

ROUTING DOCUMENTATION: FRS BENEFIT UNIT QUESTIONNAIRE

VERSION 31: APRIL 1994

AreaNum;
AdrNum;
HHNum;
BUNum;

MaxRef;
CurrRef;
MaxDK;
CurrDK;

BLOCK: I_044_1;

CARGO1;
KeyQ;

IF (Transfev.OrigAd = 0) THEN
 NoHhold;

IF (BUNum > Transfev.NofBU) THEN
 WrongBU;

{Establishing who is going to be interviewed on this questionnaire}
AllocP1; {HIDDEN}
AllocP2; {HIDDEN}
Valloc1; {HIDDEN}
Valloc2; {HIDDEN}

PersDisp;

IntDate; {HIDDEN - computed from external paragraph}

NOTES ON ROUTING IN THE BENEFIT UNIT QUESTIONNAIRE

1. Much of the Benefit Unit questionnaire is arranged in 'tables', where a table is a line of questioning administered first to Person 1, then to Person 2. The start and end of tables are indicated in the documentation.
2. Sometimes a single question is asked first of Person 1 and then of Person 2. Normally, these are not contained within tables. Such questions are represented twice in the documentation, eg. '**Ben1Q[1]**' and '**Ben1Q[2]**', representing the question **Ben1Q** asked in turn of Persons 1 and 2.
3. Certain variables calculated from the data are used to define the routing. The most widely used ones are shown below with the relevant codes:

Belowpen	below pensionable age : 1 above pensionable age : 2
Belowp2	below (pensionable age - 1) : 1 above (pensionable age - 1) : 2
Work12m	working /worked within last 12 months : 1 worked, but not within last 12 months : 2
Work3m	working /worked within last 3 months : 1 worked, but not within last 3 months : 1
Jobstat1	employee, or last main job as employee : 1 self-emp, or last main job as self-emp : 1
Jobstat2	has/had second job as employee : 1 has/had second job as self-emp : 2
Jobstat3	has/had third job as employee : 1 has/had third job as self-emp : 2

Other variables are described at the start of the block(s) to which they apply.

BLOCK: Who1;

IF (session[1] = 0) THEN

Who1;

VWho1;

IF (Person2 <> 97) AND (session[2] = 0) THEN

Who2;

VWho2;

IF (session[1] = 0) OR (session[2] = 0) THEN

BLOCK: a_CURST;

{QCurst1;}

{start of table}

IF (session = 0) AND (belowpen = 1) THEN

Train;

IF (Train = Other) THEN

TecLec;

{end of table}

{QCurst2;}

{start of table}

IF (session = 0) THEN

Working;

IF (Working = No) THEN

JobAway;

IF (Working = Yes) OR (JobAway = Yes) THEN

NumJob;

Empstat;

IF (Working = No) AND (JobAway = No) THEN

Look4;

IF (Look4 = No) AND (Belowpe2 = 1)

AND (QCurst1.Adult.Train = None) THEN

{if not looking for work AND below pensionable age AND not on govt training scheme}

LkYt4;

IF LkYt4 = No THEN

Wait;

IF Wait = No THEN

LikeWk;

IF LikeWk = Yes THEN

NoLook;

IF LikeWk = No THEN

NoWant;

IF ((Look4 = Yes) OR (LkYt4 = Yes) OR

(LikeWk = Yes) AND
(belowpe2 = 1) THEN
Start;

IF (Start = No) THEN
YStart;

{If on a training scheme and not in additional paid work or}
{if seeking work/about to start a job already obtained}

IF (Look4 = Yes) OR
(Wait = Yes) OR
(JobAway = Waiting) THEN
LookWk;

IF (LookWk = Fulltime) OR (LookWk = Parttime) THEN
AccFtPt

{end of table}

{QCurst3;}
{start of table}

IF (session = 0) THEN

{If employee}

IF QCurst2.Adult.EmpStat = employee THEN
TDayWrk;

IF (TDayWrk = No) OR (TDayWrk = NotNorm) THEN
AbsWk;

IF AbsWk = Yes THEN
AbsWhy;

IF AbsWhy In [Illness..Other] THEN
AbsPay;
Abs1No;

IF Abs1No = 0 THEN
Abs2No {see also check paragraph}

{If retired}

IF (QCurst2.Adult.NoWant = Retired) OR
((belowpe2 = 2) AND
(QCurst2.Adult.Working = No) AND
(QCurst2.Adult.JobAway = No)) THEN
Retire;

IF Retire = Yes THEN

RetPay

{If on a training scheme and not in additional paid work or}
{anyone else not in paid work who is not retired}

IF (Retire <> Yes) AND
((QCurst2.Adult.Working = No) AND
(QCurst2.Adult.JobAway = No)) OR
((QCurst2.Adult.Working = No) AND
(QCurst2.Adult.JobAway = Waiting)) THEN
LstWrk2;

IF (LstWrk2 <> '0000') THEN
LstWrk1;

{If worked in the past 12 months}

IF ((VAL(SUBSTRING(IntDate,9,2)) * 12) +
(VAL(SUBSTRING(IntDate,4,2))) - 12)
< (((VAL(LstWrk2) - 1900) * 12) +
(ORD(LstWrk1))) THEN
LstYr;

{end of table}

{QPens1;}

{start of table}

IF (session = 0) THEN

{Ask of all except never work at LstWrk2}

IF (QCurst3.Adult.LstWrk2 <> '0000') AND
(TEA <> 0) AND (TEA <> 96) THEN
FtWk;
PtWk;

{end of table} {QClaim;}

{start of table}

IF (session = 0) THEN

{Ask of all men aged 16-64 and all women aged 16-59}

IF (Belowpe2 = 1) THEN
Claimant;

{end of table}

{QQuals1;}

{start of table}

IF (session = 0) THEN

anyed;

IF (anyed=yes) THEN

edhr;
edtime;

{end of table}

Jump1;

BLOCK: b_HEALTH;

```
{All below pensionable age}
IF (belowpe2[1] = 1) AND (session[1] = 0) THEN
  Rstrct[1];

IF (belowpe2[2] = 1) AND (session[2] = 0) THEN
  Rstrct[2];

{start of table}

IF session = 0 THEN
  IF (belowpe2 = 1) THEN

    IF QHealth1.Rstrct=Nowork THEN
      InjLong;

    IF QHealth1.Rstrct=Somework THEN
      Injwk;

    {If not in paid work}
    IF (a_Curst.QCurst2.Adult.Working = no) AND
      (a_Curst.QCurst2.Adult.JobAway = no) THEN
      Nolk;

      IF Caring in Nolk THEN
        Nlper;

  {end of table}          {start of table}

IF session = 0 THEN
  Health;

  IF Health=yes THEN
    Hprob;

  IF Hprob=yes THEN
    Lareg;
    Spcreg;

    IF belowpe2 = 1 THEN
      Jcreg;
  {end of table}
```


