	974	Great Grimsby
34	070	Harrogate
34	169	Sheffield
34	976	Kingston upon Hull
34	975	Holderness
34	168	Rotherham
34	075	York
34	170	Bradford
34	076	Barnsley
34	970	Boothferry
45	967	Nottingham
45	765	Daventry
45	562	Derby
45	657	Derbyshire Dales
45	866	Newark and Sherwood

---

---

45	858	Northampton
45	565	North East Derbyshire
45	764	Corby
45	664	North West Leicestershire
45	658	Blaby
45	665	Oadby and Wigston
45	660	Harborough
45	564	High Peak
45	766	East Northamptonshire
45	758	East Lindsey
45	563	Erewash
45	666	Rutland
45	968	Rushcliffe
45	861	Ashfield
45	757	Boston
45	560	Bolsover
45	864	Gedling
45	859	South Northamptonshire
45	762	South Kesteven
45	761	South Holland
45	566	South Derbyshire
45	760	North Kesteven
45	865	Mansfield
45	763	West Lindsey
45	860	Wellingborough
45	857	Kettering
45	759	Lincoln
45	862	Bassetlaw
45	662	Leicester
45	863	Broxtowe
45	659	Charnwood
45	663	Melton
45	661	Hinckley and Bosworth
45	561	Chesterfield
45	559	Amber Valley

---

56	357	The Wrekin
56	466	Solihull
56	265	Shrewsbury and Atcham
56	166	Malvern Hills
56	358	Cannock Chase
56	465	Sandwell
56	558	Wolverhampton
56	359	East Staffordshire
56	366	Tamworth
56	266	South Shropshire
56	261	Wyre Forest
56	363	Stafford
56	364	Staffordshire Moorlands
56	365	Stoke-on-Trent

---

**FAMILY RESOURCES SURVEY****DERIVED VARIABLE SPECIFICATION**

---

56	463	Coventry
56	362	South Staffordshire
56	165	Leominster
56	260	Wychavon
56	258	South Herefordshire
56	459	Rugby
56	360	Lichfield
56	464	Dudley
56	259	Worcester
56	460	Stratford-on-Avon
56	257	Redditch
56	262	Bridgnorth
56	164	Hereford
56	263	North Shropshire
56	557	Walsall
56	462	Birmingham
56	361	Newcastle-under-Lyme
56	461	Warwick
56	264	Oswestry
56	458	Nuneaton & Bedworth
56	457	North Warwickshire
56	163	Bromsgrove

---

610	157	South Somerset (aka Yeovil)
610	853	Plymouth
610	750	Wansdyke
610	059	Forest of Dean
610	962	Poole
610	855	Teignbridge
610	159	North Wiltshire
610	058	Cotswold
610	749	Northavon
610	060	Gloucester
610	065	Taunton Deane
610	756	Penwith
610	963	Purbeck
610	753	Carrick
610	966	East Dorset (aka Wimborne)
610	960	Christchurch
610	064	Sedgemoor
610	957	Torridge
610	856	Torbay
610	849	East Devon
610	959	Bournemouth
610	160	Salisbury
610	057	Cheltenham
610	062	Tewkesbury
610	850	Exeter
610	161	Thamesdown
610	847	Restormel

---

**FAMILY RESOURCES SURVEY**

**DERIVED VARIABLE SPECIFICATION**

610	854	South Hams
610	061	Stroud
610	162	West Wiltshire
610	158	Kennet
610	965	Weymouth and Portland
610	958	West Devon
610	754	Kerrier
610	751	Woodspring
610	748	Kingswood
610	752	Caradon
610	656	Bath
610	961	North Dorset
610	852	North Devon
610	755	North Cornwall
610	964	West Dorset
610	851	Mid Devon
610	066	West Somerset
610	063	Mendip
610	747	Bristol
7	248	South Cambridgeshire
7	154	East Cambridgeshire
7	348	Ipswich
7	155	Fenland
7	253	North Norfolk
7	153	Cambridge
7	347	Forest Heath
7	255	Kings Lynn and West Norfolk
7	250	Broadland
7	247	Peterborough
7	254	South Norfolk
7	352	Waveney
7	349	Mid Suffolk
7	351	Suffolk Coastal
7	256	Babergh
7	350	St Edmundsbury
7	251	Great Yarmouth
7	156	Huntingdonshire
7	252	Norwich
7	249	Breckland
7	539	Thurrock
7	146	Mid Bedfordshire
7	540	Uttlesford
7	742	Watford
7	145	Luton
7	743	Welwyn Hatfield
7	445	Maldon
7	437	Basildon
7	238	South Bedfordshire
7	446	Rochford
7	538	Tendring

**FAMILY RESOURCES SURVEY****DERIVED VARIABLE SPECIFICATION**

---

7	237	North Bedfordshire
7	740	Stevenage
7	537	Southend-on-Sea
7	738	North Hertfordshire
7	739	St Albans
7	741	Three Rivers
7	443	Epping Forest
7	737	Hertsmere
7	440	Castle Point
7	444	Harlow
7	441	Chelmsford
7	439	Brentwood
7	438	Braintree
7	442	Colchester
7	645	Dacorum
7	644	Broxbourne
7	646	East Hertfordshire
8	551	Bromley
8	554	Enfield
8	648	Hillingdon
8	548	Barnet
8	451	Lewisham
8	650	Kingston upon Thames
8	556	Harrow
8	653	Richmond upon Thames
8	356	Hammersmith and Fulham
8	549	Bexley
8	649	Hounslow
8	555	Greenwich
8	355	Hackney
8	547	Barking and Dagenham
8	651	Merton
8	647	Havering
8	447	Haringey
8	449	Kensington and Chelsea
8	448	Islington
8	452	Newham
8	652	Redbridge
8	450	Lambeth
8	553	Ealing
8	456	Westminster, City of
8	454	Tower Hamlets
8	552	Croydon
8	655	Waltham Forest
8	453	Southwark
8	455	Wandsworth
8	550	Brent
8	654	Sutton
8	354	Camden

---

8	353	City of London
<hr/>		
9	048	Mole Valley
9	948	Tonbridge and Malling
9	056	Adur
9	949	Tunbridge Wells
9	541	Basingstoke and Deane
9	337	Milton Keynes
9	953	Vale Of White Horse
9	055	Woking
9	243	Windsor and Maidenhead
9	244	Wokingham
9	344	Lewes
9	339	Wycombe
9	152	Worthing
9	842	Maidstone
9	643	Winchester
9	346	Wealden
9	054	Waverley
9	240	Newbury
9	744	Medina
9	954	West Oxfordshire
9	151	Mid Sussex
9	242	Slough
9	241	Reading
9	845	Shepway
9	639	Portsmouth
9	147	Arun
9	952	South Oxfordshire
9	049	Reigate and Banstead
9	844	Sevenoaks
9	345	Rother
9	640	Rushmoor
9	843	Rochester upon Medway
9	746	Ashford
9	245	Aylesbury Vale
9	951	Oxford
9	338	South Buckinghamshire
9	053	Tandridge
9	846	Swale
9	052	Surrey Heath
9	642	Test Valley
9	947	Thanet
9	050	Runnymede
9	641	Southampton
9	745	South Wight
9	051	Spelthorne
9	638	New Forest

---

**FAMILY RESOURCES SURVEY****DERIVED VARIABLE SPECIFICATION**

---

9	637	Havant
9	956	Epsom and Ewell
9	837	Canterbury
9	955	Elmbridge
9	543	Eastleigh
9	246	Chiltern
9	343	Hove
9	150	Horsham
9	544	Fareham
9	950	Cherwell
9	340	Brighton
9	841	Gravesham
9	545	Gosport
9	047	Guildford
9	840	Gillingham
9	342	Hastings
9	546	Hart
9	341	Eastbourne
9	148	Chichester
9	239	Bracknell Forest
9	542	East Hampshire
9	838	Dartford
9	149	Crawley
9	839	Dover

---

---

<del>4011</del>	179	Brecknock
<del>4011</del>	876	Wrexham Maelor
<del>4011</del>	082	Ynys Mon - Isle of Anglesey
<del>4011</del>	872	Colwyn
<del>4011</del>	977	Carmarthen
<del>4011</del>	182	Cardiff
<del>4011</del>	871	Alyn and Deeside
<del>4011</del>	178	Taff-Ely
<del>4011</del>	277	Swansea
<del>4011</del>	183	Vale of Glamorgan
<del>4011</del>	077	Torfaen
<del>4011</del>	978	Ceredigion
<del>4011</del>	078	Aberconwy
<del>4011</del>	982	South Pembrokeshire
<del>4011</del>	080	Dwyfor
<del>4011</del>	081	Meirionydd
<del>4011</del>	984	Islwyn
<del>4011</del>	981	Preseli
<del>4011</del>	181	Radnorshire

---



**FAMILY RESOURCES SURVEY****DERIVED VARIABLE SPECIFICATION**

---

4011	084	Merthyr Tydfil
4011	177	Rhymney Valley
4011	875	Rhuddlan
4011	086	Rhondda
4011	186	Neath
4011	184	Port Talbot (aka Afan)
4011	979	Dinefwr
4011	085	Ogwr
4011	983	Blaenau Gwent
4011	874	Glyndwr
4011	079	Arfon
4011	985	Monmouth
4011	180	Montgomeryshire
4011	873	Delyn
4011	986	Newport
4011	980	Llanelli
4011	083	Cynon Valley
4011	185	Lliw Valley

---

4412	481	Nairn
4412	685	Motherwell
4412	386	Moray
4412	381	North East Fife
4412	286	Nithsdale
4412	679	Hamilton
4412	378	Wigton
4412	278	Berwickshire
4412	480	Lochaber
4412	385	Kincardine and Deeside
4412	380	Kirkcaldy
4412	580	Bearsden & Milngavie
4412	682	Kyle and Carrick
4412	683	Lanark (aka Clydesdale)
4412	681	Kilmarnock and Loudoun
4412	479	Inverness
4412	684	Monklands
4412	783	Western Isles Islands Area
4412	577	Midlothian
4412	382	Aberdeen City
4412	478	Caithness
4412	680	Inverclyde
4412	578	West Lothian
4412	777	Strathkelvin
4412	779	Dundee City
4412	585	Dumbarton
4412	778	Angus
4412	379	Dunfermline
4412	581	Clydebank / Clydesdale (see Lanark)
4412	782	Shetland Islands Area
4412	483	Skye and Lochalsh

---

**FAMILY RESOURCES SURVEY**

**DERIVED VARIABLE SPECIFICATION**

4412	579	Argyll and Bute
4412	584	Cunninghame
4412	284	Stirling
4412	285	Annandale and Eskdale
4412	484	Sutherland
4412	377	Stewartry
4412	583	Cumnock and Doon Valley
4412	582	Cumbernauld and Kilsyth
4412	282	Clackmannan
4412	485	East Lothian
4412	686	Renfrew
4412	279	Ettrick and Lauderdale
4412	283	Falkirk
4412	780	Perth and Kinross
4412	678	Glasgow City
4412	781	Orkney Islands Area
4412	482	Ross and Cromarty
4412	281	Tweeddale
4412	477	Badenoch and Strathspey
4412	586	East Kilbride
4412	383	Banff and Buchan
4412	677	Eastwood
4412	486	Edinburgh City
4412	280	Roxburgh
4412	384	Gordon
42	573	<b>Barrow-in-Furness</b>
42	576	<b>Eden</b>
42	574	<b>Carlisle</b>
42	575	<b>Copeland</b>
42	667	<b>South Lakeland</b>
42	572	<b>Allerdale</b>
42	267	<b>Crewe and Nantwich</b>
42	474	<b>South Ribble</b>
42	373	<b>Blackburn</b>
42	467	<b>Fylde</b>
42	374	<b>Blackpool</b>
42	369	<b>Stockport</b>
42	475	<b>West Lancashire</b>
42	476	<b>Wyre</b>
42	276	<b>Oldham</b>
42	469	<b>Lancaster</b>
42	272	<b>Warrington</b>
42	468	<b>Hyndburn</b>
42	470	<b>Pendle</b>
42	176	<b>Congleton</b>
42	269	<b>Halton</b>
42	370	<b>Tameside</b>
42	372	<b>Wigan</b>
42	368	<b>Salford</b>
42	274	<b>Bury</b>

42	375	Burnley
42	473	Rossendale
42	471	Preston
42	268	Ellesmere Port and Neston
42	273	Bolton
42	376	Chorley
42	367	Rochdale
42	371	Trafford
42	175	Chester
42	270	Macclesfield
42	472	Ribble Valley
42	271	Vale Royal
42	275	Manchester
43	570	Sefton
43	569	St Helens
43	571	Wirral
43	568	Liverpool
43	567	Knowsley

---

**HBINDBU**

Purpose : To indicate if any person in the benefit unit receives Income Support, Housing Benefit or Council Tax Benefit.

Created : JS: 19 January 1996

Database Table : BENUNIT

Minimum Value : 0

Maximum Value : 7

Units : integer

Validations :

Related Variables : HBINDHH

Children :

Parents :

Core variable/user : ASD6A

Amendments : BS - 3 August 1995. Amended to take into account changes to V31 of the Questionnaire.  
 : JS - 19 January 1996: to include additional codes for IS receipt  
 : JS - 25 January: to make clear that codes for IS refer to benefit units where any adult is in receipt of IS  
 : VE - 4 June 1996 - No initial amendments needed for V32 update

Issued : 21 May 2003

**Assumptions**

Council tax information is collected only once, any rebate is therefore assumed to be for the householder benefit unit

**1 Definition**

This variable is coded as

HBINDBU	Indicates that someone in the benefit unit is receiving Income Support, Housing Benefit or Council Tax Benefit as follows,
0	No HB/CTB/IS
1	If receives CTB only (no IS or HB)
2	If receives HB only (no IS or CTB)
3	If receives IS only (no HB or CTB)
4	If receives both HB and CTB (no IS)
5	If receives both HB and IS (no CTB)
6	If receives both CTB and IS (no HB)
7	If receives HB, CTB and IS

-1 Not applicable to this case: these should not occur since questions are asked of all households

-2 Unable to derive because of missing values

This variable is derived from questions in the HOUSEHOL and ADULT records

If the householder benefit unit receives Council Tax Benefit (CTB), this is shown by the CTREB = 1 (was any CTB allowed in connection with your last CT payment - 1 = yes, 2 = no).

Receipt of Housing Benefit is identified from four separate questions depending on the type of household:

- i HBENEFIT = 1 (has received HB in connection with last rent payment) (household record)  
There is no need to check whether rebate=1 because this question is only asked if HBENEFIT is set to 1
- ii CVHB=1 (adult record: lodgers in BUs>1 in receipt of HB) where HHSTAT=1 (conventional household)
- iii SCVHB=1 (adult record: adults in BUs>1 in receipt of HB) where HHSTAT=2 (shared household)

IS receipt is identified where any person in the benefit unit answers "yes" to BEN3Q02 (ADULT record - are you at present receiving IS)

**2 FRS Specification**

For each benefit unit in the household

Code            Condition

HBINDBU

- 0 If BENUNIT=1 (and CTREB=2) or BENUNIT>1 (CTB not in receipt) and (BENUNIT=1 and HBENEFIT=2) (HB not in receipt for first BU)
  - or if BENUNIT>1, for each adult where CONVBL=1 or 2, CVHB=2 (HB not in receipt for boarders or lodgers)
  - or if BENUNIT>1 and HHSTAT=2, for each adult, SCVHB=2 (HB not in receipt for adults in shared household benefit units)
 and for each adult in benefit unit, BEN3Q02<>1 (IS not in receipt)
- 1 If BENUNIT=1 (and CTREB=1) (CTB in receipt: only possible for householder benefit unit) and (BENUNIT=1 and HBENEFIT=2) (HB not in receipt for first BU)
  - or if BENUNIT>1, for each adult where CONVBL=1 or 2, CVHB=2 (HB not in receipt for boarders or lodgers)
  - or if BENUNIT>1 and HHSTAT=2, for each adult, SCVHB=2 (HB not in receipt for adults in shared household benefit units)
 and for each adult in benefit unit, BEN3Q02<>1 (IS not in receipt)
- 2 If BENUNIT=1 (and CTREB=2) or BENUNIT>1 (CTB not in receipt) and (BENUNIT=1 and HBENEFIT=1) (HB in receipt for first BU)
  - or if BENUNIT>1, for at least one adult where CONVBL=1 or 2, CVHB=1 (HB in receipt for boarders or lodgers)

- 
- or if BENUNIT>1 and HHSTAT=2, for at least one adult SCVHB=1 (HB in receipt for adults in shared household benefit units)  
and for each adult in benefit unit, BEN3Q02<>1 (IS not in receipt)
- 3 If BENUNIT=1 (and CTREB=2) or BENUNIT>1 (CTB not in receipt)  
and (BENUNIT=1 and HBENEFIT=2) (HB not in receipt for first BU)  
or if BENUNIT>1, for each adult where CONVBL=1 or 2, CVHB=2 (HB not in receipt for boarders or lodgers)  
or if BENUNIT>1 and HHSTAT=2, for each adult, SCVHB=2 (HB not in receipt for adults in shared household benefit units)  
and for at least one adult in benefit unit, BEN3Q02=1 (IS in receipt)
- 4 If BENUNIT=1 (and CTREB=1) (CTB in receipt: only possible for householder benefit unit)  
and (BENUNIT=1 and HBENEFIT=1) (HB in receipt for first BU)  
or if BENUNIT>1, for at least one adult where CONVBL=1 or 2, CVHB=1 (HB in receipt for boarders or lodgers)  
or if BENUNIT>1 and HHSTAT=2, for at least one adult, SCVHB=1 (HB in receipt for adults in shared household benefit units)  
and for each adult in benefit unit, BEN3Q02<>1 (IS not in receipt)
- 5 If BENUNIT=1 (and CTREB=2) or BENUNIT>1 (CTB not in receipt)  
and (BENUNIT=1 and HBENEFIT=1) (HB in receipt for first BU)  
or if BENUNIT>1, for at least one adult where CONVBL=1 or 2, CVHB=1 (HB in receipt for boarders or lodgers)  
or if BENUNIT>1 and HHSTAT=2, for at least one adult, SCVHB=1 (HB in receipt for adults in shared household benefit units)  
and for at least one adult in benefit unit, BEN3Q02=1 (IS in receipt)
- 6 If BENUNIT=1 (and CTREB=1) (CTB in receipt: only possible for householder benefit unit)  
and (BENUNIT=1 and HBENEFIT=2) (HB not in receipt for first BU)  
or if BENUNIT>1, for each adult where CONVBL=1 or 2, CVHB=2 (HB not in receipt for boarders or lodgers)  
or if BENUNIT>1 and HHSTAT=2, for each adult, SCVHB=2 (HB not in receipt for adults in shared household benefit units)  
and for at least one adult in benefit unit, BEN3Q02=1 (IS in receipt)
- 7 If BENUNIT=1 (and CTREB=1) (CTB in receipt: only possible for householder benefit unit)  
and (BENUNIT=1 and HBENEFIT=1) (HB in receipt for first BU)  
and for at least one adult in benefit unit, BEN3Q02=1 (IS in receipt)
- 1 Not applicable to this case
- 2 Unable to derive as any of the above are missing.

**HBINDHH**

Purpose : To indicate if any person in the household receives Income Support, Housing Benefit or Council Tax Benefit.

Created : VC - 1 September 1993

Database Table : HOUSEHOL

Minimum Value : 0

Maximum Value : 7

Units : integer

Validations :

Related Variables : ISRECDHH FCRECDHH  
Children :  
Parents :

Core variable/user : ASD6A

Amendments : BS - 3 August 1995. Amended to take into account changes to V31 of the Questionnaire.  
: JS - 19 January 1996: to include additional codes for IS receipt  
: JS - 25 January 1996: to make completely clear that codes for IS refer to households where any adult is in receipt of IS  
: VE - 4 June 1996 - No initial amendments needed for V32 update

Issued : 21 May 2003

**1 Definition**

This variable is coded as

HBINDHH	Indicates that someone in the household is receiving Income Support, Housing Benefit or Council Tax Benefit as follows,
0	No HB/CTB/IS
1	If receives CTB only (no IS or HB)
2	If receives HB only (no IS or CTB)
3	If receives IS only (no HB or CTB)
4	If receives both HB and CTB (no IS)
5	If receives both HB and IS (no CTB)
6	If receives both CTB and IS (no HB)
7	If receives HB, CTB and IS
-1	Not applicable to this case - not expected because questions are asked of all households
-2	Unable to derive because of missing values

This variable is derived from questions in the HOUSEHOL and ADULT records

If the household receives Council Tax Benefit (CTB), this is shown by the CTREB = 1 (was any CTB allowed in connection with your last CT payment - 1 = yes, 2 = no).

Receipt of Housing Benefit is identified from four separate questions depending on the type of household:

- i HBENEFIT = 1 (has received HB in connection with last rent payment) (household record).
  - ii CVHB=1 (adult record: lodgers in BUs>1 in receipt of HB) where HHSTAT=1 (conventional household)
- iii SCVHB=1 (adult record: adults in BUs>1 in receipt of HB) where HHSTAT=2 (shared household)

IS receipt is identified where any person in the household answers "yes" to BEN3Q02 (ADULT record - are you at present receiving IS)

## 2 FRS Specification

Code            Condition

HBINDHH

- 0      If CTREB=2  
and HBENEFIT=2  
        or for each relevant adult in the household (in BUs>1 where CONVBL=1 or 2), CVHB=2  
        or for each relevant adult in the household (in BUs>1 and HHSTAT=2), SCVHB=2  
        and for each adult in household, BEN3Q02<>1
- 1      If CTREB=1  
and HBENEFIT=2  
        or for each relevant adult in the household (in BUs>1 where CONVBL=1 or 2) CVHB=2  
        or for each relevant adult in the household (in BUs>1 and HHSTAT=2) SCVHB=2  
        and for each adult in household, BEN3Q02<>1
- 2      If CTREB=2  
and HBENEFIT=1  
        or for at least one relevant adult in the household (in BUs>1 where CONVBL=1 or 2)  
        CVHB=1  
        or for at least one relevant adult in the household (in BUs>1 and HHSTAT=2) where  
        SCVHB=1  
        and for each adult in household, BEN3Q02<>1
- 3      If CTREB=2  
and HBENEFIT=2  
        or for each relevant adult in the household (in BUs>1 where CONVBL=1 or 2) CVHB=2  
        or for each relevant adult in the household (in BUs>1 and HHSTAT=2) SCVHB=2  
        and for at least one adult in household, BEN3Q02=1
- 4      If CTREB=1  
and HBENEFIT=1



- or for at least one relevant adult in the household (in BUs>1 where CONVBL=1 or 2)  
CVHB=1
  - or for at least one relevant adult in the household (in BUs>1 and HHSTAT=2)  
SCVHB=1
  - and for each adult in household, BEN3Q02<>1
  
- 5 If CTREB=2  
and HBENEFIT=1
  - or for at least one relevant adult in the household (in BUs>1 where CONVBL=1 or 2)  
CVHB=1
  - or for at least one relevant adult in the household (in BUs>1 and HHSTAT=2)  
SCVHB=1
  - and for at least one adult in household, BEN3Q02=1
  
- 6 If CTREB=1  
and HBENEFIT=2
  - or for each relevant adult in the household (in BUs>1 where CONVBL=1 or 2), CVHB=2
  - or for each relevant adult in the household (in BUs>1 and HHSTAT=2) SCVHB=2
  - and for at least one adult in household, BEN3Q02=1
  
- 7 If CTREB=1  
and HBENEFIT=1
  - or for at least one relevant adult in the household (in BUs>1 where CONVBL=1 or 2)  
CVHB=1
  - or for at least one relevant adult in the household (in BUs>1 and HHSTAT=2)  
SCVHB=1
  - and for at least one adult in household, BEN3Q02=1
  
- 1 Not applicable to this case
  
- 2 Unable to derive as any of the above are missing.

**HBSUPRAN**

Purpose : To indicate the total amount of superannuation or pension contributions :  
deducted from a person's earnings from all jobs, excluding any additional voluntary contributions  
(AVCs).  
Created : January 1996  
Database Table : ADULT  
Minimum Value : 0  
Maximum Value :  
Units : Real  
Validations :  
Related Variables : Deducts,superann  
Children :  
Parents :  
Core variable/user : HBM  
Issue date : 21 May 2003  
Amendments : JS - 21 February 1996 to allow for skipped values when variables have been  
imputed  
: VE - 23 May 1996 - No initial amendments needed for V32 update

NB - will not include self-employed jobs separate base variables provided for these jobs

**1 Definition**

This variable is coded as

- HBSUPRAN The total amount of superannuation or pension contributions deducted from a person's earnings from all jobs.
- 1 Not applicable in this case - people who do not have superannuation deductions and those not working or self-employed
- 2 Unable to derive variable.

The amount of superannuation or pension contributions is derived from all jobs and where OTHDED1 are coded 1 to show that an amount for a pension or superannuation is deducted. The variable DEDUC1 will then hold the amount of superannuation/pension contribution. The variables OTHDED1 and DEDUC1 are created in the database to hold the answers to the multi response questions OTHDED (were there any other deductions from your wage/salary such as 1 = pension or superannuation, 3 = Union fees etc) and DEDUC which holds the amounts.

However, if the period code for the deduction is 12 or 13 (lumpsum/one-off or other period), from the PAYPD variable, the record must be set to unable to derive as it has not been possible to convert the amount of benefit into a weekly amount during the database conversion process. Therefore, if PAYPD = 12 or 13 HBSUPRAN is set to -2.

**2 FRS Specification**

For each ADULT with record Job for all jobs

<u>Code</u>	<u>Condition</u>
HBSUPRAN	If PAYPD equals -1 or 1-11 and -  If OTHDED1 = 1 and DEDUC1 exists, get the amount of the superannuation or pension contribution deducted from variable DEDUC1 else leave as skipped.
-1	Not applicable in this case - where OTHDED1 = 2 or 3 or OTHDED1 does not exist or there are no job records
-2	If variable cannot be defined because of missing data - where there is a job record but no values or if PAYPD = 12 OR 13.

**HDAGE**

Purpose :To create a variable for use in hotdecking which shows the age range in which a respondent falls  
Created :19 June 1996  
Database Table : ADULT  
Minimum Value : 1  
Maximum Value : 6  
Units : Integers  
Validations :  
Related Variables :  
    Children : HDAGECH  
    Parents :  
Core variable/user : Hotdecking  
Amendments :  
Issue date :21 May 2003

**1 Definition**

This variable is coded as

- 1 Age 16 to 24
- 2 Age 25 to 34
- 3 Age 35 to 44
- 4 Age 45 to 54
- 5 Age 55 to 64
- 6 Age 65 and over
- 2 Unable to derive due to missing values

HDAGE is derived from the AGE variable in the ADULT table. It shows the age range in which a respondent falls, and is used to specify classes for hotdecking.

**2 FRS Specification**

For each adult

Code    Condition

- 1      From ADULT table  
        If AGE>=16 and AGE<=24
- 2      From ADULT table

- If AGE $\geq$ 25 and AGE $\leq$ 34
- 3 From ADULT table
- If AGE $\geq$ 35 and AGE $\leq$ 44
- 4 From ADULT table
- If AGE $\geq$ 45 and AGE $\leq$ 54
- 5 From ADULT table
- If AGE $\geq$ 55 and AGE $\leq$ 64
- 6 From ADULT table
- If AGE $\geq$ 65
- 2 Unable to derive due to missing values - There should be no missing values for AGE

**HDAGECH**

Purpose :To create a variable for use in hotdecking which shows the age range in which a respondent falls  
Created :19 June 1996  
Database Table : CHILD  
Minimum Value : 1  
Maximum Value : 4  
Units : Integers  
Validations :  
Related Variables :  
    Children : HDAGE  
    Parents :  
Core variable/user : Hotdecking  
Amendments :  
Issue date :21 May 2003

**1 Definition**

This variable is coded as

- 1 Age 0 to 4
- 2 Age 5 to 9
- 3 Age 10 to 14
- 4 Age 15 and over
- 2 Unable to derive due to missing values

HDAGE is derived from the AGE variable in the CHILD table. It shows the age range in which a respondent falls, and is used to specify classes for hotdecking.

**2 FRS Specification**

For each child

Code    Condition

- 1      From CHILD table  
        If AGE<=4
- 2      From CHILD table  
        If AGE>=5 and AGE<=9
- 3      From CHILD table

If AGE $\geq$ 10 and AGE $\leq$ 14

4 From CHILD table

If AGE $\geq$ 15

-2 Unable to derive due to missing values - There should be no missing values for AGE

**HDBEN**

Purpose :To create a variable for use in hotdecking which shows whether any income related benefits are received by an individual

Created :19 June 1996

Database Table : ADULT

Minimum Value : 1

Maximum Value : 2

Units : Integers

Validations :

Related Variables : INIRBEN  
Children :  
Parents :

Core variable/user : Hotdecking

Amendments

Issue date : 21 May, 2003

**1 Definition**

This variable is coded as

- 1 Income related benefits received by individual
- 2 No income related benefits received by individual
- 2 Unable to derive due to missing values

HDBEN is derived from the INIRBEN derived variable in the ADULT table. This shows the total amount of income received each week by individuals from income related benefits. If INIRBEN is greater than zero, an individual is classified as receiving income related benefits. If INIRBEN is zero, an individual is classified as not receiving income related benefits.

**2 FRS Specification**

For each adult

Code   Condition

- 1 From ADULT table  
If INIRBEN $\geq$ 0
- 2 From ADULT table  
If INIRBEN=0
- 2 Unable to derive due to missing values - There should be no missing values for INIRBEN



**HDBORR**

Purpose :To create a variable for use in hotdecking which shows the range in which BORRAMT falls  
Created :19 June 1996  
Database Table : MORTGAGE  
Minimum Value : 0  
Maximum Value : 14  
Units : Real  
Validations :  
Related Variables :  
    Children :  
    Parents :  
Core variable/user : Hotdecking  
Amendments :  
Issue date :21 May 2003

**1 Definition**

This variable is coded as

- 1 BORRAMT 0 to 9999.99
- 2 BORRAMT 10000 to 19999.99
- 3 BORRAMT 20000 to 29999.99
- 4 BORRAMT 30000 to 39999.99
- 5 BORRAMT 40000 to 49999.99
- 6 BORRAMT 50000 to 59999.99
- 7 BORRAMT 60000 to 69999.99
- 8 BORRAMT 70000 to 79999.99
- 9 BORRAMT 80000 to 89999.99
- 10 BORRAMT 90000 to 99999.99
- 11 BORRAMT 100000 to 124999.99
- 12 BORRAMT 125000 to 149999.99
- 13 BORRAMT 150000 to 199999.99
- 14 BORRAMT 200000 and over
- 1 Not applicable to this case

-2 Unable to derive due to missing values

HDBORR is derived from the BORRAMT variable in the MORTGAGE table. It shows the range in which the amount of the mortgage falls, and is used to specify classes for hotdecking.