BLOCK: c_JOBDES;

CARGO2;

{Ask if person 1 in work OR if person 1 not in paid work,
but have worked in the last 12 months}
IF (a_CURST.QCurst2.Adult[1].Working = yes) OR
(a_CURST.QCurst2.Adult[1].JobAway = yes) OR
(work12m[1] = 1) THEN

Jump2;

IF (session = 0) THEN

{Subj1 - first job}

Title;
Respdo;
Qualif;
Firmdo;
Nature;

IF (Nature = CMind) THEN

Where;

Empee;

IF (Empee = SelfEmp) THEN

Dirctr;

IF (Subj1.Empee = Employee) OR
(Subj1.Dirctr = Yes) THEN

Manage;
NumEmp;

ELSEIF (Subj1.Empee = SelfEmp) AND
(Subj1.Dirctr = No) THEN

EmpAny;

{If not in work}

IF (a_curst.QCurst2.Adult.Working = no) AND
(a_curst.QCurst2.Adult.JobAway = no) THEN

NumJob2;

```
{Ask of all with a second job}
IF (a_curst.QCurst2.Adult.NumJob IN [Two..Four])
  OR (NumJob2 IN [Two..Four]) THEN
  I2;

  {Subj2 - second job}
  Title;
  Respdo;
  Qualif;
  Firmdo;
  Nature;

  IF (Nature = CMind) THEN
    Where;

  Empee;

  IF (Empee = SelfEmp) THEN
    Dirctr;
```

```
{Ask of all with a third job}
IF (a_curst.QCurst2.Adult.NumJob IN [Three..Four])
  OR (NumJob2 IN [Three..Four]) THEN
  I3;
```

```
{Subj3 - third job}
Title;
Respdo;
Qualif;
Firmdo;
Nature;
```

```
IF (Nature = CMind) THEN
  Where;
```

```
Empee;
```

```
IF (Empee = SelfEmp) THEN
  Dirctr;
```

```
{Ask of all with four or more jobs}
IF (a_curst.QCurst2.Adult.NumJob = Four) OR
  (NumJob2 = Four) THEN
  JobDisp;
```

```
{Ask if person 2 in work OR if person 2 not in paid work,
but have worked in the last 12 months}
IF (a_CURST.QCurst2.Adult[2].Working = yes) OR
  (a_CURST.QCurst2.Adult[2].JobAway = yes) OR
  (work12m[2] = 1) THEN
```

```
Jump3;
```

```
BLOCK: d_JOBDES; Same as c_JOBDES
```

PAY AND REMUNERATION:

In the Benefit Unit, the general rule of looped question ordering is:

```
    Person 1,  
      Loop 1 (questions 1 to N)  
      Loop 2 (questions 1 to N)... etc  
THEN  
    Person 2,  
      Loop 1 (questions 1 to N)  
      Loop 2 (questions 1 to N)... etc
```

In the pay/remuneration section, the order of loops is dependent on the employment status of the jobs recorded. Details of pay for ANY **Employee** job are asked about first, before details of profit etc. for ANY **self-employed** job:

```
    ANY employee job(s):  
      (if more than one person with employee job: Person 1, then  
Person 2)
```

```
THEN  
    ANY self-employed job(s):  
      (if more than one person with self-empl'd job: Person  
1, then Person 2)
```

EMPLOYEE PAY DETAILS:

The questions which follow are repeated, with appropriate internal routing, to form:

BLOCK: e_MAIN: Person 1 - First job as employee
BLOCK: f_SUB1: Person 1 - Second job as employee
BLOCK: g_SUB2: Person 1 - Third job as employee

BLOCK: h_MAIN: Person 2 - First job as employee
BLOCK: i_SUB1: Person 2 - Second job as employee
BLOCK: j_SUB2: Person 2 - Third job as employee

{Ask first person about first, second and third jobs as employee}

IF (jobstat1[pno]=1) THEN

Jump4;

BLOCK: e_MAIN; {adult 1, first job}

IF (session[pno] = 0) THEN

IF (a_curst.QCurst2.adult[pno].Working=yes) OR
(a_curst.QCurst2.adult[pno].JobAway=yes) OR
(work12m[pno]=1) THEN

CARGO3;

IF (a_curst.QCurst2.adult[pno].Working=yes) OR
(a_curst.QCurst2.adult[pno].JobAway=yes) THEN

PayDat;

PayAmt;

PayPd;

{If questions about main job}

IF Pmainjob = 1 THEN

TaxInc;

IF TaxInc = yes THEN

TaxAmt;

PAYE;

{If questions about main job and not paying PAYE}

IF (Pmainjob = 1) AND (PAYE = 0) THEN

TaxUsl;

NatIns;

IF (Pmainjob = 1) AND (NatIns = 0) THEN

NIPay;

Charity;

IF Charity = yes THEN

ChrTaxF;

IF ChrTaxF = yes THEN

AmtTaxF;

ChrOth;

IF ChrOth = yes THEN
AmtOth;

OthDed;

IF (PenDed IN OthDed) THEN
{PenDedq}
Deduc;

IF (AVCDed IN OthDed) THEN
{AVCDedq}
Deduc;

IF (UnDed IN OthDed) THEN
{UnDedq}
Deduc;

IF (FrdDed IN OthDed) THEN
{FrdDedq}
Deduc;

IF (SptDed IN OthDed) THEN
{SptDedq;}
Deduc;

IF (DedO IN OthDed) THEN
DedOth;
Payslip;

IF Payslip = yes THEN
GrWage;
GrSoFar;

MileInc;

IF MileInc = yes THEN
MileAmt;

MotInc;

IF MotInc = yes THEN
MotAmt;

```
HHInc;

IF HHInc = yes THEN
  {HH[1]}
  HHO;
  HHA;
  HHC;

  IF HH[1].HHC = yes THEN
    {HH[2]}
    HHO;
    HHA;
    HHC;

    IF HH[2].HHC = yes THEN
      {HH[3]}
      HHO;
      HHA;

IF ((a_curst.QCurst2.adult[pno].Working=yes) OR
(a_curst.QCurst2.adult[pno].JobAway=yes))
AND
((pmainjob = 1) AND (jobstat1[pno] = 1))
THEN

SSPSMP;

IF ((SSPSMP = 1) OR (SSPSMP = 2)) AND
(Payslip = Yes) THEN
SSPAmt;

IF ((SSPAmt = DONTKNOW) OR (SSPAmt =
REFUSAL)) OR
(((SSPSMP = 1) OR (SSPSMP = 2)) AND
(Payslip = No)) THEN
SSPRate;

IF (SSPSMP = 1) OR (SSPSMP = 2) THEN
MadEmp;

IF ((SSPSMP = 1) OR (SSPSMP = 3)) AND
(Payslip = Yes) THEN
SMPAmt;

IF ((SMPAmt = DONTKNOW) OR (SMPAmt =
REFUSAL)) OR
(((SSPSMP = 1) OR (SSPSMP = 3)) AND
(Payslip = No)) THEN
SMPRate;
```



```
IF ((SSPSMP = 1) OR (SSPSMP = 3)) THEN
    MatEmp;
    MatStp;

    PayUs1;

IF (PayUs1 = No) OR
    ((a_curst.QCurst2.adult[pno].Working=no) AND
    (a_curst.QCurst2.adult[pno].JobAway=no) AND
    (work12m[pno]=1) AND ((jobstat1[pno]=1) AND
    (pmainjob=1) OR (jobstat2[pno]=1) AND
    (pmainjob=2) OR (jobstat3[pno]=1) AND
    (pmainjob=3))) THEN
    UNett;
    UGross;
    UPd;

IF (PayUs1 = No) AND
    ((a_curst.QCurst2.adult[pno].Working=yes) OR
    (a_curst.QCurst2.adult[pno].JobAway=yes))
    AND
    ((pmainjob = 1) AND (jobstat1[pno] = 1))
    THEN
    U1Mot;

    IF U1Mot = Yes THEN
        U2Mot;
```

{HoursF}
as {The questions on hours are asked of main and subsidiary jobs
employee and main job as self-employed, but not about
subsidiary
job as self-employed}

IF (pemploye = 1) OR ((pemploye = 2) AND
(pmainjob = 1)) THEN

IF pemploye = 1 THEN
{EmpHr}
QHrs;

IF pmainjob = 1 THEN
EmpOvt;

ELSE
{SelfHr}
QHrsSelf;

IF (pmainjob = 1) THEN
IF (Ushours < 30) THEN
LikeHr;

IF (LikeHr = More) THEN
NoMor;

IF (Caring IN NoMor) THEN
NMPer

IF (Children IN NoMor) THEN
NMChc;

```
IF ((a_curst.QCurst2.adult[pno].Working=yes) OR
    (a_curst.QCurst2.adult[pno].JobAway=yes)) AND
    ((pmainjob = 1) AND (jobstat1[pno] = 1)) THEN
Bonus;

    {for each Bonus, max 6}
    FOR loopvar := 1 TO 6 DO
        IF (loopvar <= Bonus) AND (Bonus =
RESPONSE) THEN

            BonAmt [loopvar];
            BonTax [loopvar];

            {If received bonus and last pay was not usual}
            IF (Bonus > 0) AND (Bonus = RESPONSE) AND
                (PayUs1 = No) THEN
                UBonInc;

                IF UBonInc = yes THEN
                    UBonAmt;

            {If questions about main job}
            IF ((a_curst.QCurst2.adult[pno].Working=yes) OR
                (a_curst.QCurst2.adult[pno].JobAway=yes)) AND
                (pmainjob = 1) THEN
                TaxRel;

                IF TaxRel = yes THEN
                    AmtRel;

                LunchV;

                IF LunchV = yes THEN
                    LV7Dy;

                    IF LV7Dy = yes THEN
                        LVAmt;

                FreeM1;
                IF FreeM1 = yes THEN
                    FrM7Dy;

            IF ((a_curst.QCurst2.adult[pno].Working=yes) OR
                (a_curst.QCurst2.adult[pno].JobAway=yes))
                ((pmainjob = 1) AND (jobstat1[pno] = 1))
AND
THEN
                InKind;
```

```
IF (jobstat2[pno]=1) THEN
  Jump6;

  BLOCK: f_SUB1;      {Adult 1, second job}
                        {same as e_MAIN}
```

```
IF (jobstat3[pno]=1) THEN
  Jump8;

  BLOCK: g_SUB2;      {Adult 1, third job}
                        {same as e_MAIN}
```

{Ask second person about first, second and third jobs as
employee}

```
IF (jobstat1[pno]=1) THEN

  IF ((a_CURST.QCurst2.Adult[2].Working = yes) OR
      (a_CURST.QCurst2.Adult[2].JobAway = yes)) THEN
    Jump10;

  BLOCK: h_MAIN;      {Adult 2, first job}
                        {same as e_MAIN}
```

```
IF (jobstat2[pno]=1) THEN
  Jump12;
  BLOCK: i_SUB1;      {Adult 2, second job}
                        {same as e_MAIN}
```

```
IF (jobstat3[pno]=1) THEN
  Jump14;
  BLOCK: j_SUB2;      {Adult 2, third job}
                        {same as e_MAIN}
```

SELF-EMPLOYED PAY DETAILS:

The questions which follow are repeated, with appropriate internal routing, to form:

- BLOCK: k_MAINSF: First job as self-employed**
- BLOCK: l_SUBSF1: Second job as self-employed**
- BLOCK: m_SUBSF2: Third job as self-employed**

IF (jobstat1[1]=2) OR (jobstat1[2]=2) THEN
Jump16;
BLOCK: k_MAINSF;

{Start of table}
IF (session[pno] = 0) AND
(((pmainjob = 1) AND (jobstat1[pno] = 2)) OR
(pmainjob = 2) AND (jobstat2[pno] = 2)) OR
(pmainjob = 3) AND (jobstat3[pno] = 2))) THEN

who have {Ask of all in work as self employed plus those
worked in the last 12 months as self-employed}

IF (((a_CURST.QCurst2.Adult[pno].Working = Yes) OR
(a_CURST.QCurst2.Adult[pno].JobAway = Yes)) OR
(work12m[pno] = 1)) THEN

CARGO4;
Profit1;

IF (Profit1 = RESPONSE) AND (Profit1 > 0) THEN
Profit2;

OwnSum;

IF (OwnSum = Yes) THEN
OwnAmt;

IF (OwnAmt = RESPONSE) AND (OwnAmt > 0) THEN
OwnPd;

know} {If Profit or Loss - not nil profit or don't

IF (Profit2 IN [Profit, Loss]) THEN
Sole;

```
IF (Sole = Partner) THEN
    ParInc;

    IF (ParInc = Yes) THEN
        ParAmt;

{AND
    IF {(Profit1 > 0) AND} (Profit1 = RESPONSE)
        (Profit2 = Profit)} THEN
        Se1;
        Se2;

THEN
    IF (Se1 = RESPONSE) AND (Se2 = RESPONSE)

        SeWks;

{If nil profit, loss or don't know}
IF {(Profit1 = 0) OR (Profit2 = Loss) OR}
    (Profit1 = DONTKNOW) THEN
    SeLWks;

emp'd }
    {if less than 52 wks & not CURRENTLY self
    IF (a_CURST.QCurst2.Adult[pno].Working = No)
    AND
        (a_CURST.QCurst2.Adult[pno].JobAway = No)
    AND
        (SeLWks < 52) THEN
        SeEnd;

OR
    {If in work}
    IF (a_CURST.QCurst2.Adult[pno].Working = Yes)
    THEN
        (a_CURST.QCurst2.Adult[pno].JobAway = Yes)

THEN

    {SelfHr}
    IF (pemploye = 1) OR ((pemploye = 2) AND
        (pmainjob = 1)) THEN

        IF pemploye = 1 THEN
            {EmpHr}
            QHrs;

            IF pmainjob = 1 THEN
                EmpOvt;

        ELSE
            {SelfHr}
            QHrsSelf;
```