## 2 FRS Specification

For each mortgage

Code   Condition

1	From MORTGAGE table If BORRAMT <=9999.99
2	From MORTGAGE table If BORRAMT >=10000 and BORRAMT <=19999.99
3	From MORTGAGE table If BORRAMT >=20000 and BORRAMT <=29999.99
4	From MORTGAGE table If BORRAMT >=30000 and BORRAMT <=39999.99
5	From MORTGAGE table If BORRAMT >=40000 and BORRAMT <=49999.99
6	From MORTGAGE table If BORRAMT >=50000 and BORRAMT <=59999.99
7	From MORTGAGE table If BORRAMT >=60000 and BORRAMT <=69999.99
8	From MORTGAGE table If BORRAMT >=70000 and BORRAMT <=79999.99
9	From MORTGAGE table If BORRAMT >=80000 and BORRAMT <=89999.99
10	From MORTGAGE table If BORRAMT >=90000 and BORRAMT <=99999.99

- 11 From MORTGAGE table  
If BORRAMT $\geq$ 100000 and BORRAMT $\leq$ 124999.99
- 12 From MORTGAGE table  
If BORRAMT $\geq$ 125000 and BORRAMT $\leq$ 149999.99
- 13 From MORTGAGE table  
If BORRAMT $\geq$ 150000 and BORRAMT $\leq$ 199999.99
- 14 From MORTGAGE table  
If BORRAMT $\geq$ 200000
- 1 Not applicable to this case - where there is no mortgage
- 2 Unable to derive due to missing values

**HDGRWAG**

Purpose :To create a variable for use in hotdecking which shows the range in which GRWAGE falls  
Created :19 June 1996  
Database Table : JOB  
Minimum Value : 0  
Maximum Value : 15  
Units : Real  
Validations :  
Related Variables :  
    Children :  
    Parents :  
Core variable/user : Hotdecking  
Amendments :  
Issue date :21 May 2003

**1 Definition**

This variable is coded as

- 1 GRWAGE 0 to 49.99
- 2 GRWAGE 50 to 99.99
- 3 GRWAGE 100 to 149.99
- 4 GRWAGE 150 to 199.99
- 5 GRWAGE 200 to 249.99
- 6 GRWAGE 250 to 299.99
- 7 GRWAGE 300 to 349.99
- 8 GRWAGE 350 to 399.99
- 9 GRWAGE 400 to 499.99
- 10 GRWAGE 500 to 599.99
- 11 GRWAGE 600 to 699.99
- 12 GRWAGE 700 to 799.99
- 13 GRWAGE 800 to 899.99
- 14 GRWAGE 900 to 999.99
- 15 GRWAGE 1000 and over

- 1 Not applicable to this case
- 2 Unable to derive due to missing values

HDGRWAG is derived from the GRWAGE variable in the JOB table. It shows the range in which gross wage/salary falls, and is used to specify classes for hotdecking.

## 2 FRS Specification

For each job

<u>Code</u>	<u>Condition</u>
1	From JOB table If GRWAGE <=49.99
2	From JOB table If GRWAGE >=50 and GRWAGE <=99.99
3	From JOB table If GRWAGE >=100 and GRWAGE <=149.99
4	From JOB table If GRWAGE >=150 and GRWAGE <=199.99
5	From JOB table If GRWAGE >=200 and GRWAGE <=249.99
6	From JOB table If GRWAGE >=250 and GRWAGE <=299.99
7	From JOB table If GRWAGE >=300 and GRWAGE <=349.99
8	From JOB table If GRWAGE >=350 and GRWAGE <=399.99
9	From JOB table If GRWAGE >=400 and GRWAGE <=499.99
10	From JOB table

- If GRWAGE $\geq$ 500 and GRWAGE $\leq$ 599.99
- 11 From JOB table
- If GRWAGE $\geq$ 600 and GRWAGE $\leq$ 699.99
- 12 From JOB table
- If GRWAGE $\geq$ 700 and GRWAGE $\leq$ 799.99
- 13 From JOB table
- If GRWAGE $\geq$ 800 and GRWAGE $\leq$ 899.99
- 14 From JOB table
- If GRWAGE $\geq$ 900 and GRWAGE $\leq$ 999.99
- 15 From JOB table
- If GRWAGE $\geq$ 1000
- 1 Not applicable to this case
- 2 Unable to derive due to missing values

**HDPAY**

Purpose :To create a variable for use in hotdecking which shows the range in which  
PAYAMT falls  
Created :19 June 1996  
Database Table : JOB  
Minimum Value : 0  
Maximum Value : 15  
Units : Real  
Validations :  
Related Variables :  
Children :  
Parents :  
Core variable/user : Hotdecking  
Amendments :  
Issue date :21 May 2003

**1 Definition**

This variable is coded as

- 1 PAYAMT 0 to 49.99
- 2 PAYAMT 50 to 99.99
- 3 PAYAMT 100 to 149.99
- 4 PAYAMT 150 to 199.99
- 5 PAYAMT 200 to 249.99
- 6 PAYAMT 250 to 299.99
- 7 PAYAMT 300 to 349.99
- 8 PAYAMT 350 to 399.99
- 9 PAYAMT 400 to 499.99
- 10 PAYAMT 500 to 599.99
- 11 PAYAMT 600 to 699.99
- 12 PAYAMT 700 to 799.99
- 13 PAYAMT 800 to 899.99
- 14 PAYAMT 900 to 999.99
- 15 PAYAMT 1000 and over

- 1 Not applicable to this case
- 2 Unable to derive due to missing values

HDPAY is derived from the PAYAMT variable in the JOB table. It shows the range in which take-home pay falls, and is used to specify classes for hotdecking.

## 2 FRS Specification

For each job

<u>Code</u>	<u>Condition</u>
1	From JOB table If PAYAMT <=49.99
2	From JOB table If PAYAMT >=50 and PAYAMT <=99.99
3	From JOB table If PAYAMT >=100 and PAYAMT <=149.99
4	From JOB table If PAYAMT >=150 and PAYAMT <=199.99
5	From JOB table If PAYAMT >=200 and PAYAMT <=249.99
6	From JOB table If PAYAMT >=250 and PAYAMT <=299.99
7	From JOB table If PAYAMT >=300 and PAYAMT <=349.99
8	From JOB table If PAYAMT >=350 and PAYAMT <=399.99
9	From JOB table If PAYAMT >=400 and PAYAMT <=499.99
10	From JOB table



- If PAYAMT >= 500 and PAYAMT <= 599.99
- 11 From JOB table
- If PAYAMT >= 600 and PAYAMT <= 699.99
- 12 From JOB table
- If PAYAMT >= 700 and PAYAMT <= 799.99
- 13 From JOB table
- If PAYAMT >= 800 and PAYAMT <= 899.99
- 14 From JOB table
- If PAYAMT >= 900 and PAYAMT <= 999.99
- 15 From JOB table
- If PAYAMT >= 1000
- 1 Not applicable to this case
- 2 Unable to derive due to missing values

**HDPEN**

Purpose :To create a variable for use in hotdecking which shows the range in which  
PENPAY falls  
Created :19 June 1996  
Database Table : PENSION  
Minimum Value : 0  
Maximum Value : 11  
Units : Real  
Validations :  
Related Variables :  
Children :  
Parents :  
Core variable/user : Hotdecking  
Amendments :  
Issue date :21 May 2003

**1 Definition**

This variable is coded as

- 1 PENPAY 0 to 14.99
- 2 PENPAY 15 to 29.99
- 3 PENPAY 30 to 49.99
- 4 PENPAY 50 to 69.99
- 5 PENPAY 70 to 99.99
- 6 PENPAY 100 to 149.99
- 7 PENPAY 150 to 199.99
- 8 PENPAY 200 to 299.99
- 9 PENPAY 300 to 399.99
- 10 PENPAY 400 to 499.99
- 11 PENPAY 500 and over
- 1 Not applicable to this case
- 2 Unable to derive due to missing values

HDPEN is derived from the PENPAY variable in the PENSION table. It shows the range in which the last payment of a respondent's pension falls, and is used to specify classes for hotdecking.

**2 FRS Specification**

For each pension

<u>Code</u>	<u>Condition</u>
1	From PENSION table If PENPAY<=14.99
2	From PENSION table If PENPAY>=15 and PENPAY<=29.99
3	From PENSION table If PENPAY>=30 and PENPAY<=49.99
4	From PENSION table If PENPAY>=50 and PENPAY<=69.99
5	From PENSION table If PENPAY>=70 and PENPAY<=99.99
6	From PENSION table If PENPAY>=100 and PENPAY<=149.99
7	From PENSION table If PENPAY>=150 and PENPAY<=199.99
8	From PENSION table If PENPAY>=200 and PENPAY<=299.99
9	From PENSION table If PENPAY>=300 and PENPAY<=399.99
10	From PENSION table If PENPAY>=400 and PENPAY<=499.99
11	From PENSION table If PENPAY>=500
-1	Not applicable to this case

-2      Unable to derive due to missing values

**HDPURC**

Purpose :To create a variable for use in hotdecking which shows the range in which PURCMT falls  
Created :19 June 1996  
Database Table : MORTGAGE  
Minimum Value : 0  
Maximum Value : 14  
Units : Real  
Validations :  
Related Variables :  
    Children :  
    Parents :  
Core variable/user : Hotdecking  
Amendments :  
Issue date :21 May 2003

**1 Definition**

This variable is coded as

- 1 PURCMT 0 to 9999.99
- 2 PURCMT 10000 to 19999.99
- 3 PURCMT 20000 to 29999.99
- 4 PURCMT 30000 to 39999.99
- 5 PURCMT 40000 to 49999.99
- 6 PURCMT 50000 to 59999.99
- 7 PURCMT 60000 to 69999.99
- 8 PURCMT 70000 to 79999.99
- 9 PURCMT 80000 to 89999.99
- 10 PURCMT 90000 to 99999.99
- 11 PURCMT 100000 to 124999.99
- 12 PURCMT 125000 to 149999.99
- 13 PURCMT 150000 to 199999.99
- 14 PURCMT 200000 and over
- 1 Not applicable to this case

-2 Unable to derive due to missing values

HDBORR is derived from the PURCAMT variable in the MORTGAGE table. It shows the range in which the amount of the purchase price of the house/flat falls, and is used to specify classes for hotdecking.

## 2 FRS Specification

For each mortgage

Code Condition

1	From MORTGAGE table If PURCAMT <=9999.99
2	From MORTGAGE table If PURCAMT >=10000 and PURCAMT <=19999.99
3	From MORTGAGE table If PURCAMT >=20000 and PURCAMT <=29999.99
4	From MORTGAGE table If PURCAMT >=30000 and PURCAMT <=39999.99
5	From MORTGAGE table If PURCAMT >=40000 and PURCAMT <=49999.99
6	From MORTGAGE table If PURCAMT >=50000 and PURCAMT <=59999.99
7	From MORTGAGE table If PURCAMT >=60000 and PURCAMT <=69999.99
8	From MORTGAGE table If PURCAMT >=70000 and PURCAMT <=79999.99
9	From MORTGAGE table If PURCAMT >=80000 and PURCAMT <=89999.99
10	From MORTGAGE table If PURCAMT >=90000 and PURCAMT <=99999.99

- 11 From MORTGAGE table  
If PURCAMT $\geq$ 100000 and PURCAMT $\leq$ 124999.99
- 12 From MORTGAGE table  
If PURCAMT $\geq$ 125000 and PURCAMT $\leq$ 149999.99
- 13 From MORTGAGE table  
If PURCAMT $\geq$ 150000 and PURCAMT $\leq$ 199999.99
- 14 From MORTGAGE table  
If PURCAMT $\geq$ 200000
- 1 Not applicable to this case - where there is no mortgage
- 2 Unable to derive due to missing values

**HDQHRS**

Purpose :To create a variable for use in hotdecking which shows the range in which the contracted hours worked by a respondent as an employee falls

Created :19 June 1996

Database Table : JOB

Minimum Value : 1

Maximum Value : 6

Units : Real

Validations :

Related Variables :

    Children :

    Parents :

Core variable/user : Hotdecking

Amendments

Issue date :21 May 2003

**1 Definition**

This variable is coded as

- 1 QHRS 0 to 15.99
- 2 QHRS 16 to 29.99
- 3 QHRS 30 to 39.99
- 4 QHRS 40 to 49.99
- 5 QHRS 50 to 59.99
- 6 QHRS 60 or more
- 1 Not applicable to this case
- 2 Unable to derive due to missing values

HDQHRS is derived from the QHRS variable in the JOB table. It shows the range in which the number of contracted hours as an employee a respondent falls, and is used to specify classes for hotdecking.

**2 FRS Specification**

For each job

Code    Condition

- 1        From JOB table
- If QHRS<16



- 2 From JOB table  
If QHRS $\geq$ 16 and QHRS $<$ 30
- 3 From JOB table  
If QHRS $\geq$ 30 and QHRS $<$ 40
- 4 From JOB table  
If QHRS $\geq$ 40 and QHRS $<$ 50
- 5 From JOB table  
If QHRS $\geq$ 50 and QHRS $<$ 60
- 6 From JOB table  
If QHRS $\geq$ 60
- 1 Not applicable to this case
- 2 Unable to derive due to missing values

**HDQHRSSSE**

Purpose :To create a variable for use in hotdecking which shows the range in which the number of hours usually worked per week by a respondent in self-employment falls

Created :19 June 1996

Database Table : JOB

Minimum Value : 1

Maximum Value : 6

Units : Real

Validations :

Related Variables :

    Children :

    Parents :

Core variable/user : Hotdecking

Amendments

Issue date :21 May 2003

**1 Definition**

This variable is coded as

- 1 QHRSSSELF 0 to 15.99
- 2 QHRSSSELF 16 to 29.99
- 3 QHRSSSELF 30 to 39.99
- 4 QHRSSSELF 40 to 49.99
- 5 QHRSSSELF 50 to 59.99
- 6 QHRSSSELF 60 or more
- 1 Not applicable to this case
- 2 Unable to derive due to missing values

HDQHRSSSE is derived from the QHRSSSELF variable in the JOB table. It shows the range in which the number of hours usually worked by a respondent in self-employment falls, and is used to specify classes for hotdecking.

**2 FRS Specification**

For each self-employed job (EMPEE=2)

Code    Condition

- 1        From JOB table

- If QHRSSSELF<=16
- 2 From JOB table
- If QHRSSSELF>=16 and QHRSSSELF<30
- 3 From JOB table
- If QHRSSSELF>=30 and QHRSSSELF<40
- 4 From JOB table
- If QHRSSSELF>=40 and QHRSSSELF<50
- 5 From JOB table
- If QHRSSSELF>=50 and QHRSSSELF<60
- 6 From JOB table
- If QHRSSSELF>=60
- 1 Not applicable to this case
- 2 Unable to derive due to missing values

**HDUNETT**

Purpose :To create a variable for use in hotdecking which shows the range in which UNETT falls  
Created :19 June 1996  
Database Table : JOB  
Minimum Value : 0  
Maximum Value : 15  
Units : Real  
Validations :  
Related Variables :  
    Children :  
    Parents :  
Core variable/user : Hotdecking  
Amendments :  
Issue date :21 May 2003

**1 Definition**

This variable is coded as

- 1 UNETT 0 to 49.99
- 2 UNETT 50 to 99.99
- 3 UNETT 100 to 149.99
- 4 UNETT 150 to 199.99
- 5 UNETT 200 to 249.99
- 6 UNETT 250 to 299.99
- 7 UNETT 300 to 349.99
- 8 UNETT 350 to 399.99
- 9 UNETT 400 to 499.99
- 10 UNETT 500 to 599.99
- 11 UNETT 600 to 699.99
- 12 UNETT 700 to 799.99
- 13 UNETT 800 to 899.99
- 14 UNETT 900 to 999.99
- 15 UNETT 1000 and over

- 1 Not applicable to this case
- 2 Unable to derive due to missing values

HDPAY is derived from the UNETT variable in the JOB table. It shows the range in which usual net pay falls, and is used to specify classes for hotdecking.

## 2 FRS Specification

For each job

<u>Code</u>	<u>Condition</u>
1	From JOB table If UNETT <=49.99
2	From JOB table If UNETT >=50 and UNETT <=99.99
3	From JOB table If UNETT >=100 and UNETT <=149.99
4	From JOB table If UNETT >=150 and UNETT <=199.99
5	From JOB table If UNETT >=200 and UNETT <=249.99
6	From JOB table If UNETT >=250 and UNETT <=299.99
7	From JOB table If UNETT >=300 and UNETT <=349.99
8	From JOB table If UNETT >=350 and UNETT <=399.99
9	From JOB table If UNETT >=400 and UNETT <=499.99
10	From JOB table

- If UNETT >= 500 and UNETT <= 599.99
- 11 From JOB table
- If UNETT >= 600 and UNETT <= 699.99
- 12 From JOB table
- If UNETT >= 700 and UNETT <= 799.99
- 13 From JOB table
- If UNETT >= 800 and UNETT <= 899.99
- 14 From JOB table
- If UNETT >= 900 and UNETT <= 999.99
- 15 From JOB table
- If UNETT >= 1000
- 1 Not applicable to this case
- 2 Unable to derive due to missing values

**HHCOMP**

Purpose : To indicate household composition for use in the FRS publication.  
 Created : 29 January 1996  
 Database Table : HOUSEHOL  
 Minimum Value : 1  
 Maximum Value : 21  
 Units : Integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : Publication  
 Issue date : 21 May 2003  
 Amendments : JS - 14 March to collapse some of the households with children categories  
               : 4 June 1996 - No initial amendments needed for V32 update

HHCOMP categorises households on the basis of whether *dependent* children are present, number of adults and whether adults are over or under state retirement age. Some categories may yield very few cases. These will be combined in the final publication.

Using state retirement age is consistent with FAMTYPBU categories (although these only relate to whether the **head** is under/over pension age) but not ECSTATBU, which considers whether one or more adult is aged over 60.

As well as the DV, HHCOMP tables include a number of subtotals relating to pensioners, the sick and disabled and the unemployed (including work related government training schemes). Definitions of cases to be included in these categories are also given below. These categories may be overlapping.

**1 Definition**

HHCOMP is derived by looking at variables DEPCHLDH (number of dependent children in HH), ADULTH (number of adults in HH), PERSON (person number), AGE and SEX.

Additional subtotals used in tables are identified by age, EMPSTATB and responses to questions HEALTH, HPROB, RSTRCT, LAREG and JCREG.

The definition of sick and disabled is consistent with the proposed follow up study of the disabled, **except** that it excludes receipt of disability benefits. This will help ensure that figures are comparable over time (rules for benefits may change).

The coding for HHCOMP is:

- 1 One adult, no children over pension age
- 2 One adult, no children, under pension age
- 3 Two adults, no children, both over pension age
- 4 Two adults, no children, one over pension age
- 5 Two adults, no children, both under pension age
- 6 Three or more adults, no children
- 7 One adult, one child
- 8 One adult, two children

- 9 One adult, three or more children
- 10 Two adults, one child
- 11 Two adults, two children
- 12 Two adults, three or more children
- 13 Three or more adults, one child
- 14 Three or more adults, two children
- 15 Three or more adults, three or more children

The sub totals required are:

- A Total households without children
- B Total households with children
- C Total households with one or more adults over pension age including the head
- D Total households with one or more adults over pension age excluding the head
- E Households with one or more sick/disabled adults under pension age including the head
- F Households with one or more sick/disabled adults under pension age excluding the head
- G Households with one or more unemployed adults under pension age including the head
- H Households with one or more unemployed adults under pension age excluding the head



**2 FRS Specification**

*Households without children, codes 1-6 all where DEPCHLDH=0*

- 1 ADULTH=1 and (SEX=1 and AGE>=65) or (SEX=2 and AGE>=60)
- 2 ADULTH=1 and (SEX=1 and AGE<65) or (SEX=2 and AGE<60)
- 3 ADULTH=2 and (PERSON=1 and ((SEX=1 and AGE>=65) or (SEX=2 and AGE>=60)))  
and (PERSON=2 and ((SEX=1 and AGE>=65) or (SEX=2 and AGE>=60)))
- 4 ADULTH=2 and (PERSON=1 and ((SEX=1 and AGE>=65) or (SEX=2 and AGE>=60)))  
and (PERSON=2 and ((SEX=1 and AGE<65) or (SEX=2 and AGE<60)))  
or (PERSON=1 and ((SEX=1 and AGE<65) or (SEX=2 and AGE<60)))  
and (PERSON=2 and ((SEX=1 and AGE>=65) or (SEX=2 and AGE>=60)))
- 5 ADULTH=2 and (PERSON=1 and ((SEX=1 and AGE<65) or (SEX=2 and AGE<60)))  
and (PERSON=2 and ((SEX=1 and AGE<65) or (SEX=2 and AGE<60)))
- 6 ADULTH>=3
- 7 ADULTH=1 and DEPCHLDH=1
- 8 ADULTH=1 and DEPCHLDH=2
- 9 ADULTH=1 and DEPCHLDH >=3
- 10 ADULTH=2 and DEPCHLDH=1
- 11 ADULTH=2 and DEPCHLDH=2
- 12 ADULTH=2 and DEPCHLDH>=3
- 13 ADULTH>=3 and DEPCHLDH=1
- 14 ADULTH>=3 and DEPCHLDH=2
- 15 ADULTH>=3 and DEPCHLDH >=3

The subtotals are calculated as:

- A Cases where HHCOMP=1-6
- B Cases where HHCOMP=7-21
- C Cases where PERSON 1 is over pension age:  
(SEX=1 and AGE>=65) or (SEX=2 and AGE>=60))
- D Cases where PERSON 1 is under pension age and at least one other adult is over pension age
- E Cases where PERSON 1 is under pension age and has a long standing illness or restricted in what they do ie:  
  
PERSON=1 and ((SEX=1 and AGE<65) or (SEX=2 and AGE<60))  
and ((HEALTH=1 and HPROB=1) or (RSTRCT=1 or RSTRCT=2) or (LAREG=1) or (JCREG=1))
- F Where PERSON 1 is not restricted in what they do, but that one or more other adults in the household meet the criteria:  
  
(PERSON=1 and (HEALTH=2 and RSTRCT=3))  
and for all other adults in HH at least one is:  
((SEX=1 and AGE<65) or (SEX=2 and AGE<60))  
and ((HEALTH=1 and HPROB=1) or (RSTRCT=1 or RSTRCT=2) or (LAREG=1) or (JCREG=1))
- Note: HPROB, LAREG and JCREG are only asked if HEALTH=1, so (HEALTH=2 and RSTRCT=3) should be sufficient to identify adults who are **not** sick/disabled*
- G Where for PERSON 1, EMPSTATB=7 (unemployed) or EMPSTATB=8 (govt training scheme)
- H Where for PERSON 1, EMPSTATB<>7 and EMPSTATB<>8 and at least one other adult in the household has (EMPSTATB=7 or EMPSTAT=8)

**HHINC**

Purpose : To show the total amount of income received by each household for use in the FRS publication.  
 Created : 2 February 1996  
 Database Table : HOUSEHOL  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables :  
     Children :  
     Parents : BUINC  
 Core variable/user : ASD4A  
 Amendments : VE - 4 June 1996 - No initial amendments needed for V32 update

**1 Definition**

This variable is coded as

HHINC      The total amount of income received each week by all members of the benefit unit.  
 0            No income is received by the household  
 -2           Unable to derive due to missing values

This specification, also sets up the component DVs used in the publication. These are:

HEARNS      earned income  
 HSEINC      self-employment income  
 HHINV       investment income  
 HHRPINC     retirement pension plus any income support  
 HPENINC     other pension income  
 HHDISBEN    disability benefits  
 HHOTHBEN    other benefits  
 HHRINC      remaining income

In addition, there are two other OPCS variables included in this specification

HBENINC     Household benefit income  
 HOTHINC     Other household income

**2 FRS Specification**

For each household

<u>Code</u>	<u>Condition</u>
HHINC	From BENUNIT table get BUEARNS, BSEINC, BUINV, BURPINC, BPENINC, BUDISBEN, BUOTHBEN, BURINC and BUINC for all benefit units in the household

-2 HHINC equals total occurrences of BUINC  
If any components are missing

**Component variables are calculated as follows:**

HEARNNS equals total occurrences of BUEARNNS

HSEINC equals total occurrences of BSEINC

HHINV equals total occurrences of BUINV

HHRPINC equals total occurrences of BUPRINC

HPENINC equals total occurrences of BPENINC

HHDISBEN equals total occurrences of BUDISBEN

HHOTHBEN equals total occurrences of BUOTHBEN

HHRINC equals total occurrences of BURINC

HBENINC = HHRPINC+HHDISBEN+HHOTHBEN  
HOTHINC=HHINV+HHRINC

components are set to -2 if any variables are missing

**HHRENT**

Purpose : To show the rent eligible for Housing Benefit paid by a household for accommodation before the deduction of Housing Benefit but after taking off extras such as service charges.

Created : 26 January 1993

Database Table : HOUSEHOL

Minimum Value : 0

Maximum Value :

Units : Real

Validations :

Related Variables :

    Children : BURENT, **WATSEWRT**

    Parents :

Core variable/user : **HBM PSM (BOLD indicates lead user)**

Issue date : 21 May, 2003

Amendments : VC - 26 April 1993. To divide GRSRENT by benefit unit not by household.  
: VC - 18 May 1993. To produce separate household and benefit unit gross :  
rent variables  
: VC - 9 June 1993. To include rent free weeks.  
: AJG - 21 June 1993. If 100% rebate received, then HHRENT = housing :  
benefit amount.  
: VC - 15 February 1994. Amendments due to version 30 update.  
: VC - 1 March 1994 To exclude period codes 12 and 13.  
: BS - 2 August 1995 To include changes to questionnaire for V31.  
: JS - 21 February 1996 to allow skipped values where variables have been  
imputed and to make CWATAMTD change explicit  
: JS - 27 March - to amend values included in rent for rent holiday cases  
: VE - 10 June 1996 - Amendments for V32 - COMMINC no longer asked,  
WORSINC replaces WATERINC and SEWERINC, section relating to  
CWATAMTD removed  
: VE - 27 August 1996 - To add SCHBAMT for shared household cases as a  
result of discussions with Martin Uglow and Graham Farrant (SCPR). It is  
assumed from this date that SRENTAMT is the figure for rent after HB has  
been deducted (ie a net amount).  
: VE - 28 August 1996 - To deduct service charges for cases in receipt of HB  
where a statement is consulted.  
: VE - 29 November 1996 - To amend so that contributions to rent are only  
added back to current rent when the person paying is not the DSS (all DSS  
payments relate to rent arrears). Also to amend so that the amount can be  
added when RENT=0, ie ACCCHK=-1.  
: VE - 12 December 1996 - to change HBSERAMT to HBSERA01,  
HBSERA02 etc.  
: VE - 3 February 1997 - to amend so that rent free cases are included to pick  
up rent paid by outsiders  
: SG - 18 July 1997 - correct check for deducting WATSEWRT from  
HHRENT - previously missing certain cases

**1 Definition**

This variable is coded as

- HHRENT This is the total amount of rent eligible for HB paid by a household, before the deduction of any Housing Benefit but after taking off certain expenses such as service charges, council tax etc which are included in the rent.
- 1 Not applicable to this case.
- 2 Unable to derive because of missing values.

HHRENT is derived from the variables RENT (rent actually paid for household), ~~CWATAMT (amount of community water charge), COMMINC (any council tax included in rent y/n), COMMANT (amount of council tax included),~~ HBENEFIT (housing benefit receipt y/n), HBENCHK (HB included before/after original amount in rent), HBENAMT (amount of HB), ACCNONHH (anyone outside HH paying rent for you y/n), ACCCHK (before or after original amount), ACCAMT (amount paid by other person), SERINC (rent include any services) and SERVAMT (amount of services).

First omit owner-occupiers, boarders **and** lodgers ~~and rent free~~. Note rent-free cases have been checked to make sure not including 100% HB cases.

In shared households (HHSTAT=2) we need to aggregate individual payments (RENT, SRENTAMT) before checking for rent holidays etc.

Next the weekly amount of rent has to be checked as the household may have some rent free weeks (RENTHOL = 1) and the total rent has to be apportioned over this period. The rent should be multiplied by the number of weeks over which it is actually paid (52 weeks minus the number of rent free weeks (WEEKHOL) and then divided by 52 for the weekly amount.

If someone pays part of the rent on behalf of the tenant and this has not been included in the original amount for rent declared, this must be added to the declared rent. (Not including HB at this point). **This is only for non-DSS contributors to the rent.**

(Reordered)

Other variables ask whether ~~council tax, community water rates and~~ certain service charges are included in the rent amount stated which should be deducted in full from the gross rent amount. (These amounts have been converted to a weekly amount using period codes)

For HB cases (non 100%), if Housing Benefit is received and the original amount of rent is said to be after Housing Benefit has been deducted, ie where HBENCHK = 2 (after), the Housing Benefit has to be added back on to the rent. HHRENT is then the result of the total amount of rent paid less any ineligible services (council tax, community water rates or any other service charges), plus any HB. In case (confused?) responses have led to HHRENT being negative a check is applied and HHRENT set to zero before HB is added.

A further check makes sure that HHRENT is not less than HB.

If the household pays no rent because of 100% HB ie where REBATE = 1 (receive 100% HB) the amount of HB should be substituted for the rent from hbenamt.

If the period code for any of the above variables is 12 or 13 (lumpsum/one-off or other period) the record must be set to unable to derive as it has not been possible to convert the amount of benefit into a weekly amount during the database conversion process. Therefore, if RENTPD, COMPD, HBENPD, ACCPD or ~~CWATAMT~~ = 12 or 13 HHRENT is set to -2. Since 94/5 all code 12 are edited so this problem should be reduced.

**2 FRS Specification**

For each household

<u>Code</u>	<u>Condition</u>
HHRENT	<p>From HOUSEHOL table, if TENURE = 2 or 3 or 4 or 5 <b>or 6</b> (renting <b>or rent free</b>) get HHRENT and HHSTAT variables <del>{Drop tenure=6 rent free have non-zero}</del></p> <p>From RENTER table, if RENTPD equals -1 or 1-11 get RENT variable and calculate HHRENT = RENT</p> <p>From HOUSEHOL record if HHSTAT = 2 then from ADULT record for all BUs &gt; 1 if SRENTPD equals -1 or 1-11 and SRENTAMT exists add SRENTAMT to HHRENT.</p> <p>If RENTHOL = 1 and WEEKHOL exists (has some rent free weeks) multiply HHRENT by the number of weeks actually paid (52 - WEEKHOL) and divide the result by 52 to get the overall weekly rent else do not change HHRENT.</p> <p>If RENTHOL = 1 and WEEKHOL exists and HBENEFIT=1(HB in receipt), adjust HBENAMT as above, ie multiply HBENAMT by (52-WEEKHOL) and divide by 52. (Note: Because HB is only paid for the weeks rent is paid)</p> <p>From RENTCONT record if ACCNONHH = 1 and <b>ACCPAY ≠ 1 (2, 3, 4 or 5)</b> and ACCCHK = 2 ≠ 1 (<b>2 or -1</b>) and ACCPD equals -1 or 1-11, add ACCAMT to rent to get HHRENT (Someone outside of HH pays part of rent and original rent given after this deducted. <b>ACCCHK=-1 refers to those cases where RENT=0, in which case ACCCHK is not asked</b>). If ACCNONHH = 1 and ACCCHK = 1 or ACCNONHH = 2, do not change amount in HHRENT.</p> <p><del>If CWATAMTD exists and RENTHOL=1 and WEEKHOL exists adjust CWATAMTD as above and deduct adjusted CWATAMTD from HHRENT to get HHRENT (community water charge). Else if CWATAMTD exists and no rent holiday deduct from HHRENT. If not, do not change HHRENT.</del></p> <p><del>If COMMINC = 1 and COMPD equals -1 or 1-11, and RENTHOL=1 and WEEKHOL exists adjust COMMAMT as above and deduct adjusted COMMAMT to get HHRENT (council tax). Else if COMMINC=1 and COMPD equals -1 or 1-11 and no rent holiday deduct COMMAMT to get HHRENT. If COMMINC = 2, do not change HHRENT.</del></p> <p>If SERINC = 1 and RENTHOL=1 and WEEKHOL exists adjust SERVAMT as above and , deduct adjusted SERVAMT to get HHRENT (other services). Else if SERINC=1 and no rent holiday, deduct SERVAMT to get HHRENT. If SERINC = 2, do not change HHRENT.</p> <p>If <del>WATERINC = 1 or SEWERINC</del> <b>WORSINC = 1 or HBWSAMT&gt;0 (statement exists and is amount to deduct) or WSINCAMT&gt;0 (no statement and is amount to deduct)</b> , and RENTHOL=1 and WEEKHOL exists adjust WATSEWRT as above and deduct WATSEWRT from HHRENT; else if no rent holiday deduct unadjusted</p>

WATSEWRT from HHRENT. If WATERINC=2 and SEWERINC WORSINC=2, do not change HHRENT.

(At this stage possible that HHRENT is negative if confusion over inclusive charges. This is most likely for HB cases where rent has been declared after HB deducted). So check

If HHRENT<0 then HHRENT=0

If HBENEFIT = 1 and HBENCHK = 2 or HBENCHK=-1 and HBENPD equals -1 or 1-11, add HBENAMT to HHRENT (HB rec'd and adjusted for rent holidays where applicable & original rent given after HB and ineligible services deducted).

If HBENEFIT = 1 and HBENCHK = 1  
If HBENAMT>HHRENT then HHRENT=HBENAMT

Otherwise if HBENEFIT = 2, do not change amount in HHRENT.

If REBATE = 1 and HBENPD equals -1 or 1-11, HHRENT = HBENAMT. In this circumstance the declared rent will have been zero.

From HOUSEHOL record if HHSTAT = 2 then from ADULT record for all BUs > 1 if SCHBPD equals -1 or 1-11 and SCHBAMT exists add SCHBAMT to HHRENT.

~~If HBENEFIT = 1 and HBSTMT = 1 and WORSINC = 1 and RENTHOL = 1 and WEEKHOL exists, adjust WATSEWRT as above and deduct WATSEWRT from HHRENT; else if no rent holiday deduct unadjusted WATSEWRT from HHRENT. If WORSINC = 2, do not change HHRENT.~~

If HBENEFIT = 1 and HBSTMT = 1 and HBSERAMT 01> 0 or HBSERA02>0 or HBSERA03>0 or HBSERA04>0 or HBSERA05>0 or HBSERA06>0 or HBSERA07>0 or HBSERA08>0 or HBSERA09>0 or HBSERA10>0 or HBSERA11>0 or HBSOAMT >0 and RENTHOL = 1 and WEEKHOL exists, adjust HBSERAMT and HBSOAMT as above and deduct HBSERAMT and HBSOAMT from HHRENT; else if no rent holiday deduct unadjusted HBSERAMT and HBSOAMT from HHRENT.



**HHSIZE**

Purpose : To show the number of people in the household for publication purposes  
Created : 5 November 1996  
Database Table : HOUSEHOL  
Minimum Value : 1  
Maximum Value : 7  
Units : Integers  
Validations :  
Related Variables :  
    Children :  
    Parents :  
Core variable/user : ASD6E (Publication)  
Amendments :  
Issue date : 21 May, 2003

**1 Definition**

This variable is coded as

1 1 person  
2 2 people  
3 3 people  
4 4 people  
5 5 people  
6 6 people  
7 7+ people

-1 Not applicable to this case

-2 Unable to derive due to missing values.

HHSIZE gives the size each household, for use in publication tables.

**2 FRS Specification**

Code   Condition

For each household:

SIZE = ADULTH + DEPCHLDH

1     If SIZE = 1

2     If SIZE = 2

3     If SIZE = 3

4     If SIZE = 4

5 If SIZE = 5

6 If SIZE = 6

7 If SIZE >=7

-1 Not applicable for this case. There should be no instances where this occurs.

-2 Unable to derive due to missing values. There should be no cases where this occurs.

**HHSTATUS**

Purpose : Householder Status of Benefit Unit  
 Created : AJG 1 December 1992  
 Database Table : BENUNIT  
 Minimum Value : 1  
 Maximum Value : 3  
 Units : Integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : ISM  
 Amendments : VC - 5 February 1993  
               : VC - 15 February 1994 Amended to reflect version 30 changes  
               : VE - 4 June 1996 - No initial amendments needed for V32 update  
 Issue date : 5 June 1996

**1 Definition**

Identifies whether the Benefit Unit is the head of household Benefit Unit.

This variable is coded as

- 1 Single Benefit Unit household.
- 2 Multi Benefit Unit household, head of household Benefit Unit
- 3 Multi Benefit Unit household, not head of household Benefit Unit
- 2 Unable to derive.

The assumption is made that the first Benefit Unit is the head of household Benefit Unit.

**2 FRS Specification**

For Benefit Unit in each household

CODE CONDITION

Get variables BENUNIT and BENUNITS from BENUNIT and HOUSEHOL tables respectively.

- 1 If BENUNIT = 1 and BENUNITS = 1
- 2 If BENUNIT= 1 and BENUNITS > 1
- 3 If BENUNIT > 1
- 2 If unable to derive variable.

NB - BENUNIT indicates which benefit unit within the household and BENUNITS shows the total number of benefit units within that household.

**3 Results**

Tabulate numbers falling into each category.

**4 Test Cases**

Households:

- A 1 Benefit Unit
- B 2 Benefit Units
- C 3 Benefit Units

**HPERSON**

Purpose : To show the person number within the Household  
Created : AJG 10 September 1993  
Database Table : ADULT and CHILD  
Minimum Value : 1  
Maximum Value : 20  
Units : Integer  
Validations :  
Related Variables :  
    Children :  
    Parents :  
Core variable/user: HBAI  
Amendments : VE - 23 May 1996 - No amendments needed for initial V32 update  
Issued : 21 May 2003

**1 Definition**

This variable assigns the value 1 to the first person in the first Benefit Unit in the Household and increments by one for each adult and each child, by Benefit Unit.

**2 FRS Specification**

Process each Benefit Unit in the household in turn, incrementing HPERSON as shown.

Code    Condition

1        If BENUNIT = 1 and PERSON = 1  
+1       In the following priority:  
+1       If BENUNIT = 1, and PERSON = 2,  
+1       If BENUNIT = 1, for each dependent in descending order of age,  
          For each subsequent Benefit Unit:  
+1           For each adult in PERSON number order,  
+1           For each dependent in descending order of age.

**3 Results**

What tabulation should be produced to check the results?

**HSCOSTHH**

Purpose : Housing costs paid by a household.  
 Created : VC - 24 September 1993  
 Database Table : HOUSEHOL  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables :  
     Children : INCAHCHH INCAHCBU HHCOSTBU  
     Parents : HHRENT WATSEWRT MORTINT  
 Core variable/user : Publication  
 Amendments : VC - 12 October 1993 To emphasize that this is by household  
               : VC - 16 February 1994 Amended to reflect changes to version 30  
               : VC - 1 March 1994 To exclude period codes 12 and 13  
               : JS - 6 February 1996 to amend calculation of structural insurance and change  
               names of period codes  
               : JS - to allow for skipped values where variables have been imputed and to  
               make CWATAMTD change explicit  
               : VE - 5 June 1996 - No initial amendments needed for V32 update  
               : VE - 2 September 1996 - To adjust CWATAMTD and WSEWRT for rent free  
               holidays  
               : VE - 20 November 1996 - To remove the rent holiday adjustment to  
               CWATAMTD (because it is not included in rent, and doesn't feed into  
               HHRENT)  
               : VE - 20 November 1996 - To amend STRAMT to STRAMT1 and STRAMT2  
               where appropriate  
               : VE - 3 February 1997 - To bring in line with HBAI by assuming that 2/3 of  
               combined contents and structural insurance relates to structure  
               : VE - 6 February 1997 - To reference household charge variables from the  
               HOUSEHOL table rather than the OWNER table  
               : VE - 10 March 1997 - To amend the structural insurance part of the spec  
               : SG - 7 October 1997 - Set unable to derive if missing values  
 Issued : 21 May 2003

**1 Definition**

This variable is coded as

HSCOSTHH This is the total amount spent on housing costs by each household regardless of whether they are in rented or owned accommodation.

0 Not applicable as have no housing costs

-2 Unable to derive due to missing values.

The variable is produced from the total amount of rent paid after any charges for TVs etc, the amount of any mortgage interest paid, the amount of water and sewerage paid, any insurance on the structure of the accommodation, any community water charge per household and any other regular housing payments (eg ground rent, feu duty etc).

The gross rent is calculated by the derived variable HHRENT which produces the household's gross rent after taking off any expenses such as community charge or service charges for lifts, TV etc which might have been included in the rent.

Mortgage interest is found in the derived variable MORTINT which calculates the total amount of mortgage interest paid for all mortgages that have been taken out to buy the property.

The amount of water and sewerage rates which have to be paid by the household are also found in another derived variable - WATSEWRT. This collects both water and sewerage rates for the whole household. **WATSEWRT should be adjusted for rent free holidays before adding into HSCOSTHH.**

Insurance on the structure of the property, on the other hand, must be derived within this variable. The variables to do this are to be found in the OWNER record. The amount of insurance paid on the structure is held in a variable called STRINTH regardless of how the interviewee pays the insurance premium. Therefore, if STRINTH exists it should be added to HSCOSTHH. STRINTH needs to be amended to include structural insurance premiums paid separately from mortgage payments. **For combined contents and structural insurance, it is assumed that 2/3 of the insurance relates to structure.**

The amount of community water charge should ~~be adjusted for rent free holidays and~~ also be added into HSCOSTHH and is found in the DV CWATAMTD.

Finally any other housing costs have to added into HSCOSTHH. These are collected from the question CHARGE which asks if the household pays ground rent, feu duty, chief rent, service charge, any compulsory maintenance charge, site rent (for caravans) or any other not mentioned. These are in turn put into the database under the following variables - CHARGE1, CHARGE2, CHARGE3, CHARGE4, CHARGE5, CHARGE6 and CHARGE7 respectively. If the answer to any of the above is yes, the amount of each charge is held in the database variables CHAMT1, CHAMT2, CHAMT3, CHAMT4, CHAMT5, CHAMT6 and CHAMT7.

However, if the period code for **charges** is 12 or 13 (lumpsum/one-off or other period) the record must be set to unable to derive as it has not been possible to convert the amount of benefit into a weekly amount during the database conversion process. Therefore, if **CHARGEPI** = 12 or 13 HSCOSTHH is set to -2. However, a value of -1 (skipped) is acceptable, since this implies that the amount has been imputed. As the other parts of HSCOSTHH are derived using previously derived variables the period codes need not be looked at for these variables as they are already excluded.

## 2 FRS Specification

**If any of components missing (refused or don't know) then set to unable to derive, also if period codes set to 12 or 13.**

For each HOUSEHOLD, set HSCOSTHH to zero.

<u>Code</u>	<u>Condition</u>
-------------	------------------

HSCOSTHH	From HOUSEHOL and RENTER records,
----------	-----------------------------------

	If HHRENT exists <u>and</u> does not equal -1 or -2 (not applicable or unable to derive) add the amount in HHRENT into HSCOSTHH.
--	--

If **RENTHOL=1** and **WEEKHOL** exists and **WATSEWRT** exists and does not equal -1 or -2 (as above) **multiply WATSEWRT by (52-WEEKHOL) and divide by 52 to get the overall weekly rent, else do not change WATSEWRT.** Add the amount in **WATSEWRT** into **HSCOSTHH**.

If **MORTINT** exists and does not equal -1 or -2 (as above) add the amount in **MORTINT** into **HSCOSTHH**.

From **HOUSEHOL** record get **STRCOV**, **STRAMT1**, **STRAMT2**, **STRPD1**, **STRPD2**, **STROTHS**, **COVOTHS**, **TENURE**

set **STRINTH** equal to zero

If **STRCOV=1** ~~or **STRCOV=3**~~ and **STRAMT1** exists (structural insurance only ~~or combined with furniture and contents as part of mortgage payment~~) and **STRPD1=-1** or 1-11 **STRINTH=STRAMT1**

**If **STRCOV=3** and **STRAMT1** exists (structural insurance combined with furniture and contents as part of mortgage payment) and **STRPD1=-1** or 1-11 **STRINTH=STRAMT1\*0.6666** (to bring in line with HBAI)**

If **STROTHS=1** ~~and **TENURE=4**~~ and **COVOTHS=2** and **STRAMT2** exists (insurance premium paid on structure of accommodation separately from any mortgage payments, ~~owner occupiers only~~) and **STRPD2=-1** or 1-11 **STRINTH=STRAMT2\*0.6666**

**If **STROTHS=1** and **COVOTHS=1** and **STRAMT2** exists (insurance premium paid on structure of accommodation separately from any mortgage payments) and **STRPD2=-1** or 1-11 **STRINTH=STRAMT2****

Add **STRINTH** to **HSCOSTHH**

From ~~owner~~ **HOUSEHOL** record get **CHARGE1-7**, **CHAMT1-7**, **CHARGE1-7**

If **CHARGE1=1** and **CHARGE1=-1**, or 1-11 add **CHAMT1** to **HSCOSTHH**

If **CHARGE2=1** and **CHARGE2=-1**, or 1-11 add **CHAMT2** to **HSCOSTHH**

If **CHARGE3=1** and **CHARGE3=-1**, or 1-11 add **CHAMT3** to **HSCOSTHH**

If **CHARGE4=1** and **CHARGE4=-1**, or 1-11 add **CHAMT4** to **HSCOSTHH**

If **CHARGE5=1** and **CHARGE5=-1**, or 1-11 add **CHAMT5** to **HSCOSTHH**

If **CHARGE6=1** and **CHARGE6=-1**, or 1-11 add **CHAMT6** to **HSCOSTHH**

If **CHARGE7=1** and **CHARGE7=-1**, or 1-11 add **CHAMT7** to **HSCOSTHH**

From **RENTER HOUSEHOL** record,

If **CWATAMTD** exists ~~and **RENTHOL=1** and **WEEKHOL** exists adjust **CWATAMTD** as above and add adjusted **CWATAMTD** to **HSCOSTHH**. Else if **CWATAMTD** exists~~ add the amount in **CWATAMTD** into **HSCOSTHH**.

0 Not applicable as household has no housing costs.

-2 Unable to derive - if any of above variables are missing or where a derived variable has already been set to -2.



**3 Results**

Tabulation is required to show the number of households by the total weekly amount of housing costs they pay divided into the following bands

£50 or under  
£50.01 - £100  
£100.01 - £150  
£150.01 - £200  
£200.01 - £250  
£250.01 - £300  
£300.01 - £350  
£350.01 or over

-----  
 --

**INCSE1**

Purpose : To calculate the total amount of income from self-employment using the traditional CSO/FES approach :

Created : VC - 17 August 1993

Database Table : ADULT

Minimum Value : 0

Maximum Value :

Units : Real

Validations :

Related Variables : ~~INCSE2~~ LOSS

Children :

Parents :

Core variable/user : ASD4A

Amendments : VC - 14 September 1993 To amend period used to calculate uprating factor : as may not be full 12 months.  
 : VC - 1 March 1994 To exclude period codes 12 and 13  
 : JS - 21 February 1996 to allow for skipped values where variables have been imputed and to take on changes made by ASD6C  
 : VE - 23 May 1996 - Initial amendments for V32 changes - replaced OWNFRQ with OWNPD  
 : VE - 2 July 1996 - To include references to standard uprating table  
 : VE - 19 August 1996 - To amend the divisor on page 2, para 4 from 12 to PERIOD  
 : VE - 25 November 1996 - to allow for cases where PARINC has been imputed  
 : SG - 16 October 1997 - correct coding of unable to derive

Issued : 21 May, 2003

**1 Definition**

This variable is coded as

INCSE1 This is the total amount of income received from self-employed earnings which has been calculated using the traditional CSO/FES approach to the definition of earnings. The CSO/FES definition includes drawings from a business where a person has declared that they have made a loss or nil profit from their business. The amount will be uprated to a common time period using uprating factors provided by Mr Nicholls **found in table 5.1 (Average earnings index) of Labour Market Trends. The index in LMT needs to be scaled to be consistent with previous FRS datasets. The constants are found in the table V32Constants of h:\frs\ldvars\doc\spreads.xls**

-1 Not applicable to this case

-2 Unable to derive due to missing values

-----  
 ---

-----  
 --

The variable is derived from several variables in the ADULT and JOB tables. Firstly, the person must be working or temporarily absent from his/her self-employment, so where WORKING = 1 (did paid work in last 7 days) or where JOBAWAY = 1 (not worked in last 7 days but has job to return to) and EMPEE = 2 (indicating that the job is self-employed).

The variables used to indicate whether a profit or loss has been made and the amount of profit or loss are - PROFIT1 which holds the total amount of profit or loss made by the business and PROFIT2 which indicates whether it is a profit (PROFIT2 = 1) or a loss (PROFIT2 = 2).

Where a profit has been made the full amount of profit may only be used if the person does not have a business partner or if he/she does have a business partner the amount in PROFIT1 has been declared after the partner's share has been deducted. Therefore, if a person has a business partner (SOLE = 2) and if the partner's share has not been deducted (PARINC = 1) the amount held in PARAMT must be deducted.

Once the total amount of profit has been established it has to be increased by an uprating factor as it is likely to be from a particular accounting period which may not be consistent with the interview date. The period that the earnings have been taken from are defined by the variables SE1 and SE2 which give the start and the end of the accounting period. These are used to the nearest full month (not day) and the total number of months that this covers is worked out from the start and end date of SE1 and SE2.

The end of the self-employment period is first constrained to end no more than six years before the FRS year. An uprating factor is established from the tables ~~provided by Mr Nicolls~~ by adding together each index for the months covered by SE1 and SE2 and then dividing by the period. A second average uprating factor is also calculated from the total indices for the equivalent period before the interview date (INTDATE) divided by ~~12~~ PERIOD. The amount of earnings in PROFIT1 is then multiplied by the result of dividing these two indices to provide the uprated profit. NB A loss or drawings from the business will not be affected by uprating factors.

If PROFIT2 = 2 indicating a loss, the amount used for INCSE1 will be the amount of drawings which is established from OWNSUM = 1 (has drawn sums from business for own use) and the amount withdrawn is to be found in OWNAMT. This should also be used if the amount declared in PROFIT1 is zero.

However, if the period code for ownamt is 12 or 13 (lumpsum/one-off or other period) the record must be set to unable to derive as it has not been possible to convert the amount of benefit into a weekly amount during the database conversion process. Therefore, if ~~OWNFRQ~~ OWNPD = 12 or 13 INCSE1 is set to - 2.

**If PROFIT1 is missing (not known or refused) then INCSE1 should be set to unable to derive.**

Where variables have been imputed, questions following will have been skipped. This means allowances have to be made in the spec as follows:

- i For OWNAMT, OWNPD=-1 is allowed (imputed amounts will be weekly)
- ii For PROFIT1, have to assume that the amount is weekly because SE1 and SE2 will have been skipped (this is acceptable), that it is a profit, because PROFIT2 will have been skipped (this is ropey: although PROFIT2 was taken in to account during imputation using

-----  
 ---

-----  
 --

hotdecking, there may be some losses figures included which cannot be identified) and that they are working on their own. These cases will not be updated.

- iii **For PARAMT, if PARINC has been imputed and PARAMT remains skipped, PROFIT1 should be used to calculate INCSE1**

## 2 FRS Specification

For each ADULT, convert SE1, SE2 and INTDATE into month year only date variables.

Code            Condition

INCSE1 Where WORKING = 1 or JOBAWAY = 1, set INCSE1 to 0

For each JOB record where EMPEE = 2 (self-employed)

If PROFIT2 = 1 and SOLE = 1 ((no partner) or SOLE=2 and PARINC=2) calculate uprating factor.

If PROFIT2= 1 and SOLE = 2 (has partner) and PARINC = 1 (partner's share included in profit1) **and PARAMT exists** subtract PARAMT from PROFIT1 then calculate uprating factor using new amount for PROFIT1. **Else if PARAMT=-1 calculate uprating factor from PROFIT1.**

Calculate the uprating factor and accumulate profits as follows -

Self-employment period is constrained to end no more than six years before the FRS year:

if SE2 < April 1988 then SE2 = March 1988, SE1 = April 1987

Calculate PERIOD referred to by SE1 and SE2 = (SE2 - SE1)/30.416666 and then round to the nearest month (using RND function). This will produce the number of months between SE1 and SE2 as accurately as possible.

Calculate SEINDEX1 as sum of each index over period from SE1 to SE2 divided by PERIOD (as above) using look up table of uprating values

Calculate SEINDEX2 as sum of each index over (PERIOD-1) months prior to date of interview (INTDATE) divided by PERIOD.

Calculate uprated profit UPROFIT = PROFIT1 \* (SEINDEX2/SEINDEX1)

Calculate INCSE1 = UPROFIT + INCSE1

If (PROFIT2 = 2 or PROFIT1 = 0) and OWNSUM = 1 and OWNPD = -1 or 1-11

Compute INCSE1 = OWNAMT + INCSE1

Else if PROFIT1 exists INCSE1 = PROFIT1

-----  
--

-1                    Not applicable - people who are not working or are employees

-2                    Underivable - where any of above values are missing or OWNPD = 12 or 13.

**Note**

**A check needs to be included in the program to look at unrealistic uprating factors. All factors over 1.05 should be listed for checking (this cut off may need to be amended).**

**3        Results**

Tabulation is required to show the number of adults falling into the following bands of weekly earnings from self-employment

- Under £50
- £50 - £100
- £100 - £150
- £150 - £200
- £200 - £250
- £250 - £300
- £300 - £350
- £350 or over

-----  
 --

**INDINC**

Purpose :To indicate the amount of gross income received by an adult for use in the FRS publication (based on GROSSINC).  
 Created : 31 January 1996  
 Database Table : ADULT  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user :ASD4A  
 Amendments : JS - 21 March to correct INRINC for luncheon vouchers (only asked about first job); INPENINC to allow for skipped tax (where penpay has been imputed or TU pension) and INEARNNS for questions which are only asked of the first job  
 :JS - 12 April 1996 - to add in income for contributions to rents/mortgages from those outside the household  
 VE - 17 April 1996 - to correct ININV to add in tax for those accounts where interest is received after tax  
 : VE - 23 May 1996 - Initial amendments for V32 - Still need to look at accounts and benefits to check definitions are the same as for V31  
 : VE - 16 July 1996 - To amend typo in adjustments for SSP and SMP and to include SSP and SMP in benefits block for cases where the benefit is recorded in the benefits table but not in the job section  
 : VE - 19 August 1996 - To update ACCOUNTS table definitions in ININV for V32  
 : VE - 20 August 1996 - to amend benefit definitions for V32  
 : JS - 12 November 1996 - to amend SSP/SMP adjustment  
 : VE - 20 November 1996 - to amend INRINC  
 : VE - 20 November 1996 - to replace IND TAX with ACCTAX  
 : VE - 28 November 1996 - to amend so that UBONAMT is only subtracted from INEARNNS when UBONINC=1  
 : VE - 11 December 1996 - to amend the OJAMT step to take into account the fact that for V32 OJPD=12 or 13 have been weeklyised  
 : VE - 23 January 1997 - to adjust HBENAMT and SCHBAMT for any rent free holidays as in HHRENT  
 : VE - 31 January 1997 - to make adjustments to bring into line with HBAI  
 Issued : 21 May, 2003

**1 Definition**

This variable is coded as

INDINC The total amount of gross income received by an adult from all sources.

0 Not applicable as adult does not have any gross income.

-----  
--

-2                   Unable to derive due to missing values.

**For V32, the specification for INDINC has been adjusted to bring it in line with HBAI definitions. A new DV, OLDINC, has been set up to maintain the old definition of INDINC to enable comparisons to be made between V31 and V32.**

The total amount of gross income is derived from numerous variables from the ADULT, JOB, BENEFITS, ODDJOB and PENSIONS records which when added together form the person's total gross income. It includes gross normal earnings (cf with GROSSPAY which is last pay), self-employed earnings, tax paid on pensions annuities, other income in the form of benefit income, income in kind, royalties, other allowances, income from trust funds and odd jobs etc. Unlike V30 GROSSINC the adjustment for SSP/SMP is only used to avoid double counting. **For V32, the SMP/SSP adjustment causes problems for one case (15325171) where SMP is received on the first job where the income is less than £52.50, but adding on income received from the second job gives a value greater than SMP. Since the adjustment is made to INEARNs, SMP is deducted for this case, but it should not be. The programmes need to be amended for V33 to ensure the adjustment is done on a job by job basis.** Income Support is included gross of any direct payments or social fund repayments. Income from boarders/lodgers has been excluded to avoid double counting at a household level and to simplify definitions (this will need to be discussed in the publication).

Additional amounts for direct expenses from absent partners, regular contributions from household members, education grants and other deductions from pension income have also been included.

Private benefit schemes are included unless they are one off/lump sum payments (these are ignored).

Income from free school milk and meals and free welfare milk are also included (allocated to the head of benefit unit).

Amounts of maintenance are also checked to make sure they are usual.

However, if the period code for the benefit is 12 or 13 (lumpsum/one-off or other period) the record must be set to unable to derive as it has not been possible to convert the amount of benefit into a weekly amount during the database conversion process. Therefore, for example, if PAYPD = 12 or 13 INDINC is set to -2. However, for odd jobs, since by definition they should be irregular, lump sum payments are allowed but are divided by 4 to produce a weekly amount. Additionally, coding has been changed to allow skipped values for period codes: this will occur where the (weekly) amount has been imputed. It has been decided to leave period codes as "skipped" in these cases since this may help flag imputation.

**The addition of HB is adjusted for any rent free holidays as in the derivation of HHRENT. This adjustment is done for conventional households and for shared households, but not for boarders and lodgers. The reasoning behind this is that boarders and lodgers would probably pay rent every week even if the household had some rent free weeks.**

Note: additional categories are now included for personal pensions (some grouped together on V30)

The DV spec is also used to set up components of gross income which are accumulated to obtain BU and HH level variables. These are:

-----  
 --

INEARNS      earned income  
 INSEINC      self employment income (identical to INCSE1 but set to zero if not applicable)  
 ININV        investment income  
 INRPINC      retirement pension plus any income support  
 INPENINC     other pensions  
 INDISBEN     disability benefits  
 INOTHBEN     other benefits  
 INRINC      remaining income

For pensioners, any retirement pension is taken together with IS to avoid any issues of misreporting. Analyses of pensioner income by ASD3 also take these two together.

Disability benefits comprise war disablement benefit, DWA, SDA, AA, DLA (mob and care); IIDB and IVB

## 2      FRS Specification

For each ADULT

Code            Condition

INDINC   **Gross earnings: INEARNNS**

From ADULT record, set INEARNNS to zero

If WORKING = 1 or JOBAWAY = 1 - process each JOB record for that person and

**If ABSWHY ne 3 (To bring in line with HBAI)**

If EMPEE = 1

(PAYUSL, and therefore UGROSS if PAYUSL equals "no" is only asked for JOBTYP=1, therefore coding has been changed so that a check is made to see if UGROSS exists. Where it doesn't, gross pay is calculated dependent on whether a payslip has been consulted, where it does - which can only be JOBTYP=1 - UGROSS is used)

If UGROSS does not exist

and If PAYSLIP = 1

    If GRWAGE exists and PAYPD equals -1 or 1 to 11, add it into INEARNNS

    If it is missing set INDINC and INEARNNS -2

else if PAYSLIP = 2 (or PAYSLIP=1 and GRSWAGE=-1) and PAYPD equals -1 or 1 to 11 ,

    If PAYAMT exists, add it into INEARNNS.

    If it is missing set INDINC and INEARNNS to -2



-----  
 --

If PAYE exists, add it into INEARNNS.  
 If it is missing do not change INEARNNS

If NATINS exists, add it into INEARNNS.  
 If it is missing do not change INEARNNS

(NATINS only asked for JOBTYP=1, do avoid variable falling over at this point, only include PAYE and NATINS if they exist, else do not change INEARNNS)

Then (AMTTAXF only asked if CHARITY=1 and CHRTAXF=1 and only for first job: to avoid setting cases to -2, only add in if value exists as passing through jobtypes 1-3)  
 if AMTTAXF exists add it into INEARNNS

if AMTOTH exists add it into INEARNNS  
 (as for AMTTAXF)

If OTHDED1 = 1 add DEDUC1 to INEARNNS  
 If OTHDED2 = 1 add DEDUC2 to INEARNNS  
 If OTHDED3 = 1 add DEDUC3 to INEARNNS  
 If OTHDED4 = 1 add DEDUC4 to INEARNNS  
 If OTHDED5 = 1 add DEDUC5 to INEARNNS  
**If OTHDED6 = 1 add DEDUC6 to INEARNNS**  
 If OTHDED67 = 1 add DEDOTH to INEARNNS

Else if UGROSS exists

If UGROSS exists add UGROSS to INEARNNS  
 If it is missing do not change INEARNNS (ie use PAYAMT calculation if it exists)

**Adjustments to gross earnings for HBAI consistency:**

income tax refunds, mileage and motoring allowances, refunds for items of household expenditure

if INEARNNS<>-2 (other conditions relating to PAYAMT and PAYPD will have been met by this point if INEARNNS has not been set to -2)

and UGROSS does not exist (ie all jobtypes except jobtype=1 where pay not usual)

and JOBTYP=1 and TAXAMT exists INEARNNS=INEARNNS-TAXAMT  
 (TAXAMT only asked for first job)

and MILEAMT exists INEARNNS=INEARNNS-MILEAMT

-----  
 --

and MOTAMT exists  $INEARNNS=INEARNNS-MOTAMT$

and HHA1 exists  $INEARNNS=INEARNNS-HHA1$

and HHA2 exists  $INEARNNS=INEARNNS-HHA2$

and HHA3 exists  $INEARNNS=INEARNNS-HHA3$

else if UGROSS exists

and U2MOT exists  $INEARNNS=INEARNNS-U2MOT$

**Adjustments to gross earnings for HBAI consistency:**

addition of bonuses received in last 12 months divided by 52

for up to 6 bonuses  $i=1-6$ :

If BONAMT(i) exists and BONTAX(i) (before tax)=1  
 $INEARNNS=INEARNNS+((BONAMT(i)/52))$

If BONAMT(i) exists and BONTAX(i)=2 or -1 (after tax or skipped where  
 BONAMT imputed)  
 $INEARNNS=INEARNNS+((BONAMT(i)/52)/0.75)$

(questions about bonuses are asked regardless of whether pay usual or not,  
 however, if UGROSS has been taken, have to make sure that bonus is not  
 double counted)

If UGROSS exists  
 and **UBONINC=1** and UBONAMT exists  
 $INEARNNS=INEARNNS-(UBONAMT/0.75)$

(UBONAMT is on a net basis, assume that if included in net pay ie  
 UBONINC=yes. then it will also have been in usual gross pay; UBONAMT is  
 only asked if UBONINC=yes. **Need the condition 'if UBONINC=2' to  
 account for cases where UBONINC has been edited**.)

**Adjustment for possible receipt of SSP or SMP**

Set ADJUST (temporary variable) and INOTHBEN to zero

For JOBTYP=1 only (SSP/SMP questions only asked once)

If JOBAWAY = 1 and ABSWHY = 2 and (SSPSMP = 1 or 2) and PAYSLIP = 1  
 Calculate ADJUST = SSPAMT

-----  
 --

If JOBAWAY = 1 and ABSWHY = 6 and (SSPSMP = 1 or 3) and PAYSLIP = 1  
 Calculate ADJUST = ADJUST + SMPAMT

If JOBAWAY=1 and ABSWHY=2 and (SSPSMP=1 or 2) and PAYSLIP=2 and  
 SSPRATE=1 or 2 Calculate ADJUST =52.50

~~If JOBAWAY=1 and ABSWHY=2 and (SSPSMP=1 or 2) and PAYSLIP=2 and  
 SSPRATE=2 Calculate ADJUST=47.80~~

If JOBAWAY=1 and ABSWHY=6 and (SSPSMP=1 or 3) and PAYSLIP=2 and  
~~SSPRATE=1 SMPRATE=1 or 2~~ Calculate ADJUST=ADJUST+52.50-

~~If JOBAWAY=1 and ABSWHY=6 and (SSPSMP=1 or 3) and PAYSLIP=2 and  
 SSPRATE=2 SMPRATE=2 Calculate ADJUST=ADJUST+48.80~~

~~If ADJUST >= to employment income calculated in INEARNNS, reset ADJUST to  
 INEARNNS~~

**If ADJUST>INEARNNS, then leave INEARNNS. This is in line with HBAI  
 assumptions that where SSP is bigger than the pay it's supposedly in then the  
 respondent has made a mistake and has actually quoted their pay after  
 SSP/SMP.**

**Else** INEARNNS=INEARNNS-ADJUST.  
 Add ADJUST to INOTHBEN

This adjustment is to check whether any SMP/SSP is included in gross earnings and  
 to switch it to benefit income. Previous specifications of GROSSINC reduced earnings  
 in respect of SSP/SMP where a payslip was consulted but did not include amounts  
 with benefits because the benefits table did not record them (amounts are only recorded  
 there if respondents have not answered questions as part of pay). They therefore  
 undercounted income by the rates of SSP/SMP in appropriate cases. Within the pay  
 block, if a payslip is consulted (payslip=1), respondents are asked for the amount  
 included (SSPAMT/SMPAMT) otherwise, they are asked what rate was in payment  
 (SSPRATE). These are standard rates (apart from higher SMP where an estimate has  
 been taken - this will not affect overall income). SSPSMP is coded 1=both SSP and  
 SMP, 2=SSP only, 3=SMP only. Earned income equals amount calculated at  
 INEARNNS less the estimated SMP/SSP (the adjustment). ~~Where the total adjustment  
 is greater than recorded income, all income is assumed to be from benefit and the  
 adjustment is reset to INEARNNS (ie INEARNNS=0).~~ ADJUST is added to INOTHBEN.