- 23 -

```
IF (pmainjob = 1) THEN
  IF (Ushours < 30) THEN
    LikeHr;
  IF (LikeHr = More) THEN
    NoMor;
    IF (Caring IN NoMor) THEN
      NMPer
    IF (Children IN NoMor) THEN
      NMChc;
```

{end of table}

```
IF (jobstat2[1]=2) OR (jobstat2[2]=2) THEN
  Jump17;
  BLOCK: l_SUBSF1; {second job as self-employed}
  {same as k_MAINSF}
```

```
IF (jobstat3[1]=2) OR (jobstat3[2]=2) THEN
  Jump18;
  BLOCK: m_SUBSF2; {third job as self-employed}
  {same as k_MAINSF}
```

```
{Ask if person 1 or person 2 is working at present}
IF (a_CURST.QCurst2.Adult[1].Working = yes) OR
   (a_CURST.QCurst2.Adult[1].JobAway = yes) OR
   (a_CURST.QCurst2.Adult[2].Working = yes) OR
   (a_CURST.QCurst2.Adult[2].JobAway = yes) THEN

Jump19;

BLOCK: n TRAVEL;
CARGO5;

   {QTravel1}
   {Start of table}

   IF (session = 0) THEN
      ttwfar;

      IF ttwfar In [Under1m..Plus25m] THEN
         ttwfrq;

      {end of table}

   {QTravel2}
   {start of table}

   IF (session = 0) THEN
      {Ask of all in work}

      IF (QTravel1.Adult.TtwFar IN [Under1m..Plus25m])
THEN
      TtwMod;

      IF (Public IN TtwMod) OR (Works IN TtwMod) THEN
         TtwPss;

         IF (TtwPss=yes) THEN
            PssAmt;
            PssDate1;
            PssDate2;

         ELSEIF (TtwPss=no) THEN
            Fare;
            OneWay;
```



```
THEN
    IF (car or van IN ttwmod) OR (mcycle IN ttwmod)
        TtwPay;
        IF (TtwPay = Part) THEN
            TtwCode;
            IF (Passnger IN TtwCode) THEN
                TtwCost;
            IF (Driver IN TtwCode) THEN
                TtwRec;
        {end of table}
```

```
{Ask if person 1 or person 2 has ever worked}
IF (a_CURST.QCurst3.Adult[1].LstWrk2 <> '0000') OR
  ((a_CURST.QCurst3.Adult[2].LstWrk2 <> '0000') AND
  (Person2 <> 97)) THEN
Jump20;

BLOCK: o_PENS;

  {QPens2}
  {start of table}
  IF (session = 0) THEN
    {Ask of all except never work at LstWrk2}

    IF a_curst.QCurst3.Adult.LstWrk2 <> '0000' THEN

      IF ((a_curst.QCurst2.adult.Working=yes) OR
        (a_curst.QCurst2.adult.JobAway=yes)) AND
        (jobstat1=1) THEN
        EmpPens;

        IF (EmpPens <> Yes) THEN
          EpPres;

          IF EpPres=Yes THEN
            EpKeep;

            IF EpKeep=Yes THEN
              EpNow;

  {end of table}           {QPens3}
```

```
{start of table}

IF (session = 0) THEN
  {Ask of all except never work at LstWrk2}

  IF a_curst.QCurst3.Adult.LstWrk2 <> '0000' THEN

    IF ((a_CURST.QCurst2.adult.Working=yes) OR
        (a_CURST.QCurst2.adult.JobAway=yes)) AND
        (jobstat1=1) THEN

      IF QPens2.Adult.EmpPens=Yes THEN
        EpLong;
        Serps;
        EpEnd;
        Ep1Avc;

        IF (EpEnd<>Invested) THEN
          EpDes;

      {end of table}

{QPens4}

{Ask of all except never work at LstWrk2}
IF (a_CURST.QCurst3.Adult[1].LstWrk2 <> '0000') AND
  (session[1] = 0) THEN
  EpPrev[1];

IF (a_CURST.QCurst3.Adult[2].LstWrk2 <> '0000') AND
  (session[2] = 0) THEN
  EpPrev[2];
{QPens5}
```

```
{start of table}

IF (session = 0) THEN
  {Ask of all except never work at LstWrk2}

  IF (a_CURST.QCurst3.Adult.LstWrk2 <> '0000') THEN

    IF QPens4.EpPrev=Yes THEN

      IF (jobstat1=1) AND
        (QPens2.Adult.EmpPens=Yes) THEN
        EpTran;

      ELSE
        EpTran2;

        {PrevPen1}
        PrvRgt;

        {If kept right}
        IF PrvRgt = yes THEN
          PrvNum;

          FOR SchNum:=1 TO 3 DO
            IF SchNum <= ORD(PrvNum) THEN
              PrvNow[SchNum]

{end of table}          {QPens6}
```

```
IF ((a_curst.QCurst2.adult[1].Working=yes) OR
    (a_curst.QCurst2.adult[1].JobAway=yes)) AND
    (session[1] = 0) THEN
    PerPen[1];
```

```
IF ((a_curst.QCurst2.adult[2].Working=yes) OR
    (a_curst.QCurst2.adult[2].JobAway=yes)) AND
    (session[2] = 0) THEN
    PerPen[2];
```

```
{QPens7}
{start of table}
```

```
IF session = 0 THEN
```

```
    IF QPens6.PerPen = Yes THEN
```

```
        IF (JOBSTAT1=2) THEN
            PPNum;
            PPDat;
            PPCont;
```

```
        IF (JOBSTAT1=1) THEN
            PPExt;
```

```
        IF (PPCont = yes) OR
            (PPExt = yes) THEN
            PPPpay;
```

```
            IF (PPPpay = RESPONSE) AND (PPPpay > 0) THEN
                PPPd;
```

```
        ELSEIF (PPExt = no) OR (PPExt = DONTKNOW) THEN
            PPEv;
```

```
        IF ((JOBSTAT1=2) AND (PPCont=No)) OR
            ((JOBSTAT1=1) AND (PPEv = Yes)) THEN
            PPLast;
```

```
        PPAge;
```

```
{end of table}
```

BLOCKS: p_BEN1 and p_BEN2

Many of the questions asked about individual benefits follow a regular pattern; they are therefore defined as sub-blocks of the program. When a particular benefit is received, the questions from the appropriate sub-block are then used.