### Self - employment income: INSEINC

Set INSEINC=zero  
 If WORKING = 1 or JOBAWAY = 1 INSEINC=INCSE1

### Other sources of income: INRINC

-----  
--

Set INRINC to zero

#### **Income as a baby-sitter**

From ADULT record, if BABY1 = 1 and BABNOW=1 (doing work as a baby sitter currently) add BABPAY into INRINC

#### **Income as a mail order agent**

From ADULT record, if BABY2 = 1 and BABNOW=1 add BABPAY into INRINC

#### **Allowance from absent spouse**

From ADULT record, if ABSPAR = 1 and APPD equals -1 or 1 to 11, add APAMT to INRINC.

If APDIR=1 and APDPD equals -1 or 1 to 11 add APDAMT to INRINC

#### **Allowances from spouse in forces, friends other relatives etc**

From ADULT record, if ALLOW1 = 1 and ALLPD1 equals -1 or 1 to 11 , add ALLPAY1 to INRINC.

#### **Allowance from an organisation**

From ADULT record, if ALLOW2 = 1 and ALLPD2 equals -1 or 1 to 11 add ALLPAY2 to INRINC.

#### **Allowance from a Local Authority for a foster child**

From ADULT record, if ALLOW3 = 1 and ALLPD3 equals -1 or 1 to 11 add ALLPAY3 to INRINC.

#### **Allowance from a Local Authority for an adopted child.**

From ADULT record, if ALLOW4 = 1 and ALLPD4 equals -1 or 1 to 11, add ALLPAY4 to INRINC.

#### **Income in kind**

From JOB record, if JOBTYP=1 and LVAMT exists add amount in LVAMT into INRINC (luncheon vouchers) else do not change INRINC (*lvamt only asked if lunchv=1 and lv7dy=1, and only asked of first job*)

~~From ADULT record, if FCASH = 1 **COALCOKE=2** and FCAMTPD equals -1 or 1 to 11 add amount in FCAMT into INRINC (cash in lieu of concessionary coal)~~

#### **Royalties**

-----  
 --

From ADULT record, if ROYAL1 = 1 add ROYYR1 into INRINC

#### Income as a sleeping partner

From ADULT record, if ROYAL2 = 1 add ROYYR2 into INRINC.

#### Pension from an overseas Government

From ADULT record, if ROYAL3 = 1 add ROYYR3 into INRINC.

#### Maintenance

From ADULT record, if MNTREC = 1 ~~and MNTDSS=1~~ **MNTDSS1=1** and MNTPD1 equals -1 or 1 to 11 ~~and MNTUS1=1~~, add MNTAMT1 into INRINC.

~~Else if MNTREC=1 and MNTDSS=1 MNTDSS1=1 and MNTUS1=2 and MNTUSPD1 equals -1 or 1 to 1 add MNTUSAMT1 into INRINC~~

#### Odd jobs

From ADULT record if ODDJOB =1  
 for all occurrences of OJAMT in ODDJOB record  
 if OJPD =-1 or 1-11, add OJAMT into INRINC  
~~else if OJPD=12 or 13 add OJAMT/4 into INRINC~~

#### Income from property

If PROPARENT exists add PROPARENT to INRINC.

#### Income from sub-tenants

If SUBLET = 1, add SUBRENT into INRINC for PERSON = 1 (head of household).

#### Income from those outside the household paying towards rents/mortgages

For rented property, all contributions are included ~~(including any from DSS other than HB, this is consistent with HHRENT~~ **excluding contributions from DSS, consistent with HBAI)**. The income is assumed to be for the head of household  
 From RENTCONT record if ACCNONHH =1 **and ACCPAY <> 1** and ACCAMT exists **and RENTHOL=2** and ACCPD=-1 or 1-11 then INRINC=INRINC+ACCAMT for PERSON=1 only (head of household). **If ACCNONHH=1 and ACCPAY <> 1 and ACCAMT exists and ACCPD=-1 or 1-11 and WEEKHOL exists multiply ACCAMT by the number of weeks rent actually paid (52-WEEKHOL) and divide the result by 52. Add this to INRINC for PERSON=1 only**

For those buying their house with a mortgage, contributions from outside are included for all cases except the DSS. This is to avoid double counting of direct payments which are already included as part of benefit income. Amounts are included for all

-----  
 --

types of mortgage (having been explicitly added back for endowment mortgages where appropriate and implicit in repayment mortgages calculation)

From MORTCONT record if OUTSMORT=1 and OUTSPAY=2,3,4 or 5 (ie not equal to 1) OUTSAMT exists and OUTSPD=-1 or 1-11 then INRINC=INRINC+OUTSAMT for PERSON=1 only.

#### **Income from education grants**

If TOTGRANT exists add TOTGRANT/52 to INRINC

#### **Income from free welfare milk, free school meals and free school milk**

(these are assumed to be income of the head of benefit unit - largely for convenience, individual amounts are not held on the data base)

If from BENUNIT record FWMLKBU exists add FWMLKBU to INRINC *for head of benefit unit only*

If from BENUNIT record FSMBU exists add FSMBU to INRINC *for head of benefit unit only*

If from BENUNIT record FSMLKBU exists add FSMLKBU to INRINC *for head of benefit unit only*

#### **Income from student loans**

If from ADULT record TUBORR exists add TUBORR/52 to INRINC

#### **Interest/income from savings accounts or investments: ININV**

set ININV to zero

From ACCOUNTS record,

if ACCOUNT = 1 and ACCTAX=1, add (4/3 x amount in ACCINT) to ININV

**if ACCOUNT = 1 and ACCTAX<>1, add amount in ACCINT to ININV**

if ACCOUNT = 2 and ACCTAX=1, add (4/3 x amount in ACCINT) to ININV

**if ACCOUNT = 2 and ACCTAX<>1, add amount in ACCINT to ININV**

if ACCOUNT = 3 and ACCTAX=1, add (4/3 x amount in ACCINT) to ININV

**if ACCOUNT = 3 and ACCTAX<>1, add amount in ACCINT to ININV**

if ACCOUNT = 4, add amount in ACCINT to ININV

if ACCOUNT = 5 and ACCTAX=1, add (4/3 x amount in ACCINT) to ININV

**if ACCOUNT = 5 and ACCTAX<>1, add amount in ACCINT to ININV**

if ACCOUNT = 6 and ACCTAX=1, add (4/3 x amount in ACCINT) to ININV

**if ACCOUNT = 6 and ACCTAX<>1, add amount in ACCINT to ININV**

if ACCOUNT = 7 and ACCTAX=1, add (4/3 x amount in ACCINT) to ININV

**if ACCOUNT = 7 and ACCTAX<>1, add amount in ACCINT to ININV**

if ACCOUNT = 8 and ~~INVTAX~~ACCTAX=1, add (4/3 x amount in ACCINT) to ININV

**if ACCOUNT = 8 and ~~INVTAX~~ACCTAX<>1, add amount in ACCINT to ININV**

if ACCOUNT = 9, add (4/3 x amount in ACCINT) to ININV

-----  
 --

if ACCOUNT = 10, add ~~(4/3 x~~ amount in ACCINT) to ININV  
 if ACCOUNT = 11, add ~~(4/3 x~~ amount in ACCINT) to ININV  
~~if ACCOUNT = 12, add amount in ACCINT to ININV~~  
~~if ACCOUNT = 13, add amount in ACCINT to ININV~~  
~~if ACCOUNT = 14, add (4/3 x amount in ACCINT) to ININV~~  
~~if ACCOUNT = 15, add amount in ACCINT to ININV.~~

### Personal pensions: INPENINC

set INPENINC to zero

#### Occupational pensions

From PENSIONS record, if PENTYPE = 1 (occupational pension) and PENPD equals -1 or 1 to 11,

INPENINC=PENPAY. If PTINC exists and = 2 and PTAMT exists add PTAMT then add result into INPENINC (otherwise, do not change INPENINC) If PENOTH exists and equal to 1 and POINC exists and =2 (other deductions from pension not included in figure at PENPAY) and POAMT exists add POAMT to INPENINC otherwise, do not change INPENINC.

*(this is the same approach as taken in OCCUPPEN to deal with skipped values)*

#### Widow's employee pension

From PENSIONS record, if PENTYPE = 2 (widow's employee pension) and PENPD equals -1 or 1 to 11,

INPENINC=INPENINC+PENPAY. If PTINC exists and = 2 and PTAMT exists add PTAMT then add result into INPENINC (otherwise, do not change INPENINC) If PENOTH exists and equal to 1 and POINC exists and =2 (other deductions from pension not included in figure at PENPAY) and POAMT exists add POAMT to INPENINC otherwise, do not change INPENINC.

#### Personal pension

From PENSIONS record, if PENTYPE = 3 and PENPD equals -1 or 1 to 11,

INPENINC=INPENINC+PENPAY. If PTINC exists and = 2 and PTAMT exists add PTAMT then add result into INPENINC (otherwise, do not change INPENINC)

#### Trade union Friendly society pensions

From PENSIONS record, if PENTYPE = 4 and PENPD equals -1 or 1 to 11,

get amount from PENPAY and add to INPENINC If PTINC = 2 add PTAMT then add result into INPENINC

#### Annuity pension

From PENSIONS record, if PENTYPE = 5 and PENPD equals -1 or 1 to 11,

-----  
 --

INPENINC=INPENINC+PENPAY. If PTINC exists and = 2 and PTAMT exists add PTAMT then add result into INPENINC (otherwise, do not change INPENINC)

#### Trust/covenant

From PENSIONS record, if PENTYPE = 6 and PENPD equals -1 or 1 to 11, get amount from PENPAY and add to INPENINC. If PTINC exists and = 2 and PTAMT exists add PTAMT then add result into INPENINC (otherwise, do not change INPENINC)

#### Income from benefits: INRPINC, INDISBEN, INOTHBEN (plus bits of INRINC held in BENEFITS record)

set INRPINC, INDISBEN to zero

#### Housing Benefit

if BENUNIT = 1 and PERSON = 1: From RENTER record, **if RENTHOL = 1 and WEEKHOL exists and HBENEFIT = 1 and HBENPD equals -1 or 1 to 11, multiply HBENAMT by number of weeks rent actually paid (52 - WEEKHOL) and divide the result by 52. Add to INOTHBEN.** If RENTHOL ne 1 and HBENEFIT = 1 and HBENPD equals -1 or 1 to 11, add HBENAMT to INOTHBEN.

From HOUSEHOL record, if HHSTAT = 1 then from ADULT record if CVHB = 1 and CVPD equals -1 or 1 to 11, and exists(CHBAMT) = 1 then add CHBAMT to INOTHBEN.

From HOUSEHOL record **and RENTER record**, if HHSTAT=2 **and RENTHOL ne 1** then from ADULT record if SCVHB=1 and SCHBPD equals -1 or 1 to 11 then add SCHBAMT to INOTHBEN.

**From HOUSEHOL record and RENTER record, if HHSTAT = 2 and RENTHOL = 1 and WEEKHOL exists and SCVHB =1 and SCHBPD equals -1 or 1 to 11, multiply SCHBAMT by (52 - WEEKHOL) and divide the result by 52. Add to INOTHBEN.**

#### Council Tax Benefit

If CTREB = 1, and CTREBPD equals -1 or 1 to 11 add CTREBAMT into **INOTHBEN** only for PERSON = 1.

#### Income from benefits

From BENEFITS record, if BENPD equals -1 or 1 to 11, and

If BENEFIT = 1 add BENAMT to INDISBEN (DLA Care)

If BENEFIT = 2 add BENAMT to INDISBEN (DLA Mob)



-----  
 --

If BENEFIT = 3 add BENAMT to INOTHBEN (CHB)

If BENEFIT = 4 add BENAMT to INOTHBEN(OPB)

If BENEFIT = 5 add BENAMT to INRPINC (RP)

If BENEFIT = 7 add BENAMT to INOTHBEN(Widows Pension)

If BENEFIT = 8 add BENAMT to INDISBEN (War Disablement Pension)

If BENEFIT = 9 add BENAMT to INDISBEN (SDA)

If BENEFIT = 10 add BENAMT to INDISBEN (DWA)

If BENEFIT = 11 add BENAMT to INDISBEN (AA)

If BENEFIT = 12 add BENAMT to INOTHBEN(Invalid Care Allowance)

If BENEFIT = 13 add BENAMT to INOTHBEN(UB)

If BENEFIT = 14 add BENAMT to INDISBEN (Industrial Injuries)

**If BENEFIT = 15 add BENAMT to INOTHBEN (SSP)**

~~If BENEFIT = 16 add BENAMT to INOTHBEN(Sickness Benefit)~~

If BENEFIT = 17 add BENAMT to INDISBEN (~~IVB~~ **Incapacity Benefit**)

If BENEFIT = 18 add BENAMT to INOTHBEN(FC)

If BENEFIT = 19 and (SEX=1 and AGE>=65) or (SEX=2 and AGE>=60) add BENAMT to INRPINC (IS)

else add BENAMT to INOTHBEN

If BENEFIT = 20 and VAR2=2

and ((SEX=1 and AGE>=65) or (SEX=2 and AGE>=60)) add BENAMT to INRPINC

else if BENEFIT=20 and VAR2=2 add BENAMT to INOTHBEN

*(adjustment for IS quoted after any direct payments: pensioners income must be included in INRPINC whilst others are in INOTHBEN)*

If BENEFIT = 21 add BENAMT to INOTHBEN(Maternity Benefit)

**If BENEFIT = 22 add BENAMT/26 to INOTHBEN (Soc Fund Grant for Maternity Expenses)**

**If BENEFIT = 23 add BENAMT to INOTHBEN (SMP)**

-----  
 --

**If BENEFIT = 24 add BENAMT/26 to INOTHBEN (Soc Fund Grant for Funereal Expenses)**

**If BENEFIT = 25 add BENAMT/26 to INOTHBEN (Comm Care Grant)**

If BENEFIT = 26 and PRES = 1 add BENAMT to INOTHBEN (Any other DSS benefits)

If BENEFIT = 27 and PRES = 1 and BENPD=-1 or 1-11 add BENAMT to INRINC (Trade Union sick) else if PRES=1 and BENPD=12 or 13 do not change INRINC

If BENEFIT = 28 and PRES = 1 and BENPD=-1 or 1-11 add BENAMT to INRINC (Friendly sick) else if PRES=1 and BENPD=12 or 13 do not change INRINC

If BENEFIT = 29 and PRES = 1 add BENAMT to INRINC (Private sick) else if PRES=1 and BENPD=12 or 13 do not change INRINC

If BENEFIT = 30 and PRES = 1 add BENAMT to INRINC (Accident) else if PRES=1 and BENPD=12 or 13 do not change INRINC

If BENEFIT = 31 and PRES = 1 add BENAMT to INRINC (Hospital savings) else if PRES=1 and BENPD=12 or 13 do not change INRINC

If BENEFIT = 32 add BENAMT to (Training) INRINC

If BENEFIT = 33, add BENAMT to INOTHBEN (Guardians Allowance)

If BENEFIT=34 and VAR2=2

and ((SEX=1 and AGE>=65) or (SEX=2 and AGE>=60)) add BENAMT to INRPINC

else if BENEFIT=34 and VAR2=2 add BENAMT to INOTHBEN

*(adjustment for IS quoted after social fund repayments: pensioners income must be included in INRPINC whilst others are in INOTHBEN)*

**If BENEFIT=35, add BENAMT/26 to INOTHBEN (FC lump sums)**

INDINC will then be calculated as follows for each ADULT -

INEARNS+INSEINC+ININV+INRPINC+INPENINC+INDISBEN+INOTHBEN+INRINC

-2

If any of above variables are missing or if a period code is 12 or 13 (skipped is OK) also applied to individual component variables ININV, INRPINC, INPENINC, INDISBEN, INOTHBEN and INRINC which are calculated in this spec (missing INEARN components are already documented above).

-----  
 --

### INIRBEN, INNIRBEN, BUIRBEN, BUNIRBEN, HHIRBEN, HHNIRBEN

Purpose : To show the total amount of income received from income (means tested) and non-income related (non-means tested) benefits at an individual, benefit unit and household level for use in the FRS publication.

Created : 5 February 1996

Database Table : ADULT, BENUNIT, HOUSEHOL

Minimum Value : 0

Maximum Value :

Units : Real

Validations :

Related Variables :

    Children :

    Parents :

Core variable/user : Publication

Amendments : VE - 5 June 1996 - To note that on April 13 1995 Incapacity Benefit replaced Invalidity Benefit and NI Sickness Benefit. Specification not changed because schema has not yet been amended to reflect this change.

                  : 20 August 1996 - updated for V32

                  : 21 November 1997 - SCG - V33 updates - benefit codes, SSP/SMP variables, HB rent holidays

Issued : 21 May, 2003

#### 1 Definition

This variable is coded as

INIRBEN The total amount of income received each week by individuals from income related benefits.

0 No income is received from income related benefits

-2 Unable to derive due to missing values

INNIRBEN The total amount of income received each week by individuals from non-income related benefits

0 No income is received from non-income related benefits

-2 Unable to derive due to missing values

Income related benefits are: IS, FC, DWA, HB, CTB, **Social Fund loans, Community Care Grants**

Non-income related benefits are: SSP, SMP, DLA, CHB, OPB, RP, Widows' pension, War disablement pension, SDA, AA, ICA, UB, Industrial Injuries, ~~Sickness benefit, IVB~~, **Incapacity Benefit**, Maternity benefit, **Guardian's Allowance, Back to work bonus, War Widows Pension**

-----  
 ---

-----  
 --

Other DSS benefits are included under non-income related benefits

Note: It may be possible to code these variables as part of INDINC as most of the coding is the same.  
 If not, these DVs should be run after INDINC because they use INEARNs for SSP/SMP

INIRBEN+INNIRBEN=INRPINC+INDISBEN+INOTHBEN  
 INRPINC includes both mean and non-income related benefits

The variables are used in tabulations as analysis and categorical variables, eg proportion of income from income related benefits and in receipt of income related benefits. In receipt is identified as where the amount is >0.

**2 FRS Specification**

For each adult

Code                      Condition

Set INIRBEN and INNIRBEN to zero and get INEARNs

**SSP or SMP**

set ADJUST (temporary variable) to zero

If JOBAWAY = 1 and ABSWHY = 2 and ~~(SSPSMP = 1 or 2)~~ **INCLPAY1=1** and  
 PAYSLIP = 1  
 Calculate ADJUST = SSPAMT

If JOBAWAY = 1 and ABSWHY = 6 and ~~(SSPSMP = 1 or 3)~~ **INCLPAY2=1** and  
 PAYSLIP = 1  
 Calculate ADJUST = ADJUST + SMPAMT

If JOBAWAY=1 and ABSWHY=2 and ~~(SSPSMP = 1 or 2)~~ **INCLPAY1=1** and  
 PAYSLIP=2 and ~~SSPRATE=1 or 2~~ Calculate ADJUST =~~52.50~~ **54.55**

~~If JOBAWAY=1 and ABSWHY=2 and (SSPSMP=1 or 2) and PAYSLIP=2 and  
 SSPRATE=2 Calculate ADJUST=47.80~~

If JOBAWAY=1 and ABSWHY=6 and ~~(SSPSMP = 1 or 3)~~ **INCLPAY2=1** and  
 PAYSLIP=2 and ~~SSPRATE=4~~ **SMPRATE=1 or 2** Calculate ADJUST=ADJUST+~~52.50~~  
**54.55**

~~If JOBAWAY=1 and ABSWHY=6 and (SSPSMP=1 or 3) and PAYSLIP=2 and  
 SSPRATE=2 Calculate ADJUST=ADJUST+48.80~~

-----  
 ---

-----  
 --

~~If ADJUST >= to employment income calculated in INEARNNS, reset ADJUST to zero.  
 INEARNNS~~

If ADJUST>INEARNNS, then leave INEARNNS. This is in line with HBAI assumptions that where SSP is bigger than the pay it's supposedly in then the respondent has made a mistake and has actually quoted their pay after SSP/SMP. Do same for NINEARNNS.

If ADJUST<INEARNNS then INEARNNS=INEARNNS-ADJUST

If ADJUST<NINEARNNS then NINEARNNS=NINEARNNS-ADJUST

Add ADJUST to **INNIRBEN**

### Housing Benefit

~~From RENTER record, if BENUNIT = 1 and HBENPD equals -1 or 1 to 11, and HBENEFIT = 1 add in HBENAMT to **INIRBEN** for PERSON = 1.~~

~~From HOUSEHOL record, if HHSTAT = 1 then from ADULT record if CVHB = 1 and CVPD equals -1 or 1 to 11, and exists(CHBAMT) = 1 then add CHBAMT to **INIRBEN**.~~

~~From HOUSEHOL record, if HHSTAT=2 then from ADULT record if SCVHB=1 and SCHBPD equals -1 or 1 to 11 then add SCHBAMT to **INIRBEN**.~~

if BENUNIT = 1 and PERSON = 1: From RENTER record, if RENTHOL = 1 and WEEKHOL exists and HBENEFIT = 1 and HBENPD equals -1 or 1 to 11 or 13 or 26 or 52, multiply HBENAMT by number of weeks rent actually paid (52 - WEEKHOL) and divide the result by 52. Add to INOTHBEN. If RENTHOL ne 1 and HBENEFIT = 1 and HBENPD equals -1 or 1 to 11 or 13 or 26 or 52, add HBENAMT to **INIRBEN**.

From HOUSEHOL record, (if HHSTAT = 2 or CVPAY>0 (boarders/lodgers paying rent) ) then from ADULT record if HBOTHBU=1 (individual qualifies for HB) and CVPD equals -1 or 1 to 11 or 13 or 26 or 52, then add HBOTHAMT to **INIRBEN**.

### Council Tax Benefit

If CTREB = 1, and CTREBPD equals -1 or 1 to 11 or 13 or 26 or 52 add CTREBAMT to **INIRBEN** for PERSON = 1 only.

### Other benefits

From BENEFITS record, if BENPD equals -1 or 1 to 11 or 13 or 26 or 52 and

If BENEFIT = 1 add BENAMT to **INNIRBEN** (DLA Care)

-----  
 ---

-----  
--

If BENEFIT = 2 add BENAMT to **INNIRBEN** (DLA Mob)

If BENEFIT = 3 add BENAMT to **INNIRBEN** (CHB)

If BENEFIT = 4 add BENAMT to **INNIRBEN** (OPB)

If BENEFIT = 5 add BENAMT to **INNIRBEN** (RP)

If BENEFIT = ~~7~~ 6 add BENAMT to **INNIRBEN** (Widows Pension)

**If BENEFIT = 7 add BENAMT to INNIRBEN (Widowed mothers allowance)**

If BENEFIT = 8 add BENAMT to **INNIRBEN** (War Disablement Pension)

**If BENEFIT = 9 add BENAMT to INNIRBEN (War Widows Pension)**

If BENEFIT = ~~9~~ 10 add BENAMT to **INNIRBEN** (SDA)

If BENEFIT = ~~10~~ 11 add BENAMT to **INIRBEN** (DWA)

If BENEFIT = ~~11~~ 12 add BENAMT to **INNIRBEN** (AA)

If BENEFIT = ~~12~~ 13 add BENAMT to **INNIRBEN** (Invalid Care Allowance)

If BENEFIT = ~~13~~ 14 add BENAMT to **INNIRBEN** (UB)

If BENEFIT = ~~14~~ 15 add BENAMT to **INNIRBEN** (Industrial Injuries)

**If BENEFIT = ~~15~~ 16 add BENAMT to INNIRBEN (SSP)**

~~If BENEFIT = 16 add BENAMT to INNIRBEN (Sickness Benefit)~~

If BENEFIT = 17 add BENAMT to **INNIRBEN** (~~IVB~~ **Incapacity Benefit**)

If BENEFIT = 18 add BENAMT to **INIRBEN** (FC)

If BENEFIT = 19 add BENAMT to **INIRBEN** (IS)

If BENEFIT = 20 and VAR2=2 add BENAMT to **INIRBEN** (Direct payments not included in quoted IS )

If BENEFIT = 21 add BENAMT to **INNIRBEN** (Maternity Benefit)

**If BENEFIT=22 add BENAMT/26 to INIRBEN (Soc Fund Grant for Maternity Expenses)**

**If BENEFIT = 23 add BENAMT to INNIRBEN (SMP)**

-----  
---

-----  
 --

If BENEFIT=24 add BENAMT/26 to INIRBEN (Soc Fund Grant for Funeral Expenses)

If BENEFIT=25 add BENAMT/26 to INIRBEN (Comm Care Grant)

If BENEFIT=26 add BENAMT/26 to INNIRBEN (Back to work bonus)

If BENEFIT = ~~26~~ 30 and PRES = 1 add BENAMT to INNIRBEN (Any other DSS benefits)

If BENEFIT = ~~33~~ 37 add BENAMT to INNIRBEN (Guardian's Allowance)

~~If BENEFIT=34 and VAR2=2 add BENAMT to INIRBEN (Social Fund loan repayments included in quoted IS)~~

If BENEFIT=~~35~~ 41 add BENAMT/26 to INIRBEN (FC Lump Sum)

-2 variables set to unable to derive if any components are missing or period codes equal 12 or 13

**The benefit unit variables are calculated as:**

For each benefit unit

from ADULT table get INIRBEN and INNIRBEN

BUIRBEN equals total occurrences of INIRBEN

BUNIRBEN equals total occurrences of INNIRBEN

-2 if any components are missing

**The household variables are calculated as:**

For each household

from BENUNIT table get BUIRBEN and BUNIRBEN

HHIRBEN equals total occurrences of BUIRBEN

HHNIRBEN equals total occurrences of BUNIRBEN

-2 if any components are missing

-----  
 ---

-----  
--

**ISMNDD01 - ISMNDD10 (IN201-IN210)**

Purpose : To provide FES equivalent non-dependant deduction variables for ISM model  
Created : May 1995 as part of parallel run exercise  
Database Table : BENUNIT  
Minimum Value : 0  
Maximum Value :  
Units :  
Validations :  
Related Variables :  
    Children :  
    Parents :  
Core variable/user : ISM  
Amendments : VE - 16 April 1996 - To give fuller definitions  
              : VE - 4 June 1996 - No initial V32 updates needed  
              : VE - 26 September 1996 - To replace GROSSINC with INDINC  
Issued : 21 May 2003

**1 Definition**

These variables hold codes for how non-householder benefit units would be classified if it were a non-dependant. Up to 10 non-HoH benefit units can be assessed (ISMND01-ISMNDD10) in any given household.

Variables are consistent with FES variables IN201 to IN210

This variable is coded as

- 1 Indicating a member of the householder BU is in receipt of AA or DLA (care component).
- 2 Indicating an adult aged 18+ in full-time work in a non-HoH BU with a combined income of 130 or over.
- 3 Indicating an adult aged 18+ in full-time work in a non-HoH BU with a combined income of 100 or over but less than 130.
- 4 Indicating an adult aged 18+ in full-time work in a non-HoH BU with a combined income of 65 or over but less than 100.
- 5 Indicating an adult aged 18+ in full-time work in a non-HoH BU with a combined income of less than 65.
- 6 Indicating an adult aged 18+ in a non-HoH BU who is not a student, nor on a YTS scheme, nor under the age of 25 and in receipt of IS.
- 7 Indicating an adult aged 18+ in a non-HoH BU who is a full-time student or on a YTS scheme.

-----  
---



-----  
 --

8        Indicating an adult aged 16 or 17 in a non-HoH BU.

From the ADULT and JOB tables, a person is classified as being in full-time work if EMPSTATB=1 and QHRSELF>=16 (self-employed), or if EMPSTATB=2 and TOTHOURS>=16 (full-time employee), or if ABSPAY=1 (absent from work but being paid full pay).

Combined income is taken to be the sum of ~~GROSSINC~~ **INDINC** (from the ADULT table) for all members of the benefit unit .

**An individual is a student if**

**FTED = 1 and TYPEED = 4,5 or 7 or**

**If TEA = 96 and TYPEED = 4 or 5 or 7 or**

**If WORKING = 2 and JOBAWAY = 2 and NOWANT = 2 or NOLOOK = 2**

**2        FRS Specification**

Using the definitions of full-time work (FTW=1 implies an individual in full-time work), **students (STU=1 implies an individual satisfies the conditions above)** and combined income (CI) as above, for each non-HoH BU

Code    Condition

- 1        From table BENEFIT for HoH BU  
           If BENEFIT=1 or BENEFIT=11
- 2        From ADULT and JOB tables for non-HoH BU  
           If AGE>=18 and FTW=1 and CI>=130
- 3        From ADULT and JOB tables for non-HoH BUs  
           If AGE>=18 and FTW=1 and CI>=100 and <130
- 4        From ADULT and JOB tables for non-HoH BUs  
           If AGE>=18 and FTWK=1 and CI>=65 and <100
- 5        From ADULT and JOB tables for non-HoH BUs  
           If AGE>=18 and FTW=1 and CI<65
- 6        From ADULT and BENEFIT tables for non-HoH BUs  
           If AGE>=18 and ~~FTED=2~~ **STU=2** and (TRAIN not equal 1) and not (AGE<25 and BENEFIT=19)
- 7        From ADULT and BENEFIT tables for non-HoH BUs

-----  
 ---

-----  
--

If AGE>=18 and AGE<=24 and BENEFIT=19

8 From ADULT table for non-HoH BUs

If AGE>=18 and FTED=1 or TRAIN=1

9 From ADULT table for non-HoH BUs

If AGE=16 or AGE=17

-----  
---

-----  
 --

**KID04**

Purpose : To indicate the total number of children in the benefit unit aged 0 to 4 inclusive. :

Created : VC - 3 March 1993

Database Table : BENUNIT

Minimum Value : 0

Maximum Value:

Units : integer

Validations :

Related Variables :

    Children :

    Parents :

Core variable/user : ~~ASD6A~~ **ASD4A**

Amendments : VC - 11 May 1993 To expand and clarify the definition

              : VC - 17 February 1994 Amended to reflect version 30 changes

              : VE - 22 May 1996 - No initial Version 32 update needed

**1 Definition**

This variable is coded as

KID04 This is the number of children in each benefit unit aged 0 to 4 years inclusive.

-1 Not applicable to this case

-2 Unable to derive due to missing values.

**2 FRS Specification**

For each BENUNIT record

<u>Code</u>	<u>Condition</u>
Kid04	For each child in benefit unit, from CHILD table If age >= 0 and <= 4, count total number of children where age falls in this range.
-1	Not applicable to this case - this should not occur for this variable.
-2	Unable to derive as component information is missing.

**3 Results**

Tabulation to show the total number of children aged 0 to 4 by benefit unit.