However, to accommodate certain departures from the regular pattern, there may be some internal routing within the sub-blocks such that - although all the questions always exist within the sub-block (and hence space will be allocated for them on the datafile) - particular questions only get asked if a given benefit is being referred to.

This is achieved by defining a variable - 'BenType' - and using it to determine which questions get asked for each benefit. BenType has an identifying code for the benefit currently on the route at any given point in the questionnaire. The range of values is as follows:

<u>Question</u>	<u>Benefit</u>	<u>Code</u>	
Ben1Q	Child Benefit	3	
	One Parent Benefit	4	
	Guardian's Allowance	5	
	Invalid Care Allowance	13	
	Retirement Pension /Old Person's Pension		7
	Widow's Pension	8	
	War Disablement Pension	9	
	Severe Disablement Allowance	10	
	Disability Working Allowance	11	
	DLA care component	1	
Ben2Q	DLA mobility component	2	
	Unemployment Benefit	14	
Ben3Q	Income Support	20	
	Family Credit	19	
	Invalidity Benefit	18	
	Statutory Sick Pay	16	
	NI Sickness Benefit	17	
Ben4Q	Industrial Injury Disablement Benefit	15	
	Maternity Allowance	21	
Ben5Q	Statutory Maternity Pay	23	
	Social Fund Grant for Funeral Expenses		24
	Social Fund Grand for Maternity Expenses		22
	Social Fund Grant for Community Care	25	
	Any other NI/ State Benefit	26	

```
Jump21;  
  
BLOCK: p_BEN1;  
  
{start of table}  
  
IF (session = 0) THEN  
  Ben1Q;  
  Ben2Q;  
  B2QFut;  
  Ben3Q;  
  B3QFut;  
  
  IF (Sex = 2) AND (Age < 55) THEN  
    Ben4Q;  
  
  Ben5Q;  
  
{end of table}
```

NOTE: GENERAL RULES FOR ROUTING TO FOLLOW-UP QUESTIONS BenAmt, BenPd, ImpFlag1, ImpFlag2, ConDoc:

```
IF ((BenType =>1) AND (BenType <=26))  
  BenAmt;  
  
  IF (BenAmt = RESPONSE) AND (BenAmt > 0) THEN  
    BenPd;  
  
  IF (BenType = 1) OR (BenType = 2) OR (BenType =  
14) THEN  
    ImpFlag1;  
    ImpFlag2;  
  
  IF ((BenType >= 1) AND (BenType <= 5)) OR  
    ((BenType >= 9) AND (BenType <= 13)) OR  
    (BenType = 15) OR (BenType = 19) OR (BenType =  
20) OR  
    (BenType = 26) THEN  
    ConDoc;
```

BLOCK: p_BEN2;

{QBen2}
{start of table}

IF (session = 0) THEN

IF (ChildBen IN p_BEN1.QBen1.Adult.Ben1Q) THEN
{ChBenQ;}

BenAmt;
BenPd;
ConDoc

IF (OnePar IN p_BEN1.QBen1.Adult.Ben1Q) THEN
{OneParQ;}

BenAmt;
BenPd;
ConDoc

IF (GuardAll IN p_BEN1.QBen1.Adult.Ben1Q) THEN
{GuardQ;}

BenAmt;
BenPd;
ConDoc

IF (Invalid IN p_BEN1.QBen1.Adult.Ben1Q) THEN
{InvCareQ;}

BenAmt;
BenPd;
ConDoc

ICAPer;

IF (ICAPer = Other) THEN
ICAPerO;


```
IF (NIPens IN p_BEN1.QBen1.Adult.Ben1Q) THEN
  {PensBenQ;}
BenAmt;
BenPd;
  {BenAmtQ;}

  IF (BenAmtQ.BenAmt = RESPONSE) THEN
    Usual;

    IF (Usual = No) THEN
      NotUsAmt;

      IF (NotUsAmt = RESPONSE) AND (NotUsAmt > 0)
THEN
        NotUsPd;

        ConDoc;

        IF (ConDoc = 13) THEN
          {for each of: -
A - Basic pension
B - Basic pension increments
C - Graduated pension
G - Invalidity addition
H - Attendance Allowance
J - Additional pension
K - Contracted out deduction
L - Additional pension, after contracted out deduction
M - Additional pension increments
N - Upgrading of contracted out deduction increments
ask:}

        PenQ;
```

```
IF (WidowBen IN p_BEN1.QBen1.Adult.Ben1Q) THEN
  {WidBenQ;}
BenAmt;
BenPd;

  {BenAmtQ;}

  IF (BenAmtQ.BenAmt = RESPONSE) THEN
    Usual;

  IF (Usual = No) THEN
    NotUsAmt;

    IF (NotUsAmt = RESPONSE) AND (NotUsAmt > 0)
THEN
      NotUsPd;

      ConDoc;

      IF (ConDoc = 13) THEN
        {for each of: -
A - Basic pension
B - Basic pension increments
C - Graduated pension
G - Invalidity addition
H - Attendance Allowance
J - Additional pension
K - Contracted out deduction
L - Additional pension, after contracted out deduction
M - Additional pension increments
N - Upgrading of contracted out deduction increments
ask:}

      PenQ;
```

```
IF (WarPens IN p_BEN1.QBen1.Adult.Ben1Q) THEN  
  {WarPenQ;}
```

```
BenAmt;  
BenPd;  
ConDoc
```

```
IF (SevDisab IN p_BEN1.QBen1.Adult.Ben1Q) THEN  
  {SevAllQ;}
```

```
BenAmt;  
BenPd;  
ConDoc
```

```
IF (DisWork IN p_BEN1.QBen1.Adult.Ben1Q) THEN  
  {DisWorkQ;}
```

```
BenAmt;  
BenPd;  
ConDoc
```

```
IF (DLACare IN p_BEN1.QBen1.Adult.Ben2Q) THEN  
  {DLACarQ1;}
```

```
BenAmt;  
BenPd;  
ImpFlag1;  
ImpFlag2;  
ConDoc
```

```
(if more than 1 adult, or any children in HH)
```

```
IF (Transfev.NewAd > 1) OR (Transfev.NewCh > 0)
```

```
THEN
```

```
  {DLACarQ2;}
```

```
IF (Transfev.NewAd > 1) OR  
(Transfev.NewCh > 0) THEN
```

```
  WhoRec;
```

```
IF (DLAMob IN p_BEN1.QBen1.Adult.Ben2Q) THEN
    {DLAMobQ1;}
BenAmt;
BenPd;
ImpFlag1;
ImpFlag2;
ConDoc

(if more than 1 adult, or any children in HH)
    IF (Transfev.NewAd > 1) OR (Transfev.NewCh > 0)
THEN
    {DLAMobQ2;}

    IF (Transfev.NewAd > 1) OR
    (Transfev.NewCh > 0) THEN
        WhoRec;

IF (AttAll IN p_BEN1.QBen1.Adult.Ben2Q) THEN
    {AttAllQ1;}
BenAmt;
BenPd;
ConDoc

(if more than 1 adult, or any children in HH)
    IF (Transfev.NewAd > 1) OR (Transfev.NewCh > 0)
THEN
    {AttAllQ2;}

    IF (Transfev.NewAd > 1) OR
    (Transfev.NewCh > 0) THEN
        WhoRec;

IF ((AttAll IN p_BEN1.QBen1.Adult.Ben2Q) OR
    (DLACare IN p_BEN1.QBen1.Adult.Ben2Q)) AND
(if only one adult in HH)
    (Transfev.NoFBU = 1) AND (Person2 = 97) THEN
    GetICA;
```

```
IF (DLACare IN p_BEN1.QBen1.Adult.B2QFut) THEN
(if more than 1 adult, or any children in HH)
  IF (Transfev.NewAd > 1) OR (Transfev.NewCh > 0)
THEN
  {CareWho;}

  IF (Transfev.NewAd > 1) OR
  (Transfev.NewCh > 0) THEN
    WhoRec;

IF (DLAMob IN p_BEN1.QBen1.Adult.B2QFut) THEN
  IF (Transfev.NewAd > 1) OR (Transfev.NewCh > 0)
THEN
  {MobWho;}

  IF (Transfev.NewAd > 1) OR
  (Transfev.NewCh > 0) THEN
    WhoRec;

IF (AttAll IN p_BEN1.QBen1.Adult.B2QFut) THEN
  IF (Transfev.NewAd > 1) OR (Transfev.NewCh > 0)
THEN
  {AttWho;}

  IF (Transfev.NewAd > 1) OR
  (Transfev.NewCh > 0) THEN
    WhoRec;

IF (Unemploy IN p_BEN1.QBen1.Adult.Ben3Q) THEN
  {UeBenQ;}
BenAmt;
BenPd;
ImpFlag1;
ImpFlag2;
```

```
IF (IncSupp IN p_BEN1.QBen1.Adult.Ben3Q) THEN  
  ISWeeks;  
  {IncSupQ1;}
```

```
BenAmt;  
BenPd;
```

```
IF (IncSupQ1.BenAmt = RESPONSE) THEN  
  {IncSupQ2;}  
  BenUs;  
  IF BenUs = No THEN  
    BenAmt;  
    IF (BenAmt = RESPONSE) AND (BenAmt > 0) THEN  
      BenPd;
```

```
ConDoc;
```

```
IF (IncSupp IN p_BEN1.QBen1.Adult.Ben3Q) THEN  
  DSSPay;  
  IF (CARDINAL(DSSPay) >= 1) AND  
    (NOT(None IN DSSPay)) THEN  
    DSSAmt;  
    IF (DSSAmt = RESPONSE) AND (DSSAmt > 0)
```