-----  
 ---

-----  
--

**4 Test Cases**

None as yet

-----  
---

-----  
 --

**KID510**

Purpose : To indicate the total number of children in the benefit unit aged 5 to 10 inclusive.  
 Created : VC - 3 March 1993  
 Database Table : BENUNIT  
 Minimum Value : 0  
 Maximum Value:  
 Units : integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : ~~ASD6A~~ **ASD4A**  
 Issue date : 21 May 2003  
 Amendments : VC - 11 May 1993 amended to show coding  
               : VC - 17 February 1994 Amended to reflect version 30 changes  
               : VE - 22 May 1996 - No initial Version 32 update needed

**1 Definition**

This variable is coded as

KID510 This is the number of children in each benefit unit aged 5 to 10 inclusive.

- 1 Not applicable to this case.
- 2 Unable to derive due to missing values.

**2 FRS Specification**

For each BENUNIT record

<u>Code</u>	<u>Condition</u>
-------------	------------------

KID510 For each child in benefit unit, from CHILD table.  
 If age >= 5 and <= 10, count total number of children where age falls in this range.

- 1 Not applicable to this case - should not occur for this variable.
- 2 Unable to derive as missing data.

**3 Results**

Tabulation to show the total number of children aged 5 to 10 by benefit unit.

**4 Test Cases**

-----  
 ---

-----  
--

None as yet

-----  
---

-----  
 --

**KID1115**

Purpose : To indicate the total number of children in the benefit unit aged 11 to 15 inclusive.  
 Created : VC - 3 March 1993  
 Database Table : BENUNIT  
 Minimum Value : 0  
 Maximum Value :  
 Units : integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : ~~ASD6A~~ **ASD4A**  
 Amendments : VC - 11 May 1993 Amended to show coding  
               : VC - 17 February 1994 Amended to reflect changes to version 30  
               : VE - 22 May 1996 - No initial Version 32 update needed

**1 Definition**

This variable is coded as

- KID1115 This is the number of children in each benefit unit aged 11 to 15 inclusive.
- 1 Not applicable to this case.
- 2 Unable to derive due to missing values.

**2 FRS Specification**

For each BENUNIT record

<u>Code</u>	<u>Condition</u>
KID1115	For each child in benefit unit, from CHILD table If age >= 11 and <= 15, count number of children where age falls in this range.
-1	Not applicable to this case - should never occur for this variable.
-2	Unable to derive due to missing information.

**3 Results**

Tabulation to show the total number of children aged 11 to 15 inclusive by benefit unit.

**4 Test Cases**

-----  
 ---

-----  
--

None as yet

-----  
---



-----  
 --

**KID1618**

Purpose : To indicate the total number of dependents in the benefit unit aged 16 to 18 inclusive  
 Created : VC - 4 March 1993  
 Database Table : BENUNIT  
 Minimum Value : 0  
 Maximum Value :  
 Units : integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : ~~ASD6A~~ **ASD4A**  
 Amendments : VC - 11 May 1993 amended to show coding  
               : VC - 17 February 1994 Amended to reflect version 30 changed  
               : VE - 22 May 1996 - No initial Version 32 update needed

**1 Definition**

This variable is coded as

- KID1618 This is the number of dependents in each benefit unit aged 16 to 18 inclusive.
- 1 Not applicable to this case
- 2 Unable to derive due to missing values.

**2 FRS Specification**

For each BENUNIT record

<u>Code</u>	<u>Condition</u>
KID1618	For each child in benefit unit, from CHILD table If age >= 16 and <= 18, count total number of children where age falls in this range.  NB - The CHILD table includes all children aged 15 and under and those aged 16 to 18 inclusive who are in non-advanced full-time education.
-1	Not applicable to this case - should not occur for this variable.
-2	Unable to derive due to missing values.

**3 Results**

Tabulation to show the total number of dependents aged 16 to 18 inclusive by benefit unit.

-----  
 ---

-----  
--

**4 Test Cases**

None as yet

-----  
---

-----  
 --

**KIDS0BU, KIDS1BU....KIDS18BU**

Purpose : Total number of dependents aged under 1, aged from 1 to 2 years, ..  
 Created : NM - 3 November 1992  
 Database Table : BENUNIT  
 Minimum Value : 0  
 Maximum Value:  
 Units : Integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : ~~ASD6A~~ ASD4A  
 Amendments : VC - 9 February 1993 updated to version 29  
               : VC - 11 May 1993 amended to show coding  
               : VC - 17 February 1994 Amended to reflect version 30 changes  
               : VE - 22 May 1996 - No initial Version 32 update needed  
 Issue date : 21 May 2003

**1 Definition**

This variable is coded as

KIDS0BU	Number of dependents under age 1
KIDS1BU	Number of dependents between age 1 and 2
KIDS2BU	Number of dependents between age 2 and 3
KIDS3BU	Number of dependents between age 3 and 4
KIDS4BU	Number of dependents between age 4 and 5
KIDS5BU	Number of dependents between age 5 and 6
KIDS6BU	Number of dependents between age 6 and 7
KIDS7BU	Number of dependents between age 7 and 8
KIDS8BU	Number of dependents between age 8 and 9
KIDS9BU	Number of dependents between age 9 and 10
KIDS10BU	Number of dependents between age 10 and 11
KIDS11BU	Number of dependents between age 11 and 12
KIDS12BU	Number of dependents between age 12 and 13
KIDS13BU	Number of dependents between age 13 and 14
KIDS14BU	Number of dependents between age 14 and 15
KIDS15BU	Number of dependents between age 15 and 16
KIDS16BU	Number of dependents between age 16 and 17
KIDS17BU	Number of dependents between age 17 and 18
KIDS18BU	Number of dependents between age 18 and 19

The above are derived from the variable age in the CHILD table

-----  
 ---

-----  
 --

**2 FRS Specification**

For each BENUNIT record

Code            Condition

Using variable AGE from CHILD table which holds details of all children aged 15 and under and all 16 to 18 year olds in full-time, non-advanced education.

KIDS0BU	The sum of all dependents in the benefit unit where age < 1
KIDS1BU	The sum of all dependents in the benefit unit where age >= 1 and < 2
KIDS2BU	The sum of all dependents in the benefit unit where age >= 2 and < 3
KIDS3BU	The sum of all dependents in the benefit unit where age >= 3 and < 4
KIDS4BU	The sum of all dependents in the benefit unit where age >= 4 and < 5
KIDS5BU	The sum of all dependents in the benefit unit where age >= 5 and < 6
KIDS6BU	The sum of all dependents in the benefit unit where age >= 6 and < 7
KIDS7BU	The sum of all dependents in the benefit unit where age >= 7 and < 8
KIDS8BU	The sum of all dependents in the benefit unit where age >= 8 and < 9
KIDS9BU	The sum of all dependents in the benefit unit where age >= 9 and < 10
KIDS10BU	The sum of all dependents in the benefit unit where age >= 10 and < 11
KIDS11BU	The sum of all dependents in the benefit unit where age >= 11 and < 12
KIDS12BU	The sum of all dependents in the benefit unit where age >= 12 and < 13
KIDS13BU	The sum of all dependents in the benefit unit where age >= 13 and < 14
KIDS14BU	The sum of all dependents in the benefit unit where age >= 14 and < 15
KIDS15BU	The sum of all dependents in the benefit unit where age >= 15 and < 16
KIDS16BU	The sum of all dependents in the benefit unit where age >= 16 and < 17
KIDS17BU	The sum of all dependents in the benefit unit where age >= 17 and < 18
KIDS18BU	The sum of all dependents in the benefit unit where age >= 18 and < 19

**3 Results**

Tabulation is required to show the number of dependents in each benefit unit by age of the dependent.

**4 Test Cases**

None as yet

-----  
 ---

-----  
 --

**LODGER**

Purpose : To indicate the total weekly amount of rent paid by a lodger.  
 Created : VC - 12 March 1993  
 Database Table : BENUNIT  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : HBM  
 Amendments : VC - 24 April 1993 Change from adult to BU variable  
               : VC - 27 April 1993. To expand definition to include more details about the  
               : questions/database variables used.  
               : VC - 17 February 1994 Amended to reflect changes to version 30  
               : VC - 1 March 1994 To exclude period codes 12 or 13  
               : JS - 21 February 1996 to allow skipped CVPD where CVPAY has been  
               : imputed  
               : VE - 23 May 1996 - No initial V32 update needed  
 Issue date : 21 May 2003

NB - This is a new variable produced by FRS and does not replace FES in any way.

**1 Definition**

This variable is coded as

- LODGER The total weekly amount paid by a benefit unit classed as a lodger to the householder for a room but no food.
- 1 Not applicable to this case.
- 2 Unable to derive due to missing values.

LODGER is derived by benefit unit from the variable CONVBL which asks whether the person is a boarder or lodger or neither of these. Where CONVBL = 2 indicating that the adult is a lodger, the amount paid for lodging is to be found in cvpay.

However, if the period code for the the amount of rent paid by the lodger is 12 or 13 (lumpsum/one-off or other period) the record must be set to unable to derive as it has not been possible to convert the amount of benefit into a weekly amount during the database conversion process. Therefore, if CVPD = 12 or 13 LODGER is set to -2.

**2 FRS Specification**

-----  
 ---

-----  
 --

For each BENUNIT record and for each adult in the benefit unit

<u>Code</u>	<u>Condition</u>
LODGER	From ADULT table
	If CONVBL = 2 and CVPD equals -1 or 1-11, the amount of LODGER is the amount in CVPAY.
	If there is more than one adult in the benefit unit, the amount of LODGER is the total amount paid by both adults.
-1	Not applicable to this case - where CONVBL = 1 or 3 or is missing as the question has not been asked.
-2	Unable to derive due to value of CVPAY missing or CVPD = 12 or 13.

**3 Results**

Tabulation is required to show the number of lodgers paying rent in weekly bands of

- Under £25
- £25 - £50
- £50 - £75
- £75 - £100
- £100 - £125
- £125 - £150
- £150 or over

**4 Test Cases**

None as yet

-----  
 --

**MORTCOST**

Purpose : To show weekly housing expenditure for owner occupiers for use in the FRS publication.  
 Created : 2 February 1996  
 Database Table : HOUSEHOL  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables : HSCOSTHH, MORTINT  
     Children :  
     Parents : MORTPAY, ENDOWPAY, STRUINS, SERVPAY (also derived here)  
 Core variable/user : Publication/Regional Trends ASD6A **ASD4A** (**Bold** indicates lead user)  
 Amendments : To correct identification of mortgage protection policies  
               : VE - 4 June 1996 - No amendments needed for SERVPAY in initial V32 update  
               : VE - 5 June 1996 - No amendments needed for other DVs in V32 update  
               : VE - 20 November 1996 - To amend STRAMT to STRAMT1 and STRAMT2 where appropriate  
               : VE - 17 December 1996 - to amend INCPAMT and INCMPPD to INCPAM1, 2 and 3 and INCMPPD1, 2 and 3  
 Issue date : 21 May 2003

**1 Definition**

This variable is coded as

MORTCOST Total weekly housing (mortgage) costs of owner occupiers, including mortgage payments, endowment policies, structural insurance and service payments

0 Owner occupiers who have no housing costs

-1 Not applicable to this case (renter households)

-2 Unable to derive due to missing values

MORTCOST is derived from variables which occur in HSCOSTHH. The total is broken down into components which are to be used in a table for Regional Trends. The differences between HSCOSTHH for owner occupiers and MORTCOST are:

mortgage protection policies are included as part of mortgage payments  
 Endowment policies are included as a separate category  
 Water and Sewerage charges are excluded

-----  
 ---

-----  
 --

Mortgage payments are equal to MORTINT plus mortgage protection policies. MORTINT excludes any amounts for policies included in last payment so it can simply be added back in. Similarly, there is no problem of double counting endowment policies because these are also excluded from MORTINT.

Payments of structural insurance only should be included. However, where the premium includes furniture and contents in combination with the structure, the total is taken. Where payments are made separately from mortgage payments (STROTHS=1), questions are directed to those in rented accommodation as well as owner occupiers. Only cases where TENURE=1 are included as part of MORTCOST.

If the property is owned outright (ownhow=1), mortint will be skipped and MORTPAY should be set to zero. Similarly, there will be no endowment premium costs which will also remain as zero. However, any structural insurance or other payments should be included and MORTCOST should not be set to - 2.

**2 FRS Specification**

For each household where TENURE=1

Code            Condition

MORTCOST

**Mortgage payments**

From HOUSEHOL record get MORTINT  
 From MORTGAGE get INCMAMP1, **INCMAMP2, INCMAMP3**, INCMPPD1,  
**INCMPPD2, INCMPPD3**

Set MORTPAY equal to MORTINT

If MORTINT <>-2 and MORTPROT=1 and INCMAMP1 or **INCMAMP2 or INCMAMP3** exists and INCMPPD1, **INCMPPD2, INCMPPD3** equal to -1 or 1-11 for all mortgages then MORTPAY=MORTPAY+INCMAMP1+**INCMAMP2+INCMAMP3**

Else if OWNHOW=1 (owned outright) MORTPAY=0

**Endowment premiums**

From ENDOWMNT table get MENPOLAM, MENPOLPD

Set ENDOWPAY equal to zero

For each mortgage, if OWNHOW=2 and MENPOLPD equal to -1 or 1-11 then ENDOWPAY=total of MENPOLAM else do not change ENDOWPAY

(if OWNHOW=1, ENDOWPAY remains set as zero)

-----  
 ---



-----  
 --

### Structural insurance

From HOUSEHOL record get STRCOV, STRAMT1, STRPD1, STROTHS,  
 TENURE

set STRUINS equal to zero

If STRCOV=1 or STRCOV=3 (structural insurance only or combined with  
 furniture and contents as part of mortgage payment) and STRPD1=-1 or 1-11  
 STRUINS=STRAMT1 else do not change STRUINS

If STROTHS=1 and TENURE=1 (insurance premium paid on structure of  
 accommodation separately from any mortgage payments, owner occupiers  
 only) and STRPD2=-1 or 1-11 STRUINS=**STRUINS**+STRAMT2 else do not  
 change STRUINS

### Service payments

From OWNER record get CHARGE1-7, CHAMT1-7, CHARGE1-7

set SERVPAY=0

If CHARGE1=1 and CHARGE1=-1, or 1-11 add CHAMT1 to SERVPAY else  
 do not change SERVPAY

If CHARGE2=1 and CHARGE2=-1, or 1-11 add CHAMT2 to SERVPAY else  
 do not change SERVPAY

If CHARGE3=1 and CHARGE3=-1, or 1-11 add CHAMT3 to SERVPAY else  
 do not change SERVPAY

If CHARGE4=1 and CHARGE4=-1, or 1-11 add CHAMT4 to SERVPAY else  
 do not change SERVPAY

If CHARGE5=1 and CHARGE5=-1, or 1-11 add CHAMT5 to SERVPAY else  
 do not change SERVPAY

If CHARGE6=1 and CHARGE6=-1, or 1-11 add CHAMT6 to SERVPAY else  
 do not change SERVPAY

If CHARGE7=1 and CHARGE7=-1, or 1-11 add CHAMT7 to SERVPAY else  
 do not change SERVPAY

MORTCOST=MORTPAY+ENDOWPAY+STRUINS+SERVPAY

-1 not applicable to this case - TENURE ne 1.

-2 MORTCOST=-2 if MORTPAY is equal to -2 (-2 cases for endowpay, struins and servpay should  
 not exist)

-----  
 ---

-----  
 --

**MORTINT**

Purpose : The amount of mortgage interest paid by each household.  
 Created : 13 January 1993  
 Database Table : HOUSEHOL  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : PSM HBM ASD4A **ISM** (**Bold** indicates lead user)  
 Issued : 21 May, 2003  
 Amendments : VC - 27 January 1993 Added more groups to those picked out of tenure type.  
               : VC - 2 March 1993 Added sorting bands for tabulation.  
               : VC - 19 October 1993 Changes to specification as repayment mortgages :  
               : were calculating an average interest paid over the last 12 months and not the :  
               : last payment of interest made. See Andrew Ray's specification 13 October :  
               : 1993  
               : VC - 6 December 1993 Emphasises use of intdate to calculate mortgage :  
               : interest for repayment mortgages  
               : VC - 1 March 1994 To exclude any period codes 12 or 13  
               : JS - 6 February 1996 to exclude structural insurance payments included as  
               : part of mortgage payments  
               : JS - 12 March 1996 to update variable names and include different interest  
               : rates and MIRAS adjustments  
               : JS - 12 April 1996 to allow for contributions made from anyone outside the  
               : household  
               : VE - 5 June 1996 - Initial amendments for V32 update - adding in PEP  
               : mortgages to MORTTYPE  
               : VE - 14 June 1996 - To update mortgage interest rates for 1995-96  
               : VE - 1 July 1996 - Amended for constants being held in a separate table  
               : VE - 19 August 1996 - To amend MNTHCODE to read January - December  
               : VE - 20 November 1996 - To amend STRAMT to STRAMT1  
               : VE - 2 December 1996 - To take off any amounts relating to top-up loans  
               : unrelated to housing costs  
               : VE - 3 February 1997 - To amend the top-up section of the variable  
               : VE - 28 February 1997 - To amend MIRAS adjustment to include second  
               : mortgages for the purpose of house purchase  
               : VE - 1 May 1997 - To amend MIRAS adjustment to 15%

~~NB - This variable does not include information about top-up loans for repairing the home.~~

**1 Definition**

This variable is coded as

MORTINT      The total amount of mortgage interest payable by a household

-----  
--

- 1            Not applicable to this case
- 2            Unable to derive because of missing values

The amount of mortgage interest is derived from several variables in the HOUSEHOL, OWNER, MORTGAGE and ENDOWMNT tables. The variable TENURE in the HOUSEHOL table indicates whether the person owns the property and OWNHOW in the OWNER table indicates whether it is owned outright or with a mortgage. The variable MORTTYPE in the MORTGAGE table then indicates whether the mortgage is an endowment mortgage (including pension mortgages) or a repayment one. Once this has been established the total amount of mortgage interest can be calculated.

### Endowment Mortgages

MORINPAY holds the amount of interest paid for endowment mortgages. However, if an insurance premium or a mortgage protection policy has been included in the amount of interest recorded in MORINPAY, the amount has to be adjusted. If MENPOL = 1 and INCININT = 1 (from the ENDOWMNT table) indicating that an amount for an insurance premium is included in MORINPAY, the amount of the premium has to be deducted from the amount of mortgage interest. The amount of insurance premium is found in MENPOLAM.

Also if MORTPROT = 1 and INCMP = 1 (both from the MORTGAGE table) indicating that a mortgage protection policy is included in MORINPAY, the amount in INCMPAMT is deducted from the amount of mortgage interest.

Adjustments are also required for contributions from outside the household if mortgage interest has been quoted after these amounts.

Finally, in block g\_Insur of the questionnaire, respondents are asked whether their last payment included amounts of any insurance on the structure or contents of the accommodation (STRMORT). If yes, total amount STRAMT1 should be removed. The value of any structural insurance is added back in HSCOSTHH. **(STRAMT2 relates to insurance payments not paid as part of mortgage payments).**

### Repayment Mortgages

Repayment mortgages are slightly different ~~as the variable INTL12M only holds the average amount of interest paid over the 12 months that the mortgagee holds information for. As a result, and~~ the amount of interest paid on a repayment mortgage is calculated separately using the amount of mortgage still outstanding (MORTLEFT) multiplied by the rate of interest current for the month in which the interview took place. These interest rates are taken from the Central Statistics Office's report of Financial Statistics which will be held on the FRS database as a standard table of values and updated every year.

**The constants are held in h:\frs\dvars\docs\spreads.xls as V32Consants.** The average rate is used in preference to the basic rate because it is a better indicator of the interest rate charged on all mortgages (the basic rate excludes discounts for first time buyers and is therefore slightly higher). Note: since October 1995 the eligible mortgage interest on IS has been calculated on the basis of the standard rate which is set using the basic rather than average rate.

-----  
--

As this will calculate the total amount of mortgage interest any mortgage protection policies and/or insurance premiums may be ignored.

However, if the period code for any period is 12 or 13 (lumpsum/one-off or other period) the record must be set to unable to derive as it has not been possible to convert the amount of benefit into a weekly amount during the database conversion process. Therefore, if MORINPD, MENPOLPD, and INCMPPD = 12 or 13 MORTINT is set to -2. Cases where period codes have been skipped (where amount has been imputed) are allowed.

### Top-up loans

**Amounts borrowed for mortgage top-up loans for purposes that are unrelated to housing costs (TOPPUR=2,3,4,5 or 7) should be subtracted from the amount of mortgage interest paid by a household. For repayment mortgages, the proportion of the total amount borrowed (TOP/BORRAMT+TOP) which is made up of the top-up is calculated. The amount of the mortgage still left to pay is taken to be (1-this proportion) multiplied by MORTLEFT, assuming that the two loans are paid in tandem. For other types of mortgage, the interest on the top-up loan is calculated and subtracted from MORINPAY.**

### MIRAS

Allowance also has to be made for MIRAS which is deducted at ~~20%~~ 15% of the interest rate on the first £30,000 of mortgages taken out for the express purpose of buying a house. All payments in MORTINT are shown net of MIRAS. This means that for ENDOWMENT mortgages an adjustment is required for any payments quoted gross of MIRAS (TAXRELF=2). For REPAYMENT mortgages, where interest is calculated, an adjustment has to be applied to all cases.

**As of the release of FRS 956b, the MIRAS adjustment for second mortgages is incorrect. We have assumed that tax relief of up to £30,000 is allowed on each mortgage, when in fact from 1 August 1988 the allowance is up to £30,000 per residence. Only three cases are affected (9045071, 12595041 and 4225191) by this ie. the total value of MORTLEFT for the first and second mortgages adds to more than £30,000. We are awaiting action from PSM before amending the spec.**

### SECOND MORTGAGES

The FRS asks for details of up to two mortgages. For those buying their house with a mortgage respondents are routed to the question on the basis of "I have already asked you about the loan you had to purchase this house/flat, apart from that, do you have any OTHER mortgage or loan on this property?". This will include mortgages which have been secured on the property but which were not for house purchase (**SECMPUR=2,3,4,5 or 7**). These type of loans are not eligible for MIRAS. Rules for calculation of eligible mortgage interest for Income Support were tightened up in October 1995 and also exclude these type of loans.

~~Without sufficient information on the reason for these second loans,~~ It has been agreed that:

- i the MIRAS adjustment will only be applied to the first mortgage **and to the second mortgage where SECMPUR=1 or 6**

-----  
 --

but ii that MORTINT will continue to included interest from all loans secured on the property (this is also consistent with DoE's calculation of equity which looks at *all* loans secured on a property).

Note: those who own their property outright are also asked if they are using their property as security for a mortgage or loan of any kind but these are EXCLUDED from MORTINT.

**2 FRS Specification**

For each HOUSEHOLD

Code            Condition

**Set TOP (temporary variable) to ZERO. TOP calculates the total amount of top-up loans borrowed for purposes that are unrelated to housing costs.**

**If TOPSEQ = 1 and TOPUP = 1 and TOPPUR = 2, 3, 4, 5 or 7 then TOP = TOPAMT  
 If TOPSEQ = 2 and TOPUP = 1 and TOPPUR = 2, 3, 4, 5 or 7 then TOP = TOP + TOPAMT  
 If TOPSEQ = 3 and TOPUP = 1 and TOPPUR = 2, 3, 4, 5 or 7 then TOP = TOP + TOPAMT**

**Set INT (temporary variable) to ZERO. INT calculates the weekly interest paid on top-up loans borrowed for purposes that are unrelated to housing costs.**

**INT = (TOP \* INTRATE) / 52**

MORTINT        From HOUSEHOL table if TENURE = 1 (owns/is buying), 2 (co-ownership scheme), 3 (shared ownership) or 4 (part own part rent), get variable OWNHOW from OWNER record.

If OWNHOW = 2 (bought with mortgage or loan) process MORTGAGE record

Calculate INTRATE (temporary variable) = relevant interest rate for month of interview from standard table (look up table for basic rate relevant to month of interview: used in both endowment and repayment calculations so moved to beginning of program)

If MNTHCODE=1 ( <del>April</del> <b>January 94-95</b> )	INTRATE= <del>7.64%</del> <b>7.99%</b>	INTM1
If MNTHCODE=2 ( <del>May</del> <b>February 94-95</b> )	INTRATE= <del>7.63%</del> <b>7.97%</b>	INTM2
If MNTHCODE=3 ( <del>June</del> <b>March 94-95</b> )	INTRATE= <del>7.61%</del> <b>7.95%</b>	INTM3
If MNTHCODE=4 ( <del>July</del> <b>April 94-95</b> )	INTRATE= <del>7.59%</del> <b>7.94%</b>	INTM4
If MNTHCODE=5 ( <del>August</del> <b>May 94-95</b> )	INTRATE= <del>7.56%</del> <b>7.92%</b>	INTM5
If MNTHCODE=6 ( <del>September</del> <b>June 94-95</b> )	INTRATE= <del>7.57%</del> <b>7.90%</b>	INTM6
If MNTHCODE=7 ( <del>October</del> <b>July 94-95</b> )	INTRATE= <del>7.85%</del> <b>7.56%</b>	INTM7
If MNTHCODE=8 ( <del>November</del> <b>August 94-95</b> )	INTRATE= <del>7.83%</del> <b>7.51%</b>	INTM8
If MNTHCODE=9 ( <del>December</del> <b>September 94-95</b> )	INTRATE= <del>7.84%</del> <b>7.48%</b>	INTM9
If MNTHCODE=10 ( <del>January</del> <b>October 94-95</b> )	INTRATE= <del>7.84%</del> <b>7.28%</b>	INTM10
If MNTHCODE=11 ( <del>February</del> <b>November 94-95</b> )	INTRATE= <del>7.98%</del> <b>7.08%</b>	INTM11
If MNTHCODE=12 ( <del>March</del> <b>December 94-95</b> )	INTRATE= <del>8.00%</del> <b>7.03%</b>	INTM12

-----  
 --

From MORTGAGE table, for each mortgage get all variables

If MORTTYPE = 1 or 3 or 4 or 5 (endowment, pension mortgage, **PEP or Unit Trust mortgage** or other type of mortgage) and MORINPD not equal 12 or 13 (but may equal -1)

Calculate  $MORTINT = MORINPAY - INT$  (how much interest did you pay last time - **interest payable on any top-up loans that are unrelated to housing costs**).

**If  $MORTINT < 0$  set  $MORTINT = -2$**

If MORTPROT = 1 and  $INCMP1 = 1$  and  $INCMPPD1$  not equal 12 or 13 (but may equal -1), or  $INCMP2 = 1$  and  $INCMPPD2$  not equal 12 or 13 (but may equal -1) or  $INCMP3 = 1$  and  $INCMPPD3$  not equal 12 or 13 (but may equal -1) calculate  $MORTINT = MORINT - (INCMPAM1 + INCMPAM2 + INCMPAM3)$ . (mortgage protection policies)

From ENDOWMNT table, for each endowment policy get all variables

If  $INCININT = 1$  (insurance premium included in MORINPAY) and MENPOLPD not equal 12 or 13 (but may equal -1),  $MORTINT = MORINT - MENPOLAM$

From MORTCONT table

If  $OUTSINCL=2$  (amount of contribution to interest not included in amount mentioned earlier) and  $OUTSPD=-1$  or 1-11 then  $MORTINT=MORTINT+OUTSAMT$

If  $STRMORT=1$  and  $STRPD1=-1$ , 1-11 and  $STRAMT1 < MORTINT$  then  $MORTINT=MORTINT-STRAMT1$

Else if  $STRMORT=1$  and  $STRPD1=-1$ , 1-11 and  $STRAMT1 > MORTINT$  then do not change MORTINT

(this should only be applied to endowment mortgages because repayment mortgages are calculated and do not use respondents' answers. If, however, the amount in STRAMT1 is greater than MORTINT, assume that an error has been made and do not change MORTINT)

MIRAS adjustment is moved to the last step *after* structural mortgage payments adjustment

MIRAS adjustment to first mortgage **and second mortgage where  $SECMPUR=1$  or 6:**

If  $MORTSEQ=1$  or ( $MORTSEQ=2$  and  $SECMPUR=1$  or 6) and

If  $TAXRELF=2$  (interest quoted does not include MIRAS arrangements) and  $MORTLEFT > 30,000$

then  $MORTINT=MORTINT-((30,000*INTRATE*0.20.15)/52)$

-----  
 --

Else if TAXRELF=2 and MORTLEFT<=30,000  
 then MORTINT=MORTINT-((MORTLEFT\*INTRATE\*~~0.20.15~~)/52)

(MORINPAY will be held as a weekly amount, but the MIRAS adjustment also has to be converted)

If MORTTYPE = 2 (repayment mortgage),

**Set PROP (temporary variable) to ZERO. PROP calculates the proportion of the total amount borrowed which is made up of the top-up loan.**

**Set REMAINS (temporary variable) to ZERO. REMAINS calculates the amount remaining to pay on the mortgage multiplied by (1-PROP).**

**If TOP>0, PROP=TOP/(BORRAMT+TOP)**

**If TOPUP = 2 then REMAINS = MORTLEFT**

**Else if TOPUP = 1 then REMAINS = MORTLEFT \* (1-PROP)**

Calculate MORTINT = (~~MORTLEFT~~ **REMAINS** \* INTRATE)/52

MIRAS adjustment to first mortgage **and second mortgage where SECMPUR=1 or 6:**

If MORTSEQ=1 **or (MORTSEQ=2 and SECMPUR=1 or 6)** and

If ~~MORTLEFT~~ **REMAINS** >30,000

then MORTINT=MORTINT-((30,000\*INTRATE\*~~0.20.15~~)/52)

Else if ~~MORTLEFT~~ **REMAINS** <=30,000

then MORTINT=MORTINT-((~~MORTLEFT~~ **REMAINS** \*INTRATE\*~~0.20.15~~)/52)

-1 Not applicable to this case - property not owned with a mortgage.

-2 Unable to derive variable because of any missing values or MORINPD, INCMPPD or MENPOLPD = 12 or 13 (but -1 OK).

NB. FRS does not collect the rate of interest charged on a mortgage the interviewee's answer is taken to be correct.

-----  
 --

**NDDCTB**

Purpose : To calculate total amount of non-dependent deductions for each household :  
 using HBAI specifications - for Council Tax Benefit  
 Created : VC - 28 February 1994  
 Database Table : HOUSEHOL  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables :  
     Children :  
     Parents : DEPDEDS INDINC - both are derived variables  
 Core variable/user : ASD6A  
 Amendments : VC - 28 February 1994 To reflect changes to version 30  
               : JS - 3 April 1996 to reflect changes for V31  
               : VE - 19 April 1996 - to replace GROSSINC withINDINC  
               : VE - 4 June 1996 - No initial amendments needed for V32 update  
               : VE - 27 June 1996 - To update income levels  
               : VE - 1 July 1996 - Amended for constants being held in a separate table  
 Issued : 21 May 2003

**1 Definition**

This variable is coded as

NDDCTB This is the total amount of non-dependent deductions for each household using the HBAI specification. This specification has been created from Andrew Ray's specification dated 12 November 1993 (see attached). **The constants and income levels are held in the table V32Constants in h:\frs\dvars\docs\spreads.xls.**

**2 FRS Specification**

For each HOUSEHOL table - set NDDCTB to zero, then

Process each BENUNIT table,

If BENUNIT greater than 1 (ie not head of household BU), get DEPDEDS derived variable.

If DEPDEDS = 6 or 8 (over 25 in receipt of IS, or any other person over age of 18), calculate  $NDDCTB = NDDCTB + 4.15 \cdot C1CTB$

Else if DEPDEDS = 3 (18+ working over 16 hours), process each ADULT table

If  $INDINC < 408$  ~~414~~ **INCLEV**, calculate  $NDDCTB = NDDCTB + 4.15 \cdot C2CTB$

-----  
 ---



-----  
--

Else calculate  $NDDCTB = NDDCTB + 2.30 \text{ C3CTB}$

Else if  $BENUNIT = 1$ ,

Process each ADULT table, reset NDDCTB back to zero if any of the following conditions are met for either adult in the BU

Process each BENEFITS table for each adult in BU 1

If  $BENEFIT = 1$  or  $11$  (receiving DLA care or AA) calculate  $NDDCTB = 0$

If  $SPCREG1 = 1$  (is blind), calculate  $NDDCTB = 0$

If  $SUBLTAMT > 0$  (receives a sublet income) AND  $NOUNITS = 2$  (number of BUs in a household), calculate  $NDDCTB = 0$

### 3 Results

To show the number of households falling into the following set of categories

Up to £4  
£4 - £8  
£8 - £12  
£12 - £16  
£16 - £20  
£20 - £24  
£24 - £28  
£32 - £36  
£36 - £40  
Over £40

### 4 Test Cases

-----  
---

-----  
 --

**NDDISHC**

Purpose : To calculate total amount of non-dependent deductions for each household :  
 using HBAI specifications - for IS housing costs  
 Created : VC - 28 February 1994  
 Database Table : HOUSEHOL  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables :  
     Children :  
     Parents : DEPDEDS INDINC- both are derived variables  
 Core variable/user : ASD6A  
 Amendments : VC - 28 February 1994 To reflect changes to version 30  
               : JS - 3 April 1996 to reflect changes for V31  
               : VE - 19 April 1996 to replace GROSSINC with INDINC  
               : VE - 4 June 1996 - No initial amendments needed for V32 update  
               : VE 27 June 1996 - To update income levels and constants  
               : VE - 1 July 1996 - Amended for constants being held in a separate table  
 Issued : 21 May 2003

**1 Definition**

This variable is coded as

NDDISHC This is the total amount of non-dependent deductions for each household using the HBAI specification. This specification has been created from Andrew Ray's specification dated 12 November 1993 (see attached). **The constants and income levels are held in the table V32Constants in h:\frs\dvars\docs\spreads.xls.**

**2 FRS Specification**

For each HOUSEHOL table - set NDDISHC to zero, then

Process each BENUNIT table,

If BENUNIT greater than 1 (ie not head of household BU), get DEPDEDS derived variable.

If DEPDEDS = 6 or 8 (over 25 in receipt of IS, or any other person over age of 18), calculate NDDISHC = NDDISHC + ~~5~~ **C1IS**

Else if DEPDEDS = 3 (18+ working over 16 hours), process each ADULT table

If INDINC < ~~72-74~~ **INCL1IS**, calculate NDDISHC = NDDISHC + ~~5~~ **C1IS**

Else if INDINC < ~~108-111~~ **INCL2IS**, calculate NDDISHC = NDDISHC + ~~9-10~~ **C2IS**

-----  
 ---

-----  
--

Else if ~~INDINC < 139~~ ~~145 INCL3IS~~, calculate  $NDDISHC = NDDISHC +$   
~~13~~ ~~14~~ ~~C3IS~~

Else if ~~INDINC >= 139~~ ~~145 INCL3IS~~, calculate  $NDDISHC = NDDISHC +$   
~~25~~ ~~30~~ ~~C4IS~~

Else if BENUNIT = 1,

Process each ADULT table, reset NDDISHC back to zero if any of the following conditions are met for either adult in the BU

Process each BENEFITS table for each adult in BU 1

If BENEFIT = 1 or 11 (receiving DLA care or AA) calculate  
 $NDDISHC = 0$

If SPCREG1 = 1 (is blind), calculate  $NDDISHC = 0$

If SUBLTAMT > 0 (receives a sublet income) and NOUNITS = 2 (number of BUs in household from HOUSEHOL table), calculate  $NDDISHC = 0$

### 3 Results

To show the number of households falling into the following set of categories

Up to £4  
£4 - £8  
£8 - £12  
£12 - £16  
£16 - £20  
£20 - £24  
£24 - £28  
£32 - £36  
£36 - £40  
Over £40

### 4 Test Cases

-----  
---

-----  
 --

**NDDRENTR**

Purpose : To calculate total amount of non-dependent deductions for each household :  
 using HBAI specifications - for rent rebate.  
 Created : VC - 28 February 1994  
 Database Table : HOUSEHOL  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables :  
     Children :  
     Parents : DEPDEDS INDINC - both are derived variables  
 Core variable/user : ASD6A  
 Amendments : VC - 28 February 1994 To reflect changes to version 30  
               VE - 19 April 1996 - to replace GROSSINC with INDINC  
               VE - 4 June 1996 - No amendments needed for initial V32 update  
               VE - 27 June 1996 - To update income levels and constants  
               VE - 1 July 1996 - Amended for constants being held in a separate table.  
 Issued : 21 May 2003

**1 Definition**

This variable is coded as

NDDRENTR This is the total amount of non-dependent deductions for each household using the HBAI specification. This specification has been created from Andrew Ray's specification dated 12 November 1993 (see attached). **The constants and income levels are held in the table V32Constants in h:\frs\dvars\docs\spreads.xls.**

**2 FRS Specification**

For each HOUSEHOL table - set NDDRENTR to zero, then

Process each BENUNIT table,

If BENUNIT greater than 1 (ie not head of household BU), get DEPDEDS derived variable.

If DEPDEDS = 6 or 8 (over 25 in receipt of IS, or any other person over age of 18), calculate NDDRENTR = NDDRENTR + ~~5~~**C1RT**

Else if DEPDEDS = 3 (18+ working over 16 hours), process each ADULT table

If ~~INDINC < 72~~ ~~74~~ **INCL1RT**, calculate NDDRENTR = NDDRENTR + ~~5~~**C1RT**

-----  
 ---

-----  
 --

Else if ~~INDINC < 108~~ ~~111~~ **INCL2RT**, calculate  $NDDRENTR = NDDRENTR + 9$  ~~10~~ **C2RT**

Else if ~~INDINC < 139~~ ~~145~~ **INCL3RT**, calculate  $NDDRENTR = NDDRENTR + 13$  ~~14~~ **C3RT**

Else if ~~INDINC >= 139~~ ~~145~~ **INCL3RT**, calculate  $NDDRENTR = NDDRENTR + 25$  ~~30~~ **C4RT**.

Else if BENUNIT = 1,

Process each ADULT table, reset NDDRENTR back to zero if any of the following conditions are met for either adult in the BU

Process each BENEFITS table for each adult in BU 1

If BENEFIT = 1 or 11 (receiving DLA care or AA) calculate  $NDDRENTR = 0$

If SPCREG1 = 1 (is blind), calculate  $NDDRENTR = 0$

If SUBLTAMT > 0 (receives a sublet income) and NOUNITS = 2 (number of BUs in a household from HOUSEHOL table), calculate  $NDDRENTR = 0$

### 3 Results

To show the number of households falling into the following set of categories

Up to £4  
 £4 - £8  
 £8 - £12  
 £12 - £16  
 £16 - £20  
 £20 - £24  
 £24 - £28  
 £32 - £36  
 £36 - £40  
 Over £40

### 4 Test Cases

-----  
 ---

-----  
 --

**NETOCPEN**

Purpose : To show the amount of income received from all forms of occupational pensions from former employers net of tax (REVISED OCCUPPEN) :

Created : January 1995

Database Table : ADULT

Minimum Value : 0

Maximum Value :

Units : Real

Validations :

Related Variables : OCCUPPEN

Children :

Parents :

Core variable/user : HBM

Amendments : JS - 21 February 1996 to allow for skipped values where amounts have been imputed

: VE - 23 May 1996 - No initial amendments needed for V32 update

Issue date : 6 June 1996

**1 Definition**

This variable is coded as

- NETOCPEN This is the total gross amount received from **all** occupational pensions paid by a person's former employer. It includes any occupational pension which is being paid by an overseas government/company paid in foreign currency (converted to sterling).
- 1 Not applicable to this case - adults who do not have occupational pensions
- 2 Unable to derive variable

NETOCPEN will be derived from variables PENTYPE, PENPAY, ROYAL3 AND ROYYR3. PENTYPE and PENPAY indicate that the person is in receipt of a pension from a previous employer. The variable PENTYPE is a database variable created to indicate which of the incomes listed in ANYPEN the person has and which one this particular record refers to. A person may have up to 5 occupational pensions and as a result NETOCPEN must be the total of all pensions.

To get the net amount, the amount held in PENPAY must be looked at in relation to the variables which ask if any tax has been deducted at source or whether any other deductions had been taken into account when the original amount of PENPAY had been given. These variables are PENTAX (has tax been deducted at source y/n), PTINC (was the original amount before or after this was deducted 2 = after), PTAMT (amount deducted), PENOTH (any other deductions y/n), POINC (original amount before/after deduction 2 = after) and POAMT (amount deducted). Consequently, if the original amount in PENPAY was after either of these amounts had been deducted they must be added back to get the gross amount.

-----  
 ---

-----  
 --

ROYAL3 and ROYYR3 deal with an occupational pension paid by an overseas government or company which is paid in a foreign currency. This amount is taken to be a gross amount for OCCUPPEN but must be considered to be net of tax for NETOCPEN as the questionnaire does not collect information regarding any deductions from the amount held in ROYYR3.

ROYAL3 and ROYYR3 are also database variables which hold the information obtained from questions royal and royyr (NB - royal1 = royalties, royal2 = sleeping partners and royal3 = occ pen from o/s govt). However, if the period code for the pension is 12 or 13 (lumpsum/one-off or other period) the record must be set to unable to derive as it has not been possible to convert the amount of benefit into a weekly amount during the database conversion process. Therefore, if PENPD = 12 or 13 NETOCPEN is set to -2.

**2 FRS Specification**

For each ADULT with a pension record

<u>Code</u>	<u>Condition</u>
NETOCPEN	<p>From PENSION table, for each pension calculate a temporary variable OCCUP</p> <p>If PENTYPE = 1 (indicating an occupational pension is being received) and PENPD equal to -1 or 1-11, get variables PENPAY, PENTAX, PTAMT, PTINC, PENOTH, POAMT and POINC.</p> <p>Compute OCCUP = PENPAY.</p> <p>If PENTAX = equal to 1 or has been skipped (tax deducted at source) and PTINC = 2 (original amount declared <u>after</u> amount of tax deducted) do not change OCCUP</p> <p>If PENTAX=1 (tax deducted at source) and PTINC=1 and PTAMT exists (original amount before amount of tax deducted) subtract PTAMT from OCCUP else do not change OCCUP</p> <p>If PENOTH = 1 (other deductions) and POINC = 2 (original amount declared <u>after</u> deduction), add POAMT to OCCUP.</p> <p>From ADULT table</p> <p>If ROYAL3 = 1 (pension from an overseas government) get amount from ROYYR3 and add to OCCUP.</p> <p>NETOCPEN will then be the sum of all occurrences of OCCUP as each adult is able to have up to 5 occupational pensions.</p>
-1	Not applicable to this case -
-2	Unable to derive because any of the above variables are missing or PENPD = 12 or 13.

-----  
 ---

-----  
 --

**OCCUPNUM**

Purpose : To show the total number of occupational pensions a person receives.  
 Created : VC - 13 July 1993  
 Database Table : ADULT  
 Minimum Value : 0  
 Maximum Value : 6  
 Units : Integer  
 Validations :  
 Related Variables : OCCUPPEN - total amount of occupational pensions  
     Children :  
     Parents :  
 Core variable/user : PSM  
 Amendments : VC - 17 February 1994 Amended to reflect version 30 changes  
               : VE - 23 May 1996 - No initial amendments needed for V32 update  
 Issue date : 6 June 1996

**1 Definition**

This variable is coded as

OCCUPNUM The total number of occupational pensions a person receives from both a former employer or from any pensions from overseas governments or companies.

-1 Not applicable to this case

-2 Unable to derive due to missing values.

This variable is a simple count of the number of occupational pensions a person receives and is derived from processing the PENSION record in the database and counting the number of these records where PENTYPE = 1 (PENTYPE is a database variable indicating a record holding information about occupational pensions). A person may have up to five of this type of pension record.

The variable should also include pensions paid by overseas governments or companies. Information about these pensions are to be found where ROYAL3 = 1. ROYAL3 is a database variable produced from the question ROYAL where royal1 = royalties, royal2 = sleeping partners and royal3 = occupational pensions. Only one extra pension need be counted as the questionnaire only collects information about one of these pensions.

**2 FRS Specification**

For each adult

Code

Condition

OCCUPNUM Set OCCUPNUM to zero

-----  
 ---



-----  
--

From PENSION table, count number of records where PENTYPE = 1

From ADULT record, if ROYAL3 = 1 add one to total number of pension records calculated above.

**3 Results**

Tabulation is required to show the number of people by the number of pensions they receive sorted into bands of

- No occupational pensions
- One pension
- Two pensions
- Three pensions
- Four pensions
- Five pensions
- Six or more pensions

**4 Test Cases**

None produced yet - to be added at a later date.

-----  
---

-----  
 --

**OCCUPPEN**

Purpose : To show the amount of income received from all forms of occupational pensions from former employers :

Created : 21 January 1993

Database Table : ADULT

Minimum Value : 0

Maximum Value :

Units : Real

Validations :

Related Variables :

    Children :

    Parents :

Core variable/user : ISM HBM FCM PSM

Amendments : VC - 2 March 1993 change to multi response questions

              : VC - 27 April 1993. To expand definition to include descriptions of all questions/database variables used for the derived variable. :

              : VC - 13 July 1993 Amended to add back any tax deducted at source if the amount declared in penpay is net of tax. This is now a gross income variable : although the amount collected by royrr2 cannot be said to be either gross or : net of tax.

              : VC - 17 February 1994 Amended to reflect version 30 changes

              : VC - 1 March 1994 To exclude any period codes 12 or 13.

              : JS - 21 February 1996 to allow for skipped values where amounts have been imputed

              : VE - 23 May 1996 - No amendments needed for initial V32 update

Issue date :21 May, 2003

**1 Definition**

This variable is coded as

OCCUPPEN This is the total gross amount received from any occupational pensions paid by a person's former employer. It includes any occupational pension which is being paid by an overseas government/company paid in foreign currency (converted to sterling).

-1 Not applicable to this case - adults who do not have occupational pensions

-2 Unable to derive variable

OCCUPPEN will be derived from variables PENTYPE, PENPAY, ROYAL3 AND ROYYR3. PENTYPE and PENPAY indicate that the person is in receipt of a pension from a previous employer. The variable PENTYPE is a database variable created to indicate which of the incomes listed in ANYPEN the person has and which one this particular record refers to. A person may have up to 5 occupational pensions and as a result OCCUPPEN must be the total of all pensions.

-----  
 ---

-----  
 --

To get the gross amount, the amount held in PENPAY must be looked at in relation to the variables which ask if any tax has been deducted at source or whether any other deductions had been taken into account when the original amount of PENPAY had been given. These variables are PENTAX (has tax been deducted at source y/n), PTINC (was the original amount before or after this was deducted 2 = after), PTAMT (amount deducted), PENOTH (any other deductions y/n), POINC (original amount before/after deduction 2 = after) and POAMT (amount deducted). Consequently, if the original amount in PENPAY was after either of these amounts had been deducted they must be added back to get the gross amount.

ROYAL3 and ROYYR3 deal with an occupational pension paid by an overseas government or company which is paid in a foreign currency. This amount has to be taken to be a gross amount as the questionnaire does not collect information regarding any deductions from the amount held in ROYYR3

ROYAL3 and ROYYR3 are also database variables which hold the information obtained from questions royal and royyr (NB - royal1 = royalties, royal2 = sleeping partners and royal3 = occ pen from o/s govt. However, if the period code for the pension is 12 or 13 (lumpsum/one-off or other period) the record must be set to unable to derive as it has not been possible to convert the amount of benefit into a weekly amount during the database conversion process. Therefore, if PENPD = 12 or 13 OCCUPPEN is set to -2.

Where values have been imputed, answers to questions which follow will remain as skipped. To overcome this problem, the specification needs to be amended to allow:

- i skipped values of PENPD and PENTAX where PENPAY has been imputed (PENTAX assumed to be equal to 2 for these cases)
- ii missing PTAMT where PENTAX has been imputed to yes
- iii missing PTINC where PTAMT has been skipped (assumed to be after)

## 2 FRS Specification

For each ADULT with a pension record

Code                      Condition

OCCUPPEN      From PENSION table, for each pension

If PENTYPE = 1 (indicating an occupational pension is being received) and PENPD equal to -1 or 1-11, get variables PENPAY, PENTAX, PTAMT, PTINC, PENOTH, POAMT and POINC.

Compute OCCUP = PENPAY.

If PENTAX exists and = 1 (tax deducted at source) and PTINC exists and equals to 2 (original amount declared after amount of tax deducted), and PTAMT exists add PTAMT to OCCUP (otherwise, do not change OCCUP).

-----  
 ---

-----  
--

If PENOTH = 1 (other deductions) and POINC= 2 (original amount declared after deduction), add POAMT to OCCUP.

From ADULT table

If ROYAL3 = 1 (pension from an overseas government) get amount from ROYYR3 and add to OCCUP.

OCCUPPEN will then be the sum of all occurrences of OCCUP as each adult is able to have up to 5 occupational pensions.

-1 Not applicable to this case

-2 Unable to derive because any of the above variables are missing or PENPD = 12 or 13.

### 3 Results

Tabulation is required to show the number of people with occupational pensions by the total amount of pension received sorted into the following bands

Under £25  
£25 - £50  
£50 - £75  
£75 - £100  
£100 - £125  
£125 - £150  
£150 - £175  
£175 - £200  
£200 - £250  
£250 and over

### 4 Test Cases

None as yet

-----  
---

## ONS

Purpose : To indicate in which Family group (based on ONS classification) the interviewee lives.

Created : 11 June 1997

Database Table :

Minimum Value : 1

Maximum Value : 6

Units : Integer

Validations :

Related Variables :

    Children :

    Parents : LAC

Lead user: ASD3E

Issue date : 21 May, 2003

### 1 Definition

This variable is coded as

- 1 Rural Areas
- 2 Prospering Areas
- 3 Maturer Areas
- 4 Urban Areas
- 5 Mining and Industrial Areas
- 6 Inner London

-2 Unable to derive as Local Authority Code is missing (there should be no such values in the data set)

This variable is taken from the ONS classification of local authorities. It is the final stage of a three level hierarchy, which is termed a Family.

More information on this classification is available in the publication by Meryll Wallace and Chris Denham *The ONS classification of local and health authorities of Great Britain* (TSO, 1996), ISBN 0 11 6916540. The intention is that the classification should be readily accessible and widely used, provided that source and copyright are acknowledged. Any enquiries should be put to ONS.

-----  
--

**PACCTYPE**

Purpose : To indicate the number of households in any specific accommodation type for use in the FRS publication.  
Created : 29 January 1996 (although similar coding existed for 1993/94 publication)  
Database Table : HOUSEHOL  
Minimum Value : 1  
Maximum Value : 5  
Units : Integer  
Validations :  
Related Variables :  
    Children :  
    Parents :  
Core variable/user : ASD4A  
Issue date : 21 May 2003  
Amendments : VE - 5 June 1996 - No initial amendments needed for V32 update

**1 Definition**

ACCTYPE is a cut down version of TYPEACC in the household table. A completely new DV is probably not necessary. It is coded as:

- 1 Detached
- 2 Semi-detached
- 3 Terraced (including end of terrace)
- 4 Flat/maisonnette (including part of house/converted flat etc)
- 5 Other

**2 FRS Specification**

- 1 ACCTYPE=1
- 2 ACCTYPE=2
- 3 ACCTYPE=3
- 4 ACCTYPE=4 or 5
- 5 ACCTYPE=6

-----  
---

-----  
 --

**PTENTYPE**

Purpose : To indicate the number of households in any specific tenure type for use in the FRS publication.  
 Created : 29 January 1996 (although coding existed for 1993/94 publication)  
 Database Table : HOUSEHOL  
 Minimum Value : 1  
 Maximum Value : 5  
 Units : Integer  
 Validations :  
 Related Variables : TENTYPE  
     Children :  
     Parents :  
 Core variable/user : Publication  
 Issue date : 21 May 2003  
 Amendments : JS - 21 February 1996 - to allow for missing values where variables have been imputed  
               : JS - 15 March 1996 to bring in line with SEH categories (at request of ASD3F)  
               : VE - 19 August 1996 - To amend so that part owner/part renter cases are now included with cases buying with a mortgage  
               : VE - 10 October 1996 - To bring in line with Harmonisation of Outputs categories

**1 Definition**

PTENTYPE is derived using variables TENURE (household record), FURNISH and **LANDLORD (renter)** OWNHOW (owner). Codes are effectively a recode of the TENTYPE breakdown. ~~Rent free category (TENTYPE=7) is now put in with renters. (All cases where TENTYPE=7 and LANDLORD=1, 2 or 3 will be edited to rebate cases.)~~ FURNISH is not asked for TENTYPE=8 so assumed to be rented furnished.

This variable is coded as

- 1 Rented ~~unfurnished from local authority council~~ **unfurnished from local authority council** (including part owners/part renters)
- 2 ~~Rented furnished (including part own/part rent)~~ **Rented from housing association**
- 3 ~~Rent free (no cases are expected)~~ **Rented privately unfurnished - rent free**
- 4 **Rented privately - furnished rent paid**
- 4 5 Owned with mortgage (including co- and shared ownership)
- 5 6 Owned outright

-----  
 ---

-----  
--

-2      unable to derive due to missing values

**2      FRS Specification**

1      TENTYPE=1

2      TENTYPE=2

3      TENTYPE=3 or (TENTYPE=7 and FURNISH=2)

4      TENTYPE=4 or ~~TENTYPE=3~~ or TENTYPE=8 or (TENTYPE=7 and (FURNISH=1 or FURNISH=1))

~~4 5~~    If ((TENURE=1 or TENURE=2 or TENURE=3 or **TENURE=4**) and OWNHOW=2)  
~~TENTYPE=5,9,10~~

~~5 6~~    If ((TENURE=1 or TENURE=2 or TENURE=3 or **TENURE=4**) and (OWNHOW=1 or **OWNHOW=-1**))  
~~TENTYPE=6~~

-----  
---



-----  
 --

**STDREGN**

Purpose : To indicate in which standard region the interviewee lives.  
 Created : 18 January 1993  
 Database Table : HOUSEHOL  
 Minimum Value : 1  
 Maximum Value : 12  
 Units : Integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : ISM HBM PSM **ASD6A** (**Bold** indicates lead user)  
 Amendments : VC - 2 March 1993. Revised list of local authority codes received.  
               : VC - 26 April 1993. South Somerset was in incorrect region moved to region :  
               7. Amended description of region one to North (including Cumbria).  
               : VC - 27 April 1993. to extend the definition to include meanings of each :  
               question/database variable.  
               : VC - 13 July 1993. Amended to same order produced by FES  
               : VC - 6 August 1993. Moved Bromsgrove from region 5 to region 8, corrected :  
               South Hertfordshire to South Herefordshire  
               : VE - 4 June 1996 - No amendments needed for intitial V32 update  
 Issue date : 17 July, 1996

NB - There are currently no codes for the Isles of Scilly or the New Towns

**1 Definition**

This variable is coded as

- 1 North (including Cumbria)
- 2 Yorkshire and Humberside
- 3 North West
- 4 East Midlands
- 5 West Midlands
- 6 East Anglia
- 7 Greater London
- 8 South East - excluding London
- 9 South West
- 10 Wales
- 11 Scotland
- 1 Not applicable to this case
- 2 Unable to derive as variable LAC is missing

-----  
 ---

-----  
--

The standard region is derived from the Local Authority Code in the variable LAC. The standard regions have been provided by Ms Odwell and are slightly different to the original FES definition as Northern Ireland is not included. The order in which the regions are coded are also different.

**2 FRS Specification**

Code each region according to the attached list.

-1 Not applicable to this case (shouldn't be any)

-2 Unable to derive as the variable LAC is missing.

NB - These LA codes are unique to FRS.

**3 Results**

Tabulation to show the number of households in each standard region

**4 Test Cases**

None

-----  
---

**FAMILY RESOURCES SURVEY**

**DERIVED VARIABLE SPECIFICATION**

-----  
 --

STANDARD REGION CODE	LAC AUTHORITY	LOCAL
1	668	Hartlepool
1	772	Blyth Valley
1	771	Berwick-upon-Tweed
1	776	Gateshead
1	867	Newcastle upon Tyne
1	573	Barrow-in-Furness
1	576	Eden
1	669	<del>Langbaugh-on-Tees</del> <b>Redar and Cleveland</b>
1	674	Derwentside
1	672	Chester-le-Street
1	773	Castle Morpeth
1	574	Carlisle
1	575	Copeland
1	673	Darlington
1	675	Durham
1	868	North Tyneside
1	676	Easington
1	670	Middlesbrough
1	768	Teesdale
1	774	Tynedale
1	870	Sunderland
1	671	Stockton-on-Tees
1	869	South Tyneside
1	767	Sedgefield
1	667	South Lakeland
1	572	Allerdale
1	775	Wansbeck
1	770	Alnwick
1	769	Wear Valley

---

2	972	East Yorkshire
2	969	East Yorks Borough of Beverley
2	971	Cleethorpes
2	174	Wakefield
2	167	Doncaster
2	068	Craven
2	973	Glanford
2	171	Calderdale
2	173	Leeds
2	067	Scunthorpe
2	073	Scarborough
2	071	Richmondshire

-----  
 ---  
 -3-

**FAMILY RESOURCES SURVEY****DERIVED VARIABLE SPECIFICATION**

-----  
--

2	072	Ryedale
2	172	Kirklees
2	074	Selby
2	069	Hambleton
2	974	Great Grimsby
2	070	Harrogate
2	169	Sheffield
2	976	Kingston upon Hull
2	975	Holderness
2	168	Rotherham
2	075	York
2	170	Bradford
2	076	Barnsley
2	970	Boothferry

---

3	267	Crewe and Nantwich
3	474	South Ribble
3	373	Blackburn
3	467	Fylde
3	374	Blackpool
3	369	Stockport
3	475	West Lancashire
3	569	St Helens
3	476	Wyre
3	276	Oldham
3	567	Knowsley
3	469	Lancaster
3	272	Warrington
3	468	Hyndburn
3	470	Pendle
3	176	Congleton
3	269	Halton
3	370	Tameside
3	568	Liverpool
3	372	Wigan
3	368	Salford
3	274	Bury
3	375	Burnley
3	571	Wirral
3	473	Rossendale
3	471	Preston
3	268	Ellesmere Port and Neston
3	273	Bolton
3	376	Chorley
3	367	Rochdale
3	371	Trafford

-----  
---

**FAMILY RESOURCES SURVEY****DERIVED VARIABLE SPECIFICATION**

-----  
--

3	175	Chester
3	270	Macclesfield
3	472	Ribble Valley
3	271	Vale Royal
3	275	Manchester
3	570	Sefton

---

4	967	Nottingham
4	765	Daventry
4	562	Derby
4	657	Derbyshire Dales
4	866	Newark and Sherwood
4	858	Northampton
4	565	North East Derbyshire
4	764	Corby
4	664	North West Leicestershire
4	658	Blaby
4	665	Oadby and Wigston
4	660	Harborough
4	564	High Peak
4	766	East Northamptonshire
4	758	East Lindsey
4	563	Erewash
4	666	Rutland
4	968	Rushcliffe
4	861	Ashfield
4	757	Boston
4	560	Bolsover
4	864	Gedling
4	859	South Northamptonshire
4	762	South Kesteven
4	761	South Holland
4	566	South Derbyshire
4	760	North Kesteven
4	865	Mansfield
4	763	West Lindsey
4	860	Wellingborough
4	857	Kettering
4	759	Lincoln
4	862	Bassetlaw
4	662	Leicester
4	863	Broxtowe
4	659	Charnwood
4	663	Melton
4	661	Hinckley and Bosworth

-----  
---

**FAMILY RESOURCES SURVEY**

**DERIVED VARIABLE SPECIFICATION**

-----  
 --

4	561	Chesterfield
4	559	Amber Valley

---

5	357	The Wrekin
5	466	Solihull
5	265	Shrewsbury and Atcham
5	166	Malvern Hills
5	358	Cannock Chase
5	465	Sandwell
5	558	Wolverhampton
5	359	East Staffordshire
5	366	Tamworth
5	266	South Shropshire
5	261	Wyre Forest
5	363	Stafford
5	364	Staffordshire Moorlands
5	365	Stoke-on-Trent
5	463	Coventry
5	362	South Staffordshire
5	165	Leominster
5	260	Wychavon
5	258	South Herefordshire
5	459	Rugby
5	360	Lichfield
5	464	Dudley
5	259	Worcester
5	460	Stratford-on-Avon
5	257	Redditch
5	262	Bridgnorth
5	164	Hereford
5	263	North Shropshire
5	557	Walsall
5	462	Birmingham
5	361	Newcastle-under-Lyme
5	461	Warwick
5	264	Oswestry
5	458	Nuneaton & Bedworth
5	457	North Warwickshire

---

6	248	South Cambridgeshire
6	154	East Cambridgeshire
6	348	Ipswich
6	155	Fenland
6	253	North Norfolk

-----  
 ---

**FAMILY RESOURCES SURVEY****DERIVED VARIABLE SPECIFICATION**

-----  
--

6	153	Cambridge
6	347	Forest Heath
6	255	Kings Lynn and West Norfolk
6	250	Broadland
6	247	Peterborough
6	254	South Norfolk
6	352	Waveney
6	349	Mid Suffolk
6	351	Suffolk Coastal
6	256	Babergh
6	350	St Edmundsbury
6	251	Great Yarmouth
6	156	Huntingdonshire
6	252	Norwich
6	249	Breckland

---

7	551	Bromley
7	554	Enfield
7	648	Hillingdon
7	548	Barnet
7	451	Lewisham
7	650	Kingston upon Thames
7	556	Harrow
7	653	Richmond upon Thames
7	356	Hammersmith and Fulham
7	549	Bexley
7	649	Hounslow
7	555	Greenwich
7	355	Hackney
7	547	Barking and Dagenham
7	651	Merton
7	647	Havering
7	447	Haringey
7	449	Kensington and Chelsea
7	448	Islington
7	452	Newham
7	652	Redbridge
7	450	Lambeth
7	553	Ealing
7	456	Westminster, City of
7	454	Tower Hamlets
7	552	Croydon
7	655	Waltham Forest
7	453	Southwark
7	455	Wandsworth

-----  
---

**FAMILY RESOURCES SURVEY**

**DERIVED VARIABLE SPECIFICATION**

-----  
 --

7	550	Brent
7	654	Sutton
7	354	Camden
7	353	City of London

---

5	163	Bromsgrove
8	539	Thurrock
8	146	Mid Bedfordshire
8	048	Mole Valley
8	948	Tonbridge and Malling
8	540	Uttlesford
8	056	Adur
8	949	Tunbridge Wells
8	541	Basingstoke and Deane
8	337	Milton Keynes
8	953	Vale Of White Horse
8	742	Watford
8	055	Woking
8	243	Windsor and Maidenhead
8	145	Luton
8	244	Wokingham
8	344	Lewes
8	339	Wycombe
8	152	Worthing
8	842	Maidstone
8	643	Winchester
8	346	Wealden
8	054	Waverley
8	240	Newbury
8	743	Welwyn Hatfield
8	744	Medina
8	445	Maldon
8	954	West Oxfordshire
8	151	Mid Sussex
8	437	Basildon
8	242	Slough
8	241	Reading
8	845	Shepway
8	639	Portsmouth
8	147	Arun
8	238	South Bedfordshire
8	952	South Oxfordshire
8	049	Reigate and Banstead
8	844	Sevenoaks
8	446	Rochford