THEN

```
      DSSPd;
```

```
      DSSBefor;
```

```
      SFRepay;
```

```
      IF (SFRepay = Yes) THEN  
        RepayAmt;
```

```
      IF (RepayAmt = RESPONSE) THEN  
        SFInc;
```

```
      ELSEIF (RepayAmt=DONTKNOW) OR  
             (RepayAmt=REFUSAL) THEN
```

```
        LoanNum;
```

```
        IF (LoanNum IN [One..Three]) THEN  
          LoanDatY[1];  
          LoanDatM[1];  
          LoanAmt[1];
```

```
IF (LoanNum IN [Two..Three]) THEN
    LoanDatY[2];
    LoanDatM[2];
    LoanAmt[2];

IF (LoanNum IN [Three]) THEN
    LoanDatY[3];
    LoanDatM[3];
    LoanAmt[3];

IF (FamCred IN p_BEN1.QBen1.Adult.Ben3Q) THEN
    ContinFC;

    {FamCredQ;}
    BenAmt;
    BenPd;
    ConDoc

    IF (InvalBen IN p_BEN1.QBen1.Adult.Ben3Q) THEN
        {InvBenQ;}
        BenAmt;
        BenPd;

    IF (StatSick IN p_BEN1.QBen1.Adult.Ben3Q) AND
        (((Perno = 1) AND ((e_MAIN.SSPSMP <> RESPONSE) OR
        (e_MAIN.SSPSMP = 3) OR (e_MAIN.SSPSMP = 4))) OR
        ((Perno = 2) AND ((h_MAIN.SSPSMP <> RESPONSE) OR
        (h_MAIN.SSPSMP = 3) OR (h_MAIN.SSPSMP = 4))))
THEN
    {StatsQ;}
    BenAmt;
    BenPd;

DONTKNOW) IF (StatsQ.BenAmt <> RESPONSE) OR (StatsQ.BenPd =
    OR (StatsQ.BenPd = REFUSAL) THEN
        SSPRat;
    IF (NISick IN p_BEN1.QBen1.Adult.Ben3Q) THEN
        {NISickQ;}
        BenAmt;
        BenPd;
```

```
IF (InvalBen IN p_BEN1.QBen1.Adult.Ben3Q) OR
  ((StatSick IN p_BEN1.QBen1.Adult.Ben3Q) AND
   (StatSQ.BenAmt <> EMPTY)) OR
  (NISick IN p_BEN1.QBen1.Adult.Ben3Q) THEN
  MadEmp;

  IF (MadEmp = yes) THEN
    MduPwk;

    IF (MduPwk = SomeWeek) THEN
      MduPNo;

  IF (IndDisab IN p_BEN1.QBen1.Adult.Ben3Q) THEN
    {IndDisQ;}
  BenAmt;
  BenPd;
  ConDoc

  IF (MatAllw IN p_BEN1.QBen1.Adult.Ben4Q) THEN
    {MatAllQ;}
  BenAmt;
  BenPd;

  IF (MatStat IN p_BEN1.QBen1.Adult.Ben4Q) AND
    (((Perno = 1) AND ((e_MAIN.SSPSMP <> RESPONSE) OR
      (e_MAIN.SSPSMP = 2) OR (e_MAIN.SSPSMP = 4))) OR
     ((Perno = 2) AND ((h_MAIN.SSPSMP <> RESPONSE) OR
      (h_MAIN.SSPSMP = 2) OR (h_MAIN.SSPSMP = 4))))
  THEN
    {MatStatQ;}
  BenAmt;
  BenPd;

  IF (MatStatQ.BenAmt <> RESPONSE) OR
    (MatStatQ.BenPd = DONTKNOW) OR
    (MatStatQ.BenPd = REFUSAL) THEN
    MatRat;

  MatEmp;
  MatStp;
```