-----  
 ---



**FAMILY RESOURCES SURVEY****DERIVED VARIABLE SPECIFICATION**

-----  
--

8	345	Rother
8	640	Rushmoor
8	843	Rochester upon Medway
8	746	Ashford
8	245	Aylesbury Vale
8	951	Oxford
8	338	South Buckinghamshire
8	053	Tandridge
8	846	Swale
8	052	Surrey Heath
8	538	Tendring
8	642	Test Valley
8	947	Thanet
8	050	Runnymede
8	237	North Bedfordshire
8	740	Stevenage
8	641	Southampton
8	745	South Wight
8	537	Southend-on-Sea
8	051	Spelthorne
8	738	North Hertfordshire
8	739	St Albans
8	741	Three Rivers
8	638	New Forest
8	443	Epping Forest
8	637	Havant
8	956	Epsom and Ewell
8	837	Canterbury
8	955	Elmbridge
8	737	Hertsmere
8	543	Eastleigh
8	246	Chiltern
8	343	Hove
8	150	Horsham
8	544	Fareham
8	950	Cherwell
8	340	Brighton
8	440	Castle Point
8	841	Gravesham
8	545	Gosport
8	047	Guildford
8	840	Gillingham
8	342	Hastings
8	546	Hart
8	444	Harlow
8	441	Chelmsford
8	341	Eastbourne
8	148	Chichester

-----  
---

**FAMILY RESOURCES SURVEY**

**DERIVED VARIABLE SPECIFICATION**

-----  
 --

8	239	Bracknell Forest
8	542	East Hampshire
8	439	Brentwood
8	838	Dartford
8	438	Braintree
8	442	Colchester
8	645	Dacorum
8	149	Crawley
8	644	Broxbourne
8	646	East Hertfordshire
8	839	Dover

---

9	157	South Somerset (aka Yeovil)
9	853	Plymouth
9	750	Wansdyke
9	059	Forest of Dean
9	962	Poole
9	855	Teignbridge
9	159	North Wiltshire
9	058	Cotswold
9	749	Northavon
9	060	Gloucester
9	065	Taunton Deane
9	756	Penwith
9	963	Purbeck
9	753	Carrick
9	966	East Dorset (aka Wimborne)
9	960	Christchurch
9	064	Sedgemoor
9	957	Torridge
9	856	Torbay
9	849	East Devon
9	959	Bournemouth
9	160	Salisbury
9	057	Cheltenham
9	062	Tewkesbury
9	850	Exeter
9	161	Thamesdown
9	847	Restormel
9	854	South Hams
9	061	Stroud
9	162	West Wiltshire
9	158	Kennet
9	965	Weymouth and Portland
9	958	West Devon

-----  
 ---

**FAMILY RESOURCES SURVEY****DERIVED VARIABLE SPECIFICATION**

-----  
--

9	754	Kerrier
9	751	Woodspring
9	748	Kingswood
9	752	Caradon
9	656	Bath
9	961	North Dorset
9	852	North Devon
9	755	North Cornwall
9	964	West Dorset
9	851	Mid Devon
9	066	West Somerset
9	063	Mendip
9	747	Bristol

---

10	179	Brecknock
10	876	Wrexham Maelor
10	082	Ynys Mon - Isle of Anglesey
10	872	Colwyn
10	977	Carmarthen
10	182	Cardiff
10	871	Alyn and Deeside
10	178	Taff-Ely
10	277	Swansea
10	183	Vale of Glamorgan
10	077	Torfaen
10	978	Ceredigion
10	078	Aberconwy
10	982	South Pembrokeshire
10	080	Dwyfor
10	081	Meirionnydd
10	984	Islwyn
10	981	Preseli
10	181	Radnorshire
10	084	Merthyr Tydfil
10	177	Rhymney Valley
10	875	Rhuddlan
10	086	Rhondda
10	186	Neath
10	184	Port Talbot (aka Afan)
10	979	Dinefwr
10	085	Ogwr
10	983	Blaenau Gwent
10	874	Glyndwr

-----  
---

**FAMILY RESOURCES SURVEY****DERIVED VARIABLE SPECIFICATION**

-----  
--

10	079	Arfon
10	985	Monmouth
10	180	Montgomeryshire
10	873	Delyn
10	986	Newport
10	980	Llanelli
10	083	Cynon Valley
10	185	Lliw Valley

---

11	481	Nairn
11	685	Motherwell
11	386	Moray
11	381	North East Fife
11	286	Nithsdale
11	679	Hamilton
11	378	Wigton
11	278	Berwickshire
11	480	Lochaber
11	385	Kincardine and Deeside
11	380	Kirkcaldy
11	580	Bearsden & Milngavie
11	682	Kyle and Carrick
11	683	Lanark (aka Clydesdale)
11	681	Kilmarnock and Loudoun
11	479	Inverness
11	684	Monklands
11	783	Western Isles Islands Area
11	577	Midlothian
11	382	Aberdeen City
11	478	Caithness
11	680	Inverclyde
11	578	West Lothian
11	777	Strathkelvin
11	779	Dundee City
11	585	Dumbarton
11	778	Angus
11	379	Dunfermline
11	581	Clydebank / Clydesdale (see Lanark)
11	782	Shetland Islands Area
11	483	Skye and Lochalsh
11	579	Argyll and Bute
11	584	Cunninghame
11	284	Stirling
11	285	Annandale and Eskdale
11	484	Sutherland

-----  
---

-----  
 --

11	377	Stewartry
11	583	Cumnock and Doon Valley
11	582	Cumbernauld and Kilsyth
11	282	Clackmannan
11	485	East Lothian
11	686	Renfrew
11	279	Ettrick and Lauderdale
11	283	Falkirk
11	780	Perth and Kinross
11	678	Glasgow City
11	781	Orkney Islands Area
11	482	Ross and Cromarty
11	281	Tweeddale
11	477	Badenoch and Strathspey
11	586	East Kilbride
11	383	Banff and Buchan
11	677	Eastwood
11	486	Edinburgh City
11	280	Roxburgh
11	384	Gordon

-----  
 ---

-----  
 --

**SUBLTAMT**

Purpose : To show the amount of rent received by a benefit unit from sub-letting.  
 Created : VC - 11 January 1993  
 Database Table : BENUNIT  
 Minimum Value : 0  
 Maximum Value:  
 Units : Real  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : ISM **HBM** (**Bold** indicates lead user)  
 Amendments : VC - 27 April 1993. To expand the definition to make meaning of each :  
                     question/database variable clear.  
                     : VC - 14 September 1993 To make by benefit unit not household  
                     : VC - 17 February 1994 Amended to reflect version 30 changes  
                     : VE - 4 June 1996 - No initial amendments needed for V32  
 Issue date : 21 May 2003

NB - This variable has changed slightly from the FES variable as FES uses the amount of "profit" from sub-letting, whereas FRS uses the actual rent charged.

**1 Definition**

This variable is coded as

SUBLTAMT The amount of rent received by a benefit unit from sub-letting part of the property to a person who is not a member of the household or a boarder/lodger.

-1 Not applicable in this case.

-2 Unable to derive variable.

The amount of rent received from sub-letting is to be found in the HOUSEHOL table in the variable SUBLET. If SUBLET = 1 (indicating that there is a sub-letting arrangement in the household) the amount of rent charged is to be found in SUBRENT.

As this variable is collected on a household basis, the amount received in SUBRENT has to attributed to a benefit unit. In FES this was attached to the record of benefit unit number 1 (head of household BU) and this will be duplicated by FRS. Any other benefit unit will be designated not applicable.

**2 FRS Specification**

For each BENUNIT record, set SUBLTAMT to zero.

-----  
 ---

-----  
--

CodeCondition

SUBLTAMT      If BENUNIT = 1, process HOUSEHOL record and  
                  If SUBLET = 1 get the amount of rent charged from SUBRENT.

-1                Not applicable to this case where BENUNIT > 1 or SUBLET = 2

-2                Unable to derive in this case - where any of the above variables are missing.

**3        Results**

Tabulation is required to show the number of benefit units which have income from sub-tenants by the amount of rent received each week sorted into bands of, for example,

Under £25  
£25 - £50  
£50 - £75  
£75 - £100  
£100 - £125  
£125 - £150  
£150 - £175  
£175 - £200  
Over £200

-----  
---

-----  
 --

**SUPERAN**

Purpose : To indicate the total amount of superannuation or pension contributions :  
 deducted from a person's earnings from all jobs.  
 Created : VC  
 Database Table : ADULT  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables : Deducts  
     Children :  
     Parents :  
 Core variable/user : ISM HBM FCM PSM  
 Issue date : 21 May 2003  
 Amendments : VC - 9 February 1993 change to multi response.  
               : VC - 11 May 1993 amended to show superannuation or pension payments :  
               made from all income from jobs.  
               : VC - 23 August 1993 amended to emphasise that for employees only - also :  
               to indicate which groups are not applicable  
               : VC - 17 February 1994 Amended to reflect version 30 changes - it has an :  
               extra category detailing the contribution to an additional voluntary pension :  
               which has now been included.  
               : VC - 1 March 1994 To exclude any period codes 12 or 13  
               : JS - 21 February 1996 - to allow skipped values of PAYPD where PAYAMT  
               has been imputed  
               : VE - 23 May 1996 - No amendments needed for initial V32 update

NB - will not include self-employed jobs separate base variables provided for these jobs

**1 Definition**

This variable is coded as

- SUPERAN The total amount of superannuation or pension contributions deducted from a person's earnings from all jobs also includes contributions to additional voluntary pension.
- 1 Not applicable in this case - people who do not have superannuation deductions and those not working or self-employed
- 2 Unable to derive variable.

The amount of superannuation or pension or additional voluntary pension contributions is derived from all jobs and where OTHDED1 or OTHDED2 are coded 1 to show that an amount for a pension or superannuation is deducted. The variable DEDUC1 will then hold the amount of superannuation/pension contribution and DEDUC2 holds the amount of additional voluntary contributions. The variables OTHDED1, OTHDED2, DEDUC1 and DEDUC2 are created in the database to hold the answers to the

-----  
 ---



-----  
 --

multi response questions OTHDED (were there any other deductions from your wage/salary such as 1 = pension or superannuation, 3 = Union fees etc) and DEDUC which holds the amounts.

However, if the period code for the deduction is 12 or 13 (lumpsum/one-off or other period), from the PAYPD variable, the record must be set to unable to derive as it has not been possible to convert the amount of benefit into a weekly amount during the database conversion process. Therefore, if PAYPD = 12 or 13 SUPERAN is set to -2.

## 2 FRS Specification

For each ADULT with record Job for all jobs

<u>Code</u>	<u>Condition</u>
SUPERAN	If PAYPD equals -1 or 1-11 and -  If OTHDED1 = 1 and DEDUC1 exists, get the amount of the superannuation or pension contribution deducted from variable DEDUC1.  If OTHDED2 = 1, and DEDUC2 exists get the amount of the additional voluntary contribution deducted from variable DEDUC2.
-1	Not applicable in this case - where OTHDED1 = 2 or 3, OTHDED2 = 2 or 3 or OTHDED1/OTHDED2 not exist or there are no job records
-2	If variable cannot be defined because of missing data - where there is a job record but no values or if PAYPD = 12 OR 13.

## 3 Results

Tabulation is required to show the weekly amount of superannuation or pension contribution deducted from earnings sorted into bands, for example,

Under £5.00  
 £5.00 - £10.00  
 £10.00 - £15.00  
 £15.00 - £20.00  
 £20.00 - £25.00  
 Over £25.00

-----  
 ---

-----  
 --

**TENTYPE**

Purpose : To indicate the number of households in any specific tenure type.  
 Created : 14 December 1992  
 Database Table : HOUSEHOL  
 Minimum Value : 1  
 Maximum Value : 7  
 Units : Integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : **ASD6A**  
 Issue date : 21 May 2003  
 Amendments : VC - 28 January 1993  
               : VC - 15 March 1993 Amendments to insert the part-own part-renter category  
               : into the appropriate renter category, to put tenure = 2 or 3 into the owner :  
               : categories and to exclude tenure = 5 as these tenants are not eligible for :  
               : Housing Benefit.  
               : VC - 18 May 1993. Amended to re-insert tenure = 5 as a separate :  
               : category and also to add another category for co-ownership schemes : as both  
               : of these groups are not eligible for HB.  
               : VC - 9 June 1993. Amended to insert extra category for shared ownership.  
               : BS - 3 August 1995. Amended to include changes to V31 of the  
               : questionnaire. New var FURNISH asked if TENURE equals new codes 6, 7 or  
               : 8.  
               : JS - to clarify tentype=3 category and to allow skipped values of LANDLORD,  
               : FURNISH and OWNHOW (in cases where LANDLORD or TENURE have been  
               : imputed)  
               : JS - to note categories are actually slightly different from the name suggested!  
               : VE - 22 May 1996 - No initial Version 32 update needed  
               : VE - 10 June 1996 - To bring specification in line with Harmonisation of Survey  
 Outputs: part owner/part renter cases are now included with cases buying with  
 a mortgage and lettings which go with the job of someone in the household are  
 now classified as rented privately.

**1 Definition**

This variable is coded as

- 1 LA rented
- 2 Housing Association
- 3 Other private rented unfurnished
- 4 Other private rented furnished
- 5 Owned with mortgage (**Includes part rent/part own**)
- 6 Owned outright (**Includes part rent/part own**)
- 7 Rent free
- 8 Other Crown Estates/Government Departments.

-----  
 ---

**FAMILY RESOURCES SURVEY**

**DERIVED VARIABLE SPECIFICATION**

-----  
 --

- 9 Co-ownership schemes
- 10 Shared ownership
- 1 Not applicable to this case
- 2 Unable to derive variable

Tenure type is derived from the variable tenure in the HOUSEHOL record which when combined with a second variable from either the owner or the renter records will indicate to which category of tenure that particular household belongs.

FURNISH is only asked where landlord=6-8, coding for TENTYPE=3 needs to reflect this

IT SHOULD BE NOTED THAT THIS SPEC IS ONLY A BASIC REWORKING OF *TENTYPE* (V30). MORE DETAILED INFORMATION CAN BE OBTAINED IF *TENURE* = 6, 7 OR 8. THEN *FURNISH* IS ASKED TOGETHER WITH *SHORT* AND *OTHWAY* (*RENTER* TABLE). THESE QUESTIONS OBTAIN DATA RELATING TO HOW THE PROPERTY IS LET. AT PRESENT (3 AUGUST 1995) *TENTYPE* DOES NOT USE THIS INFORMATION.

NB/

<b>TENURE</b>	1) Owns/is buying	<b>LANDLORD</b>	1) Council
	2) Co-ownership scheme		2) New Town Corporation
	3) Shared ownership		3) Housing Association
	4) Part own/Part rent		4) Crown Estates Commissioners
	5) Rented		5) Other Crown/Government Depts
	6) Rent free		6) Friend or relative
			7) Other organisation
			8) Other individual
<b>FURNISH</b>	1) Furnished	<b>SHORT</b>	1) An assured shorthold
	2) Unfurnished		2) A shorthold, but not assured
			3) Or it does not say it is a shorthold at all.
			4) A shorthold, but not sure if assured or not.

-----  
 ---

-----  
 --

<b>OTHWAY</b>	1) Company licence		
	2) College licence		
	3) Non-exclusive occupancy agreement		
	4) Holiday let		
	5) Low season let		

**2 FRS Specification**

For each household, process HOUSEHOL record

Code   Condition

Where TENURE = 4 ~~or~~ 5, process RENTER record for details of type of landlord

- 1      If TENURE = 4 ~~or~~ 5 and LANDLORD = 1 or 2 (council/New Town Corporation) **and ACCJOB=2**
- 2      If TENURE = 4 ~~or~~ 5 and LANDLORD = 3 (Housing Association) **and ACCJOB=2**
- 3      If TENURE = 4 ~~or~~ 5 and ((LANDLORD = 4) or (LANDLORD= 6, 7 or 8 and FURNISH = 2)) **or (ACCJOB=1 and FURNISH=2)** (Crown Estates/other unfurnished)
- 4      If TENURE = 4 ~~or~~ 5 and ((LANDLORD = 6, 7 or 8 and (FURNISH = 1 or FURNISH= -1)) or LANDLORD=-1) **or (ACCJOB=1 and FURNISH=1)** (other furnished)
- 8      If TENURE = 4 ~~or~~ 5 and LANDLORD = 5 (other Crown Estates/Govt Dept)

Where TENURE = 1 **or** 4, process OWNER record for details of how the property is owned

- 5      If TENURE = 1 **or** 4 and OWNHOW = 2 (owns/bought with mortgage or loan)
- 6      If TENURE = 1 and OWNHOW = 1 or OWNHOW= -1 (owns/outright) *(this closes down the route)*

Where TENURE = 6

- 7      If TENURE = 6 (rent free).

Where TENURE = 2

-----  
 ---

-----  
--

9 If TENURE = 2 (co-ownership)

Where TENURE = 3

10 If TENURE = 3 (shared ownership)

-1 Not applicable to this case

-2 Unable to derive - if any of the above variables are missing.

**3 Results**

Tabulation is required to show the numbers of households which fall into each category.

**4 Test Cases**

A Single person living in Local Authority housing.

B Single person living in Housing Association accommodation.

C Single person living in an unfurnished flat rented from a private landlord.

D Single person living in a furnished flat rented from a private landlord.

E Single person living in his/her own home with a mortgage outstanding on the property.

F Single person living in his/her own home with no outstanding mortgage.

G Single person living rent free.

Repeat all of the above for couples.

-----  
---

-----  
 --

**TOTCAPBU (amended TOTCAP)**

Purpose : To show the total amount of capital a benefit unit (head plus spouse) possesses  
 Created : 15 January 1993  
 Database Table : BENUNIT  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables : TOTCAPCH - total amount of child's capital  
     Children :  
     Parents :  
 Lead User : ASD3A  
 Amendments: : updated to V31: totsav now held on BU record  
               : VE - 5 June 1996 - Initial amendments for V32 - Removed SAMPLQTR  
               information because no longer needed  
               : VE - 4 July 1996 - Amended to refer to constants table  
               : VE - 3 December 1996 - Amended to clarify the situation where TOTSAVE is  
               missing and to use HOWMUCHE for assettype 8 where it exists  
 Issue date: : 21 May, 2003

**1 Definition**

This variable is coded as

TOTCAPBU The total amount of capital adults in a benefit unit possess from all sources.

-1 Not applicable to this case.

-2 Unable to derive due to missing values.

The total amount of capital a person has is calculated using two methods. Firstly, for those cases which have an assets record from several variables which have information about the number and type of different assets/accounts. The variable assettype indicates whether or not a person has a particular asset so if this variable is coded 1,2,4,6,8,9,11 or 13 there should be an additional variable which will hold the amount. If a person has more than one asset, an individual's capital will be the total of all of these variables. TOTCAPBU is derived by adding together amounts for the head and, where appropriate, the spouse.

Note: during the FRS interview, for asset types 4, 6, 8, and 13 (gilts, stocks and shares, national savings certificates, SAYE, national savings bonds) respondents are asked for an estimate of their holdings (coded at HOWMUCH). During the office edit, actual values are looked up and stored in HOWMUCHE. Coding therefore is amended to look at HOWMUCHE if it exists and otherwise, HOWMUCH. (This is reflected in the flat file which holds HOWMUCHE in preference to HOWMUCH where appropriate.) However, for national savings certificates, issue value is used in place of HOWMUCH so coding is unaffected.

-----  
 ---

-----  
 --

The asset questions in FRS are only asked of individuals within benefit units who have declared that the total value (all adults combined) have capital between £1500 and £20000. For benefit units which have answered that they have assets of less than £1500 or over £20000 no further details about assets are collected. This is also the case for benefit units who have refused to answer any more questions about their assets. For these BUs, therefore, the total amount of capital it possesses needs to be calculated in another way.

This second method uses the amount of interest received on a person's capital investments to calculate an approximate amount of capital per individual which could have generated that amount of interest. The interest is divided by an appropriate (weekly) interest rate and then multiplied by 100 to get a capital amount. This figure then needs to be multiplied by 52 to produce an annual figure. The rates used have been produced by Mr Ray and follow the traditional FES approach to calculating capital. This method gives a capital amount for people who have not answered any further questions about their assets. Again, these are totalled for head and spouse to calculate TOTCAPBU. **The constants r\*\*\* are held in the table V32Constants in the spreadsheet h:\frsldvars\docs\spreads.xls.**

However, there is a mismatch between the interest/dividend payments and the amounts collected in the assets block. Whilst amounts for National Savings Certificates, SAYE, premium bonds and National Savings Bonds are included, respondents are only asked whether they have these type of savings at question otinva. Therefore, there are no interest/dividend amounts on which to calculate holdings. In these cases totcap should be increased by 20%.

The coding of Totsav was changed at the beginning of July 1994 (SAMPLQTR=2). The four codes used in 1993/94 were split into 5 categories:

was	from	
	July <del>95</del> <b>94</b>	
1	1	less than £1,500
2	2	£1,500 to £8,000
2	3	£8,000 to £20,000
3	4	£20,000 and over
4	5	does not wish to say

The routing remains the same, but for the final quarter's data the program has to be altered to correctly identify the method to calculate TOTCAPBU

**2 FRS Specification**

<u>Code</u>	<u>Condition</u>
TOTCAPBU	From BENUNIT record, if TOTSAV = 2 or ( <del>TOTSAV=3 and SAMPLQTR=2,3 or 4</del> )  From ASSETS record, for each asset held for each adult  If assetype = 1, get amount of capital in current accounts from HOWMUCH. If assetype = 2, get amount of capital in savings accounts from HOWMUCH. If assetype = 4, get HOWMUCHE if it exists else use HOWMUCH (gilts/trusts).

-----  
 ---

-----  
 --

If assetype = 6, get **HOWMUCHE** if it exists else use the issue value of National Savings Certs from ISSVAL.

If assetype = 8, get HOWMUCHE if it exists else use HOWMUCH (SAYE)

If assetype = 9, get amount of capital in premium bonds from HOWMUCH.

If assetype = 11, get amount of capital in Nat Sav Income Bonds from HOWMUCH.

If assetype = 13, get HOWMUCHE if it exists else use HOWMUCH (Nat Sav Capital Bonds).

TOTCAPBU is then the total of any occurrences of the above

Else if from BENUNIT record, ~~(TOTSAV=1,3,4 or missing and SAMPLQTR=1) or (TOTSAV=1,4 or 5 or missing and SAMPLQTR=2,3 or 4)~~

Set the interest rates for each type of account

Then calculate amounts as follows -

From ACCOUNTS record

If ACCOUNTS = 1 and TOTSAV = 1 calculate CAP = ACCINT/r01a%

**Else if ACCOUNTS = 1 and TOTSAV = 4 calculate CAP = ACCINT/r01b%**

**Else if ACCOUNTS = 1 and TOTSAV = 5 or missing calculate CAP = ACCINT/r01c%**

If ACCOUNTS = 2 and TOTSAV = 1 calculate CAP = ACCINT/r02a%

**Else if ACCOUNTS = 2 and TOTSAV = 4 calculate CAP = ACCINT/r02b%**

**Else if ACCOUNTS = 2 and TOTSAV = 5 or missing calculate CAP = ACCINT/r02c%**

If ACCOUNTS = 3 and TOTSAV = 1 calculate CAP = ACCINT/r03a%

**Else if ACCOUNTS = 3 and TOTSAV = 4 calculate CAP = ACCINT/r03b%**

**Else if ACCOUNTS = 3 and TOTSAV = 5 or missing calculate CAP = ACCINT/r03c%**

If ACCOUNTS = 4 and TOTSAV = 1 calculate CAP = ACCINT/r04a%

**Else if ACCOUNTS = 4 and TOTSAV = 4 calculate CAP = ACCINT/r04b%**

**Else if ACCOUNTS = 4 and TOTSAV = 5 or missing calculate CAP = ACCINT/r04c%**

If ACCOUNTS = 5 and TOTSAV = 1 calculate CAP = ACCINT/r05a%

**Else if ACCOUNTS = 5 and TOTSAV = 4 calculate CAP = ACCINT/r05b%**

**Else if ACCOUNTS = 5 and TOTSAV = 5 or missing calculate CAP = ACCINT/r05c%**

If ACCOUNTS = 6 and TOTSAV = 1 calculate CAP = ACCINT/r06a%

**Else if ACCOUNTS = 6 and TOTSAV = 4 calculate CAP = ACCINT/r06b%**

**Else if ACCOUNTS = 6 and TOTSAV = 5 or missing calculate CAP = ACCINT/r06c%**

-----  
 ---



-----  
--

If ACCOUNTS = 7 and TOTSAV = 1 calculate CAP = ACCINT/r07a%  
Else if ACCOUNTS = 7 and TOTSAV = 4 calculate CAP = ACCINT/r07b%  
Else if ACCOUNTS = 7 and TOTSAV = 5 or missing calculate CAP =  
ACCINT/r07c%

If ACCOUNTS = 8 and TOTSAV = 1 calculate CAP = ACCINT/r08a%  
Else if ACCOUNTS = 8 and TOTSAV = 4 calculate CAP = ACCINT/r08b%  
Else if ACCOUNTS = 8 and TOTSAV = 5 or missing calculate CAP =  
ACCINT/r08c%

If ACCOUNTS = 9 and TOTSAV = 1 calculate CAP = ACCINT/r09a%  
Else if ACCOUNTS = 9 and TOTSAV = 4 calculate CAP = ACCINT/r09b%  
Else if ACCOUNTS = 9 and TOTSAV = 5 or missing calculate CAP =  
ACCINT/r09c%

If ACCOUNTS = 10 and TOTSAV = 1 calculate CAP = ACCINT/r10a%  
Else if ACCOUNTS = 10 and TOTSAV = 4 calculate CAP = ACCINT/r10b%  
Else if ACCOUNTS = 10 and TOTSAV = 5 or missing calculate CAP =  
ACCINT/r10c%

If ACCOUNTS = 11 and TOTSAV = 1 calculate CAP = ACCINT/r11a%  
Else if ACCOUNTS = 11 and TOTSAV = 4 calculate CAP = ACCINT/r11b%  
Else if ACCOUNTS = 11 and TOTSAV = 5 or missing calculate CAP =  
ACCINT/r11c%

If ACCOUNTS = 12 and TOTSAV = 1 calculate CAP = ACCINT/r12a%  
Else if ACCOUNTS = 12 and TOTSAV = 4 calculate CAP = ACCINT/r12b%  
Else if ACCOUNTS = 12 and TOTSAV = 5 or missing calculate CAP =  
ACCINT/r12c%

If ACCOUNTS = 13 and TOTSAV = 1 calculate CAP = ACCINT/r13a%  
Else if ACCOUNTS = 13 and TOTSAV = 4 calculate CAP = ACCINT/r13b%  
Else if ACCOUNTS = 13 and TOTSAV = 5 or missing calculate CAP =  
ACCINT/r13c%

If ACCOUNTS = 14 and TOTSAV = 1 calculate CAP = ACCINT/r14a%  
Else if ACCOUNTS = 14 and TOTSAV = 4 calculate CAP = ACCINT/r14b%  
Else if ACCOUNTS = 14 and TOTSAV = 5 or missing calculate CAP =  
ACCINT/r14c%

If ACCOUNTS = 15 and TOTSAV = 1 calculate CAP = ACCINT/r15a%  
Else if ACCOUNTS = 15 and TOTSAV = 4 calculate CAP = ACCINT/r15b%  
Else if ACCOUNTS = 15 and TOTSAV = 5 or missing calculate CAP =  
ACCINT/r15c%

Note: r\* is for demonstration only; actual values included in program may be specified differently

-----  
---

-----  
--

TOTCAPBU is then the total of each occurrence of CAP

If ACCOUNT=16,17,18,20,21,22,23 then TOTCAP=TOTCAP\*1.20

Then TOTCAPBU=TOTCAPBU\*52

-2            Unable to derive because of missing values.

-----  
---



-----  
 --

child's assets is collected. In these cases, therefore, the total amount of capital must be calculated in another way.

The amount of interest received on a child's capital investments is used to calculate an approximate amount of capital which could have generated that amount of interest. The interest is divided by an appropriate interest rate percentage to get a capital amount. The rates used have been produced by Mr Ray and follow the traditional FES approach to calculating capital (see ~~attached minute~~ **V32Constants held in h:\frs\dvars\docs\spreads.xls**). This method also gives a capital amount for people who have refused to give any further details about their assets which also follows the FES.

However, there must also be a check built into the programme so that if this calculation produces an amount which does not agree with the original answer to TOTSAVE, it is assumed that TOTSAVE is more accurate and a standard amount is used instead. Therefore, if the amount is over £1500 and TOTSAVE = 1, TOTCAP is set to £1499 and if the amount calculated is under £20000 and TOTSAVE = ~~3~~ **4**, TOTCAP is set to £20001.

## 2 FRS Specification

For each CHILD

### Code

### Condition

TOTCAPCH From CHILD table, if TOTSAVE = 2 **or** 3

From ASSETS record, for each asset held for each child

If assetype = 3, get amount of capital in NSB ordinary, NSB investment, Building Society, bank accounts from HOWMUCH.

If assetype = 5, get amount of capital in Gilts, Unit Trusts, Stocks and Shares) from HOWMUCHE **if exists, else use HOWMUCH.**

If assetype = 7, get **amount from HOWMUCHE if it exists else use** issue value of all Index-linked and Fixed National Savings Certificates from ISSVAL

If assetype = 10, get amount of capital in Premium Bonds from HOWMUCH.

If assetype = 12, get amount of capital in National Savings Income Bonds from HOWMUCH.

If assetype = 14, get amount of capital in National Savings Capital Bonds and Children's Bonus Bonds from HOWMUCHE **if exists, else use HOWMUCH.**

TOTCAPCH is then the sum of the above where TOTSAVE = 2 **or** 3.

If TOTSAVE = 1, ~~3 or 4~~ **4 or 5 or missing**

~~Set the following rates for 1992/93 (standard table?)~~

~~r1 = 3.5~~

~~r2 = 5.0 } Use r2 if ACCINT > £500, use r3 if <= £500.~~

~~r3 = 2.5 }~~

~~r4 = 7.0~~

-----  
 ---

-----  
 --

r5 = 4.0  
 r6 = sum of r1 to r5 and r7 to r8/7  
 r7 = 7.0  
 r8 = 4.5

If INTDATE (interview date) is on or after 1/1/93 use the following rates for 1992/93

----- r1 = 3.5  
 ----- r2 = 3.75 } Use r2 if ACCINT > £500, use r3 if <= £500.  
 ----- r3 = 3.75 }  
 ----- r4 = 7.0  
 ----- r5 = 4.0  
 ----- r6 = sum of r1 to r5 and r7 to r8/7  
 ----- r7 = 7.0  
 ----- r8 = 4.5

Then calculate amount as follows -

From ACCOUNTS record

If ACCOUNTS = 24 and TOTSAV = 1, calculate CAP = ACCINT/r2% or r3% r02a%

Else if ACCOUNTS = 24 and TOTSAV = 4, calculate CAP = ACCINT/r02b%

Else if ACCOUNTS = 24 and TOTSAV = 5 or missing, calculate CAP =  
 ACCINT/r02c%

If ACCOUNTS = 25 and TOTSAV = 1, calculate CAP = ACCINT/r4% r03a%

Else if ACCOUNTS = 25 and TOTSAV = 4, calculate CAP = ACCINT/r03b%

Else if ACCOUNTS = 25 and TOTSAV = 5 or missing, calculate CAP =  
 ACCINT/r03c%

If ACCOUNTS = 26 or ACCOUNTS = 27 and TOTSAV = 1, calculate CAP =  
 ACCINT/r5% r01a%

Else if ACCOUNTS = 26 or ACCOUNTS = 27 and TOTSAV = 4, calculate CAP =  
 ACCINT/r01b%

calculate Else if ACCOUNTS = 26 or ACCOUNTS = 27 and TOTSAV = 5 or missing,  
 CAP = ACCINT/r01c%

If ACCOUNTS = 27, calculate CAP = ACCINT/r5%

If ACCOUNTS = 28, calculate CAP = ACCINT/r7% r11a%

If ACCOUNTS = 29, calculate CAP = ACCINT/r8% r13a%

If ACCOUNTS = 30, calculate CAP = ACCINT/r8% r14a%

TOTCAP is then the total of each occurrence of CAP.