IF (FCLump IN p_BEN1.QBen1.Adult.Ben5Q) THEN
FCLumpQ;

IF (Funeral IN p_BEN1.QBen1.Adult.Ben5Q) THEN
{FuneralQ;}
BenAmt;

IF (MatGrant IN p_BEN1.QBen1.Adult.Ben5Q) THEN
{MatGrntQ;}
BenAmt;

IF (ComCare IN p_BEN1.QBen1.Adult.Ben5Q) THEN
{CareAmtQ;}
BenAmt;

IF (OthBen IN p_BEN1.QBen1.Adult.Ben5Q) THEN
OthPres;
OthWeeks
{OthBenQ;}

BenAmt;
BenPd;
ConDoc

{end of table} {QBen4;}

```
IF session[1] = 0 THEN
  Ben7Q[1];
IF session[2] = 0 THEN
  Ben7Q[2];
{QBen5}

{start of table}
IF session = 0 THEN
  IF (TradUn IN QBen4.Ben7Q) THEN
    {TUBenQ1;}
    Pres;
    NumWeeks:
    {TUBenQ2;}
    {BenAmt, BenPd}

  IF (FrSoc IN QBen4.Ben7Q) THEN
    {FRBenQ1;}
    Pres;
    NumWeeks;
    {FRBenQ2;}
    {BenAmt, BenPd}

  IF (PrivSick IN QBen4.Ben7Q) THEN
    {PSBenQ1;}
    Pres;
    NumWeeks;
    {PSBenQ2;}
    {BenAmt, BenPd}

  IF (AccIns IN QBen4.Ben7Q) THEN
    {ACCBenQ1;}
    Pres;
    NumWeeks;
    {ACCBenQ2;}
    {BenAmt, BenPd}

  IF (HospSav IN QBen4.Ben7Q) THEN
    {HSBenQ1;}
    Pres;
    NumWeeks;

    {HSBenQ2;}
    {BenAmt, BenPd}
{end of table}
```

```
{QBen6;}  
{start of table}
```

```
IF session = 0 THEN
```

```
  IF (A_CURST.QCurst1.Adult.Train IN [YT..Other]) THEN  
    PrgAmt;
```

```
    IF (PrgAmt = RESPONSE) AND (PrgAmt > 0) THEN  
      PrgPd;
```

```
{end of table}
```

BLOCK: q_OTHIN1;

{QOIncA4;}

IF (session = 0) THEN

or
NotWk and {Ask of all who are on a Government training scheme
AvSeek/AvNSeek/InjSeek/InjWait/Wait at question
who have worked in last 12 months}

IF ((a_curst.QCurst1.Adult.train IN [ET..Other]) OR
(a_curst.QCurst2.adult.Start=Yes) OR
(a_curst.QCurst2.adult.Start=Yes) AND
(a_curst.QCurst2.adult.LikeWk=Yes)) OR
(a_curst.QCurst2.adult.Start=No) AND
(a_curst.QCurst2.adult.YStart IN
[TempSick..LongSick]) AND
(a_curst.QCurst2.adult.LikeWk=EMPTY)) OR
(a_curst.QCurst2.adult.JobAway=Waiting) OR
(belowpe2=1) AND
(a_curst.QCurst2.adult.Wait=Yes)))) AND
(work12m = 1) AND
(a_curst.QCurst2.adult.JobAway=no) THEN

RedAny;

IF RedAny=Yes THEN

RedAmt;

```
IF (session[1] = 0) OR (session[2] = 0) THEN  
  Jump24;
```

```
{QOIncA5;}  
IF (session[1] = 0) THEN  
  AnyPen[1];  
  
  IF (Annuity IN AnyPen[1]) THEN  
    PenLump[1];
```

```
{QOIncA6;}  
IF (session = 0) THEN  
  IF (EmpPens IN QOIncA5.AnyPen) THEN  
    FOR loopvar:= 1 TO 4 DO  
      IF (loopvar = 1) OR (Another = yes) THEN  
        IF (loopvar = 1) THEN  
          EmpInt[loopvar];
```

```
    PenPay;
```

```
    IF (PenPay = RESPONSE) AND (PenPay > 0) THEN  
      PenPd;
```

```
    PenTax;
```

```
    IF PenTax=Yes THEN  
      PTAmt;  
      PTInc;
```

```
    IF (PenType = 1) OR (PenType = 2) THEN  
      PenOth;
```

```
      IF (PenOth = Yes) THEN  
        PoTyp;  
        PoAmt;  
        PoSour;  
        PoInc;
```

```
    IF ((PenType = 1) AND (loopvar <> 4)) OR  
      (((PenType = 2) OR (PenType = 3)) AND  
      (loopvar <> 3)) THEN  
      Another;
```

```
IF (SpouPens IN QOIncA5.AnyPen) THEN
  FOR loopvar:= 1 TO 3 DO
    IF (loopvar = 1) OR
      (SpouPrev[loopvar-1].Another = yes) THEN
      IF (loopvar = 1) THEN
        SpouInt[loopvar];
        {SpouPrev[loopvar];}
        {repeat PenPay - Another}
      IF (PersPens IN QOIncA5.AnyPen) THEN
        FOR loopvar:= 1 TO 3 DO
          IF (loopvar = 1) OR
            (PersPen[loopvar - 1].Another = Yes) THEN
            IF (loopvar = 1) THEN
              PerInt[loopvar];
              {PersPen[loopvar];}
              {repeat PenPay - Another}

IF (TUPens IN QOIncA5.AnyPen) THEN
TUInt;
PenPay;
PenPd;

IF (Annuity IN QOIncA5.AnyPen) THEN
AnnInt;
PenPay;
PenPd;
```

PenTax;

IF PenTax=Yes THEN

PTAmt;

PTInc;

IF (Trust IN QOIncA5.AnyPen) THEN

TrustInt;

PenPay;

PenPd;

PenTax;

IF PenTax=Yes THEN

PTAmt;

PTInc;