If TOTCAP >= 1500 and TOTSAVE = 1, reset TOTCAP to £1499

----- If TOTCAP <= 20000 and TOTSAVE = 3, reset TOTCAP to £20001.

-1 Not applicable to this case.

-----  
 ---

-----  
--

-2                    Unable to derive because of missing values.

**3        Results**

Tabulations to show the total number of adults by the amount of capital they possess sorted into bands, for example

- Under £500
- £500 - £1000
- £1000 - £2000
- £2000 - £4000
- £4000 - £6000
- £6000 - £8000
- £8000 - £10000
- £10000 - £12000
- £12000 - £14000
- £14000 - £16000
- £16000 - £18000
- £18000 - £20000
- £20000 and over

NB - Test cases removed for immediate future as test dataset still to be finalised.

-----  
---

-----  
 --

**TOTGNTCH**

Purpose : To show the total amount of educational maintenance grants or scholarships  
 : received **directly** by a child.  
 Created : 15 July 1993  
 Database Table : ~~ADULT~~ **CHILD**  
 Minimum Value : 0  
 Maximum Value :  
 Units : Integer  
 Validations :  
 Related Variables : TOTGRANT - total grant received by an adult  
 Children :  
 Parents :  
 Core variable/user : ASD6A  
 Amendments : AG 9 August 1993. The FRS database will now keep the grant amounts as  
 : annual figures.  
 : JS - 7 March 1996: amended to look at payments actually received by  
 : students, rather than the total awarded  
 : VE - 22 May 1996 - No initial amendments needed for V32  
 : JS - 24 March 1997 - add in where 3 grants recorded at GRTNUM  
 Issue date : 21 May 2003

NB - This will be provided as an annual amount and the individual models will have to calculate weekly entitlement for the duration of the academic year.

**1 Definition**

This variable is coded as

TOTGNTCH The total amount of income received by a child from educational grants, maintenance grants or scholarships.

-1 Not applicable to this case

-2 Unable to derive

TOTGNTCH will be derived from the variables GRTNUM, GRTSCE1, GRTSCE2, GRTDIR1, GRTDIR2. GRTNUM gives the number of grants/scholarships received and the amount of the grant then depends on whether the grant/scholarship is funded by the state or from a private or overseas source. This will be an annual amount as requested by ISM.

**2 FRS Specification**

For each CHILD where FTED = 1 (in full time education)

-----  
 ---

-----  
 --

<u>Code</u>	<u>Condition</u>
TOTGNTCH	<p>From CHILD table,</p> <p>If GRTNUM = 1 (Number of grants = one) and GRTSCE1 = 1 (source is state), get amount from GRDIR1 (amount of grant paid directly to student).            If GRTNUM = 1 (Number of grants = one) and GRTSCE1 = 2 or 3 (source is private or overseas), get amount from GRDIR1 (amount of grant paid directly to student).</p> <p>If GRTNUM = 2 <b>or</b> 3 (two or more grants) get the amount for first grant as above in addition to:-            If GRTSCE2 = 1 (source is state) get amount from GRDIR2.            If GRTSCE2 = 2 or 3 (source is private or overseas), get amount from GRDIR2.</p> <p>TOTGNTCH will then be the sum of the two grants.            This is equivalent to summing all occurrences of GRDIR1 and GRDIR2 for each child in full time education</p> <p>(NB - this does not include top-up loans as they are not payable to children)</p>
-1	Not applicable to this case
-2	Unable to derive as any of the above variables are missing.

**3 Results**

Tabulation is required to show the number of children receiving educational grants, maintenance grants or scholarships by the amount received sorted into bands of

- Under £500
- £5100 - £1000
- £1000 - £1500
- £1500 - £2000
- £2000 - £2500
- £2500 - £3000
- £3000 - £4000
- £4000 - £5000
- £5000 and over

**4 Test Cases**

None as yet

-----  
 ---



-----  
 --

**TOTGRANT**

Purpose : To show the total amount of educational maintenance grants or scholarships  
 : for higher education received **directly** by an adult.  
 Created : 22 January 1993  
 Database Table : ADULT  
 Minimum Value : 0  
 Maximum Value :  
 Units : Integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : ASD6A  
 Amendments : VC - 8 March 1993. To specify an annual amount as requested by ISM.  
               : VC - 22 March 1993. To amend the sort bands in the tabulation.  
               : AG - 9 August 1993. The database will store grant amounts as annual figures  
               : so no multiplication required.  
               : JS - 19 January 1996: amended to include all students in further education  
               : JS - 7 March 1996: amended to look at payments actually received by  
               : students, rather than the total awarded (whether including or excluding fees)  
               : VE - 22 May 1996 - No initial amendments needed for V32  
               : VE - 4 December 1996 - To set TOTGRANT to zero initially  
               : VE - 31 January 1997 - To remove the condition for TOTGRANT only to be  
               : derived for students in line with HBAI  
               : JS - 24 March 1997 - to add in where 3 or more grants recorded  
 Issue date : 21 May, 2003

NB - This will be provided as an annual amount and the individual models will have to calculate weekly entitlement for the duration of the academic year. Totgrant replaces FES variable IN411 as FRS cannot differentiate between a grant or a scholarship, therefore, the total amount of grants or scholarships are included.

**1 Definition**

This variable is coded as

TOTGRANT	The total amount of income received by an adult from educational grants, maintenance grants or scholarships.
-1	Not applicable to this case
-2	Unable to derive

TOTGRANT will be derived from the variables GRTNUM, GRTSCE1, GRTDIR1, GRTSCE2, and GRTDIR2. GRTNUM gives the number of grants/scholarships received and the amount of the grant then

-----  
 ---

-----  
 --

depends on whether the grant/scholarship is funded by the state or from a private or overseas source. This will be an annual amount as requested by ISM.

**2 FRS Specification**

For each adult ~~where FTED = 1 or TEA = 06 and TYPEED = 7 (university, polytechnic, further education)~~

Code            Condition

**Set TOTGRANT to ZERO**

TOTGRANT      From ADULT table,

If GRTNUM = 1 (Number of grants = one) and GRTSCE1 = 1 (source is state), get amount from GRTDIR1 (amount of grant).

If GRTNUM = 1 (Number of grants = one) and GRTSCE1 = 2 or 3 (source is private or overseas) get amount from GRTDIR1 (amount of grant) .

If GRTNUM = 2 **or** 3 (two or more grants) get the amount for first grant as above in addition to:-

If GRTSCE2 = 1 (source is state) get amount from GRTDIR2.

If GRTSCE2 = 2 or 3 (source is private or overseas) get amount from GRTDIR2

TOTGRANT will then be the sum of the two grants.  
 (NB - this does not include top-up loans)

-1                Not applicable to this case - adult not in full-time education.

-2                Unable to derive as any of the above variables are missing.

**3 Results**

Tabulation is required to show the number of adults receiving educational grants, maintenance grants or scholarships by the amount received sorted into bands of

- Under £500
- £5100 - £1000
- £1000 - £1500
- £1500 - £2000
- £2000 - £2500
- £2500 - £3000
- £3000 - £4000
- £4000 - £5000
- £5000 and over

-----  
 ---

-----  
--

**4 Test Cases**

None as yet

-----  
---

-----  
 --

**TOTHOOURS**

Purpose : To indicate the total number of hours a person works each week (main and : subsidiary)  
 Created : 10 February 1993  
 Database Table : ADULT  
 Minimum Value : 0  
 Maximum Value :  
 Units : Integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : ISM  
 Amendments : VC - 16 March 1993, to change the database table.  
               : VE - 22 May 1996 - No initial amendments needed for V32  
 Issue date : 6 June 1996

NB - This replaces SEFTPT as this was only produced for self-employed people - TOTHOOURS also includes employees .

**1 Definition**

This variable is coded as

TOTHOOURS The total number of hours worked by an adult from all jobs whether main or subsidiary, employed or self-employed including all regular overtime.

-1 Not applicable to this case

-2 Unable to derive due to missing values

TOTHOOURS is derived from the variables QHRS, QHRSSELF and EMPOVT. These are to be obtained for each job a person has and the total number of hours are then calculated. The variables include the number of hours as an employed earner, as a self-employed person and from any regular overtime.

**2 FRS Specification**

For each ADULT, set TOTHOOURS to zero

Code            Condition

TOTHOOURS For each adult access all JOB records

If EMPEE = 1 (employed person), get QHRS (number of hours worked by an employee) and EMPOVT (number of hours of overtime worked by an employee) and add into TOTHOOURS

-----  
 ---

-----  
--

If EMPEE = 2 (self-employed), get QHRSELF (number of hours worked by a self-employed person) and add into TOTHOURS

TOTHOURS will then be the sum of all occurrences of QHRS, EMPOVT and QHRSELF.

**3 Results**

Tabulation is required to show adults by the number of hours worked each week sorted into bands of for example

- Under 16
- 16 - 24
- 24 - 30
- Over 30

**4 Test Cases**

None as yet

-----  
---

## TOTSAVBU

Purpose : To create variable consistent with V30 totsavbu with 4 categories ~~for the whole of 1994/95~~

Created :

Database Table : BENUNIT

Minimum Value : 0

Maximum Value :

Units : Real

Validations :

Related Variables : TOTCAPBU - total amount of benefit unit's capital

Children :

Parents : TOTSAV

Core variable/user : Publication

Amendments : VE - 5 June 1996 - To tidy up spec for V32

Issue date : 5 June 1996

### 1 Definition

This variable is coded as

- |    |  |
|----|--|
| 1  | Less than £1,500   |
| 2  | Over £1,500 and up to £20,000  |
| 3  | Over £20,000   |
| 4  | Does not wish to say   |
| -1 | Not applicable to this case (where TOTSAV has been skipped: question only asked where respondents have already stated interest on savings)                                 |
| -2 | Unable to derive due to missing values (should only occur where "don't know" has been input, "refused" cases should be included as part of code 4 - does not wish to say). |

The question TOTSAV is asked of adults and gives respondents' estimate of their total savings held by partner/spouse. From July 1995 (~~SAMPLQTR=2~~), the coding was changed to include an additional category for £1,500 to £8,000 such that:

- |   |                               |
|---|-------------------------------|
| 1 | Less than £1,500              |
| 2 | Over £1,500 and up to £8,000  |
| 3 | Over £8,000 and up to £20,000 |
| 4 | Over £20,000                  |
| 5 | Does not wish to say          |

This split coding is held on the data base in TOTSAV. ~~To make available a single variable which can be used across the whole year, TOTSAVBU condenses the 5 codes for the second half of the year using variable SAMPLQTR (common variable for sample quarter).~~

## 2 FRS specification

From table BENUNIT

<u>Code</u>	<u>Condition</u>
1	TOTSAV=1
2	TOTSAV=2 or <del>(SAMPLQTR= 2,3 or 4 and TOTSAV=3)</del>
3	<del>(SAMPLQTR=1 and TOTSAV=3)</del> <del>or (SAMPLQTR= 2,3 or 4 and TOTSAV=4)</del>
4	<del>(SAMPLQTR=1 and TOTSAV=4)</del> or TOTSAV=5 or TOTSAV=-8
-1	TOTSAV=-1
-2	TOTSAV=-9

-----  
 --

**TTWCOSTS**

Purpose : To show weekly travel to work costs for each adult  
 Created :  
 Database Table : ADULT  
 Minimum Value : 0  
 Maximum Value :  
 Units : Real  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : ASD4A  
 Amendments : JS: for V31, to change calculation of weekly travel pass costs  
               : JS 18/12/95 to take on changes made to V30 (additional methods of  
               transport, coding of variable work costs cases)  
               : JS - 21 February 1996 to allow for skipped values where variables have been  
               imputed  
               : JS - 15 March 1996, to stop weeklyising of pass within program (already  
               weekly on the data base)  
               : JS - 17 April 1996 to amend cases where TTWCOST or TTWPAY has been  
               skipped to use calculated amount for car/bike (currently no change made)  
               : VE - 5 June 1996 - to include new category of TTWCODE for V32  
               : VE - 14 June 1996 - To amend for 1995-96 rates for cost per mile  
               : VE - 1 July 1996 - Amended for constants being held in a separate table  
               : VE - 29 October 1996 - to include TTWREC  
               : VE - 9 December 1996 - To amend spec for change in data structure  
               regarding TTWCODE1 and TTWCODE2  
               : VE - 1 April 1997 - To tidy up spec to bring in line with SAS code for V32  
 Issue date : 21 May 2003

**1 Definition**

This variable is coded as

TTWCOSTS	Total weekly travel to work costs paid by adult
0	Free travel
-1	Not applicable to this case - including adults with no usual place of work, or coded "other" form of transport (no information collected)
-2	Unable to derive due to missing values

Travel to work costs are based on the number of round trips per week (TTWFRQ). Adults may be coded as having up to 6 modes of transport and costs are collected on all types except walk/bicycle or "other"

-----  
 ---



-----  
 --

(TTWMOD=1 or 6). Public transport costs are calculated using the costs of bus passes or season tickets, recorded fare, contributions to drivers of shared cars, or on costs per mile of journeys in cars or on motorcycle. DSS rates for costs per mile have been used. **The rates (CARRATE and BIKERATE) are held on the table V32Constants (h:\frs\dvars\docs\spreads.xls)**

Questionnaire asks about total distance travelled to usual place of work. Where respondents use both car/motorcycle and train/bus/tube, original specification will double count costs. Therefore have to assume that most of the journey is by train/bus/tube and that the car/van/motorcycle is just to get individuals to the station. For want of anything better at the moment (although potential sources of better information are being pursued), assume average distance travelled by car/motorcycle is 2 miles.

Where PSSAMT has been imputed, PSSDATE1/2 will have been skipped. Amount held in PSSAMT is taken as weekly. Where TTWCOST has been imputed, TTWCOST is skipped: do not change TTWCOSTS in these cases

## 2 FRS Specification

TTWCOSTS From adult record, add up costs for each mode of transport TTWMOD 2 - 5

TTWMOD4=1 (bus/train/tube) or TTWMOD5=1 (works bus/company transport)

If TTWPSS=1 (Yes) and PSSAMT exists TTWCOSTS=PSSAMT  
 Else if PSSDATE1/2 is skipped then cost of pass equals PSSAMT

Else if respondent does not have a season ticket, calculate cost of round trip and multiply by number of trips each week

If TTWPSS=2 (No)  
     If ONEWAY=1 then TTWCOSTS=FARE\*2\*TTWFRQ  
     else If ONEWAY=2 then TTWCOSTS=FARE\*TTWFRQ

Else if travels in a car/van, first check whether car used in combination with bus/train/tube or works bus/company transport, if yes, assume average journey is 2 miles

If TTWMOD2=1 and (TTWMOD4=1 or TTWMOD5=1) and TTWPAY ne 3 (pays all/some of costs of taking car/van to work)  
     then TTWCOSTS=TTWCOSTS+(4\*TTWFRQ\*CARRATE)

Else if only car/van used (or used in combination with walking/bicycle) calculate total costs per mile per week (CARCOST - based on midpoints of TTWFAR categories),

If TTWMOD2=1 (car or van)  
     and TTWFAR=3 then CARCOST=0.5\*2\*TTWFRQ\*CARRATE  
     and TTWFAR=4 then CARCOST=2\*2\*TTWFRQ\*CARRATE  
     and TTWFAR=5 then CARCOST=4\*2\*TTWFRQ\*CARRATE  
     and TTWFAR=6 then CARCOST=7.5\*2\*TTWFRQ\*CARRATE

-----  
 ---

-----  
 --

and TTWFAR=7 then CARCOST=17.5\*2\*TTWFRQ\*CARRATE  
 and TTWFAR=8 then CARCOST=32.5\*2\*TTWFRQ\*CARRATE

~~1994/95 carrate=42p per mile~~ **1995-96 carrate=43p per mile**

Then calculate TTWCOSTS, taking into account any contributions

If TTWMOD2=1

If TTWPAY=1 (all) then TTWCOSTS=TTWCOSTS+CARCOST

~~else~~ If TTWPAY=2 (some) and TTWCODE1=1 then TTWCOSTS=TTWCOSTS+TTWCOST

~~else~~ If TTWPAY=2 and **TTWCODE2=1**

then TTWCOSTS=TTWCOSTS+(CARCOST- ~~TTWCOST~~ **TTWREC**)

(if CARCOST < ~~TTWCOST~~ **TTWREC** then set TTWCOSTS to zero)

else if ~~TTWCOST or~~ **TTWREC have been skipped (where TTWCODE1/TTWCODE2 are set)** or TTWPAY has been skipped **or if TTWCODE1=3 or and TTWCODE2=3** then TTWCOSTS=TTWCOSTS+CARCOST

Else if respondent drives a motorcycle, use similar approach, first checking whether motorcycle used in combination with bus/train/tube or works bus/company transport

If TTWMOD3=1 and (TTWMOD4=1 or TTWMOD5=1) **and TTWPAY ne 3 (pays all/some of costs of taking motorcycle to work)**

then TTWCOSTS=TTWCOSTS+(4\*TTWFRQ\*BIKERATE)

Else if only motorbike used (or in combination with walking/bicycle) calculating variable BIKECOST

If TTWMOD3=1 (motorcycle)

and TTWFAR=3 then BIKECOST=0.5\*2\*TTWFRQ\*BIKERATE

and TTWFAR=4 then BIKECOST=2\*2\*TTWFRQ\*BIKERATE

and TTWFAR=5 then BIKECOST=4\*2\*TTWFRQ\*BIKERATE

and TTWFAR=6 then BIKECOST=7.5\*2\*TTWFRQ\*BIKERATE

and TTWFAR=7 then BIKECOST=17.5\*2\*TTWFRQ\*BIKERATE

and TTWFAR=8 then BIKECOST=32.5\*2\*TTWFRQ\*BIKERATE

~~1994/95~~ **1995-96 bikerate=9p per mile**

Then calculate TTWCOSTS, taking into account any contributions

If TTWMOD3=1

If TTWPAY=1 (all) then TTWCOSTS=TTWCOSTS+BIKECOST

~~else~~ If TTWPAY=2 (some) and TTWCODE1=1 then TTWCOSTS=TTWCOSTS+TTWCOST

~~else~~ If TTWPAY=2 and **TTWCODE2=1**

then TTWCOSTS=TTWCOSTS+(BIKECOST- ~~TTWCOST~~ **TTWREC**)

(if BIKECOST < ~~TTWCOST~~ **TTWREC** then set TTWCOSTS to zero)

-----  
 ---

-----  
--

else if TTWCOST or **TTWREC** have been skipped (where TTWCODE1/TTWCODE2 are set) or TTWPAY has been skipped or if TTWCODE1=3 or TTWCODE2=3 then TTWCOSTS=TTWCOSTS+BIKECOST

- 0 TTWFAR=1 (work at home, live at work, no work journey)  
If TTWPSS=1 and PSSAMT=0 (has pass and costs nothing)  
or TTWPSS=2 and FARE=0 (does not have pass but fare costs nothing)  
or cash received from passengers etc greater than calculated costs (costs therefore set to zero)  
or ((TTWMOD2=1 or TTWMOD3=1) and TTWPAY=3) (uses car/motorcycle and pays no costs)

Note: Free travel cards/fares and contributions will have already been calculated: all categories shown here for completeness

Note some cases exist where individual drives to a station but has a free travel pass. These cases still count as zero travel costs.

- 1 If questions in n\_Travel have been skipped (questions do not apply to this case)  
or TTWFAR=2 (varies, no usual place of work)  
or TTWMOD6=1 (other form of transport: no information collected)
- 2 Any variables missing

-----  
---

-----  
 --

**UGRSPAY**

Purpose : To show the total amount of usual earnings received by an adult from each job as an employee, excluding any income from odd jobs..

Created :

Database Table : JOB

Minimum Value : 0

Maximum Value :

Units : Real

Validations :

Related Variables : GROSSPAY

Children :

Parents :

Core variable/user : ISM HBM PSM **FCM** (**Bold** indicates lead user)

Issue date : 21 May 2003

Amendments : JS - 21 February 1996 to allow for skipped values where variables have been imputed

: JS - 21 March 1996 to correct treatment of UGROSS so it is only considered if jobtype=1

: VE - 4 June 1996 - Initial amendments for V32 - New categories of DEDUC and OTHDED

**1 Definition**

This variable is coded as

UGRSPAY The total usual gross earnings before deductions for Income Tax, NI etc from each jobs an adult may have as an employed earner, excluding any income from odd jobs.

-1 Not applicable where an adult does not have any jobs.

-2 Unable to derive where any variables missing.

The variable UGRSPAY is derived from a variety of variables held in the ADULT and JOB tables and for each job held by that adult. *It is similar to GROSSPAY but looks at jobs individually and gives usual rather than last pay where appropriate.*

UGRSPAY is derived from the variable GRSWAGE which holds the person's gross earnings before tax, NI etc but only where the payslip has been consulted (where PAYSLIP = 1).

If the payslip has not been consulted (where PAYSLIP = 2) the amount of net pay is obtained from PAYAMT. This variable holds the total amount of net pay after all deductions have been taken off and these deductions must be added back to PAYAMT to find UGRSPAY.

The amount of income tax deducted is found in PAYE and National Insurance in NATINS these must be found in all cases. Other deductions for example trade union fees, payments to charities etc are also to be added back to PAYAMT but may not be relevant to every case. Therefore, if CHRTAXF = 1 (indicating that the person has a deduction for charities) (see deducts for reson using this not charity

-----  
 ---

-----  
 --

variable) the amount held in AMTTAXF must be added to PAYAMT and if CHROTH = 1 (indicates that there is another deduction for a charity) the amount held in AMTOTH must be added to PAYAMT.

If any of OTHDED1 to OTHDED67 = 1 there will be a deduction for pension/superannuation, union fees, friendly societies, sports social clubs, **repayment of a loan from employer** and any other deductions with the amount held in the relevant DEDUC variable. DEDUC1 holds the amount for pension/superannuation, DEDUC2 holds the amount additional voluntary contributions, DEDUC3 holds the amount of union fees, DEDUC4 holds amount for friendly societies, DEDUC5 holds amount for sports clubs, **DEDUC6 holds amount for repayment of a loan from employer** and DEDOTH holds the amount for any other deduction not already mentioned and any occurrence of these must be added to PAYAMT.

The variables OTHDED1 to OTHDED67, DEDUC1 to DEDUC56 and DEDOTH are database variables collected from the questions OTHDED in the e-main block which asks were there any other deductions from your wage or salary and DEDUC which holds the amount of deduction in each case.

To get a person's gross earnings a check must be made to see if an income tax refund was included in PAYAMT. Therefore, if TAXINC = 1 the amount held in TAXAMT has to be deducted from PAYAMT to get a true amount of gross earnings. However, if last pay was not usual, then UGRSPAY uses UGROSS.

**where payslip has been imputed to yes, GRSWAGE will have been skipped. Similarly where PAYAMT is imputed, PAYPD and other variables may also be skipped. This has to be catered for in the specification.**

## 2 FRS Specification

For each ADULT and each JOB set UGRSPAY to zero.

UGRSPAY From JOB table

**IF WORKING=1 or JOBAWAY=1 - process each JOB record for that person where EMPEE=1**

If PAYPD equals -1 or 1-11 and

If PAYSZIP = 1 and GRSWAGE exists, calculate UGRSPAY = amount in GRSWAGE.

Or if PAYSZIP = 2 or (PAYSZIP=1 and GRSWAGE=-1), calculate UGRSPAY as follows

UGRSPAY = PAYAMT and

If PAYE exists add PAYE to UGRSPAY

If NATINS exists add NATINS to UGRSPAY

If AMTTAXF exists add AMTTAXF to UGRSPAY

-----  
 ---

-----  
--

If AMTOTH exists add AMTOTH to UGRSPAY

If DEDUC1 exists add DEDUC1 to UGRSPAY

If DEDUC2 exists add DEDUC2 to UGRSPAY

If DEDUC3 exists add DEDUC3 to UGRSPAY

If DEDUC4 exists add DEDUC4 to UGRSPAY

If DEDUC5 exists add DEDUC5 to UGRSPAY

**If DEDUC6 exists add DEDUC6 to UGRSPAY**

If DEDOTH exists add DEDOTH to UGRSPAY

Then if JOBTYP=1 and TAXAMT exists subtract TAXAMT from total UGRSPAY

If JOBTYP=1 and PAYUSL=no and UGROSS exists and UPD=-1 or 1-11 then  
UGRSPAY=UGROSS

*(this should be the same as coding for INDINC. The only difference is that check on  
UGROSS at the beginning is not necessary because DV is on a job basis and  
therefore it is possible to overwrite UGRSPAY with UGROSS at the end)*

-2 If any of the above variables are missing or PAYPD = 12 or 13 *(should only be where  
PAYAMT is missing because for other variables, values are ignored if they do not exist)*

-----  
 --

**UPERSON**

Purpose : To show the person number within the Benefit Unit  
 Created : AJG 10 September 1993  
 Database Table : ADULT and CHILD  
 Minimum Value : 1  
 Maximum Value: 12  
 Units : Integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : HBAI  
 Amendments : VC - 17 February 1994 Amended to reflect version 30 changes  
               : VE - 22 May 1996 - No initial Version 32 update needed  
 Issued : 21 May 2003

**1 Definition**

This variable assigns the value 1 to the first person in each Benefit Unit increments by one for each adult and each child, within that Benefit Unit Benefit Unit. Dependents are number in descending order of age.

**2 FRS Specification**

Process each BENUNIT record in the household in turn, incrementing UPERSON as shown.

Code    Condition

For each BENUNIT record:

- 1        For the first ADULT in the Benefit Unit  
          Then in the following priority:
- +1        for the second ADULT in the Benefit Unit
- +1        for each dependent in descending order of age,

**3 Results**

What tabulation should be produced to check the results?

**4 Test Cases**

-----  
 ---

-----  
 --

**WATSEWRT**

Purpose :To show the total amount of water and sewerage rates paid by each household in England and Wales.

Created :2 February 1993

Database Table :HOUSEHOL

Minimum Value :0

Maximum Value :

Units :Real

Validations :

Related Variables :

    Children :

    Parents :

Issue Date :21 May 2003

Core variable/user :HBM

Amendments : VC - 4 May 1993 As per Linda Odwell's minute of 22 April 1993 to include sewerage for England and Wales but water only for Scotland.

: VC - 6 August 1993 Amended to include Scottish households as imputation will ensure that all spaces are filled.

: BS - 1 August 1995. Amended to take into account changes to V31.

: JS - 18 January 1996: amended to take into account routing of renter block: households with an HB statement are sent down a different route than those without one or not on HB

: JS - 21 February 1996: to allow skipped values where variables have been imputed and to make v30 amendments explicit

: JS - 8 March 1996: to simplify coding

: VE - 22 May 1996 - Amended to reflect initial V32 changes - WATERINC AND SEWERINC replaced by WORSINC. New question SEWSUPP.

: VE - 26 September 1996 - Final paragraph amended. WSINCAMT replaced by WATSEWRT (error in original form)

: VE - 10 March 1997 - Amended to bring into line with routing

: SG - 18 July 1997 - Amend check for whether water/sewerage included in rent

**1 Definition**

This variable is coded as

WATSEWRT The total weekly amount of water and sewerage rates paid by each household in England and Wales

-1 Not applicable to this case (Scottish Cases)

-2 Unable to derive due to missing values.

Amounts paid for water and sewerage are asked depending on whether paid separately or as part of rent. For renters, if they are on housing benefit and have a statement, amounts are held in HBWSAMT. If they are not on HB but pay charges as part of rent, the amount is held in WSINCAMT. For renters who pay charges separately or owner occupiers, amounts are held in WATAMT, SEWAMT and WSEWAMT.

-----  
 ---



-----  
 --

Previous versions of this program did checks on whether the household was connected to water and/or sewerage mains and calculated each component separately. Since the questions are only asked if this is the case, and there is no requirement to separate water and sewerage payments, these checks have been dropped.

The first step is to set WATSEWRT to zero. Since these questions contain missing values which have been imputed, some of the questions will have been skipped. Instead of setting the DV to -2, no change should be made.

For renters who pay water and sewerage as part of rent, if questions have been skipped WATSEWRT will be imputed using WATER. For households who did not know what they were connected to or whether they paid water and sewerage (WATERPAY or SEWERPAY missing) WATSEWRT will be imputed to zero by default. There are around 1000 zero cases on the 1994/95 data base.

**2 FRS Specification**

For each household, from HOUSEHOL table.

Code

Condition

WATSEWRT **Set WATSEWRT to zero.**

If STDREGN=11 then set WATSEWRT to skipped and exit record

**Cases where water and sewerage paid separately**

From HOUSEHOL record

**If (SEWSEP=1 or (WATERPAY=1 and SEWERPAY=2)) and # WATAMT exists**  
 WATSEWRT=WATSEWRT+WATAMT else don't change WATSEWRT

**If (SEWSEP=1 or (SEWERPAY=1 and WATERPAY=2)) and # SEWAMT exists**  
 WATSEWRT=WATSEWRT+SEWAMT else don't change WATSEWRT

**If SEWSEP=2 and # WSEWAMT exists** WATSEWRT=WATSEWRT+WSEWAMT else don't change WATSEWRT

**Cases where renting, water and sewerage paid as part of rent, and HB Statement**

From the RENTER record

If HBSTMT ~~exists~~ ~~and~~ =1 and HBWSAMT exists  
 WATSEWRT=WATSEWRT+HBWSAMT else don't change WATSEWRT

-----  
 ---

-----  
 --

**Cases where renting, water and sewerage paid as part or rent, and no HB statement (including those who didn't know if they had one) or not on HB**

From the RENTER record

**If WORSINC=1 or WSINC=1 or 2 or 3 (water or sewerage or both included in rent and no HB statement)**

If WSINCAMT exists  $WATSEWRT=WATSEWRT+WSINCAMT$   
 Else if ~~WATERINC exists and -1 or SEWERINC exists and -1~~ and WSINCAMT=-1  
 (values of waterinc or sewerinc or **WORSINC has** have been imputed but WSINCAMT  
 has been skipped) call WATER ~~WSINCAMT-WSINCAMT+WATER~~  
**WATSEWRT=WATSEWRT+WATER**  
 Else don't change WATSEWRT

-1 Not applicable to this case : Scotland

-2 Not able to derive as any of the above variables are missing.

-----  
 ---

-----  
 --

**YOUNGCH**

Purpose : To show the age of the youngest child in any benefit unit.  
 Created : 19 January 1993  
 Database Table : BENUNIT  
 Minimum Value : 0  
 Maximum Value :  
 Units : Integer  
 Validations :  
 Related Variables :  
     Children :  
     Parents :  
 Core variable/user : ~~ASD6A~~ **ASD4A**  
 Amendments : VC - 3 March 1993. To use MINR function to get age of youngest child.  
               : VC - 4 May 1993. To expand the definition to explain meaning of database :  
               variable used.  
               : VC - 17 February 1994 Amended to reflect version 30 changes  
               : VE - 21 May 1996 - No initial Version 32 update needed  
 Issue date : 21 May 2003

**1 Definition**

This variable is coded as

YOUNGCH The age of the youngest child in the benefit unit.

-1 Not applicable to this case

-2 Unable to derive because of missing values.

This variable is derived from a comparison of the ages of all children within a benefit unit.

**2 FRS Specification**

For each benefit unit, get variable DEPCHILDB (number of dependent child in BU) from BENUNIT table,

Code            Condition

YOUNGCH        From CHILD table,  
                   If DEPCHILDB > 0, use MINR function to obtain smallest value of age.

-1                Not applicable to this case - DEPCHILDB = 0

-2                Unable to derive due to age variable missing

**3 Results**

-----  
 ---

-----  
--

Tabulation to show number of youngest children

**4 Test Cases**

-----  
---