TRights;

```
{QOIncA7;}
{start of table}

IF (session = 0) THEN
  Royal;

  IF Roy IN Royal THEN
    RoyYr [1];

  IF Sleep IN Royal THEN
    RoyYr [2];

  IF Pens IN Royal THEN
    RoyYr [3];

{end of table}

IF (session[1] = 0) OR (session[2] = 0) THEN
  Jump25;

{QOIncA8;}
{start of table}

IF (session = 0) THEN
  {Ask of all who are married with partner not in
household}
  IF (Marstat = 2) THEN
    AbsPar;

    IF (AbsPar = Yes) THEN
      ApAmt;

      IF (ApAmt = RESPONSE) AND (ApAmt > 0) THEN
        ApPd;

      ApDir;

      IF (ApDir = Yes) THEN
        ApHH;
        ApdAmt;

        IF (ApdAmt = RESPONSE) AND (ApdAmt > 0) THEN
          ApdPd;
    {end of table}
```



```
BLOCK: r_OTHIN2;
    {QOIncB1;}
    IF (session[1] = 0) THEN
        Allow[1];
    IF session[2] = 0 THEN
        Allow[2];

    {QOIncB2;}
    {start of table}
    IF (session = 0) THEN
        FOR Count:= 1 TO 4 DO
            IF (Count IN QOIncB1.allow) THEN
                {QAllow[Count];}

                AllPay;

                IF (AllPay = RESPONSE) AND (AllPay > 0) THEN
                    AllPd;

    {end of table}
```

```
IF (session[1] = 0) OR (session[2] = 0) THEN  
  Jump26;
```

```
{QOIncB31; - for perno = 1}  
IF (session = 0) THEN
```

```
  MntRec;           IF (MntRec <> Skip) THEN  
    MntCt;
```

```
  IF MntRec=Yes THEN
```

```
    MntDSS;  
    MntAmt;
```

```
    IF (MntAmt = RESPONSE) AND (MntAmt > 0) THEN  
      MntPd;
```

```
    IF (MntPd IN [OneWeek..Year]) THEN  
      MntUs;
```

```
      IF (MntUs = No) THEN  
        MntUsAmt;
```

```
0) THEN  
      IF (MntUsAmt = RESPONSE) AND (MntUsAmt >
```

```
        MntUsPd;
```

```
    MntTotal;  
    MntFor;
```

```
MntPay;
IF (MntPay = Yes) THEN
  {QMaint; table for each set of maintenance
    payments,
      max 4}
  FOR loopvar:= 1 TO 4 DO
    IF (loopvar = 1) OR
      (Maint[loopvar - 1].M = Yes) THEN
      {Maint[loopvar];}
      MrR;
      IF (MrR IN [Child..Both]) THEN
        MrKid;
        MrAge;
        MrAmt;
        IF (MrAmt = RESPONSE) AND (MrAmt > 0)
          THEN
            MrPd;
            IF (MrAmt = RESPONSE) THEN
              MrUs;
              IF (MrUs = No) THEN
                MrUAmt;
                IF (MrUAmt = RESPONSE) AND (MrUAmt >
                  0) THEN
                    MrUPd;
                    MrCt;
                    MrV;
                    IF (MrV IN [YesUp..YesDown]) THEN
                      MrChWhy;
                      IF (loopvar < 4) THEN
                        M;
                        {QOIncB32; - for perno = 2, repeat MatRec - M}
```

```
IF (session[1] = 0) OR (session[2] = 0) THEN
  Jump27;

{QOIncB5;}
IF session = 0 THEN
  Fuel;

  IF (Fuel = no) THEN
    FCash;

    IF (FCash = yes) THEN
      FCAmt;

      IF (FCAmt = RESPONSE) AND (FCAmt > 0) THEN
        FCAmtPd;

{QOIncB6;}
{start of table}

IF session = 0 THEN
  Baby;

  IF (Babysit IN Baby) OR (Mail IN Baby) THEN
    BabPay;
    BabNow;

{end of table}

{QOIncB7;}
{start of table}

IF session = 0 THEN
  OthTax;

  IF OthTax=Yes THEN
    OtAmt;
    OtSour;

{end of table}
{start of table}          {QOIncB8;}
```

```
IF session = 0 THEN
  RefTax;
```

```
  IF RefTax=Yes THEN
    RefAmt;
    RefPay;
    RefUe;
```

```
    IF RefUe=No THEN
      RefWhy;
```

```
{end of table}
```

```
{QOIncB9;}
```

```
{start of table}
```

```
IF session = 0 THEN
```

```
{*** GOVT TRAINING WITH NO PAID WORK ***}
```

```
{*** MEN UNDER 66/WOMEN UNDER 61 - SELF-EMPLOYED ***}
```

```
{*** MEN UNDER 66/WOMEN UNDER 61 - WITH NO PAID WORK
```

```
***}
```

```
  IF ((a_CURST.QCurst1.Adult.train IN [ET..Other]) AND
      (a_CURST.QCurst2.Adult.JobAway = No)) OR
      ((belowpen=1) AND
      (jobstat1<>1) AND
      (jobstat2<>1) AND
      (jobstat3<>1) OR
      (a_CURST.QCurst2.Adult.JobAway = No)) THEN
    NIREg;
```

```
  IF (NIREg = Yes) THEN
    NIAmt;
```

```
    IF (NIAmt = RESPONSE) AND (NIAmt > 0) THEN
      NIPd;
```

```
{end of table}
```

```
{QOIncB10;}
```

```
{start of table}
```

```
IF session = 0 THEN
  NILump;
```

```
  IF NILump=Yes THEN
    NILAmt;
    NILWhy;
```

```
{end of table}
```

```
{QOIncB11;}

IF session[1] = 0 THEN
  OddJob[1];

IF session[2] = 0 THEN
  OddJob[2];

THEN IF (QOIncB11.OddJob[1]=Yes) OR (QOIncB11.OddJob[2]=Yes)
  Jump28;

  {QOIncB12;}
  {start of table}

  IF (session = 0) AND
    (QOIncB11.OddJob = Yes) THEN

    {for each 'odd job', max 3}

    FOR loopvar:=1 TO 3 DO

      IF (loopvar=1) OR (OjOther[loopvar-1]=yes) THEN
        OjTyp[loopvar];
        OjAmt[loopvar];

        IF (OjAmt[loopvar] = RESPONSE) AND
(OjAmt[loopvar] > 0) THEN
          OjPd[loopvar];

          IF loopvar <> 3 THEN
            OjOther[loopvar];

      {end of table}
```

```
IF Childnum > 0 THEN
  Jump29;

  BLOCK: s_CHINC;

    FOR cno := 1 TO 8 DO

      IF (Chnos[cno] <> 0) THEN
        {Child[cno]}

          IF (session[1] = 0) OR (session[2] = 0) THEN
            CARGO7;
            ChEarn;

            IF ChEarn=Yes THEN
              {ChildJob;}
              Ident;
              ChYr;

              IF ChYr=No THEN
                ChWk;

                ChAmt;
                IF (ChAmt = RESPONSE) AND (ChAmt > 0)
                THEN
                  ChPd;

            ChTst;
            IF ChTst=Yes THEN
              {ChildTr;}
              Ident;
              ChYr;

              IF ChYr=No THEN
                ChWk;

                ChAmt;
                IF (ChAmt = RESPONSE) AND (ChAmt > 0)
                THEN
                  ChPd;
```

```
Jump30;      BLOCK: t_ADINT;

{q_ADINT1;}

IF (session[1] = 0) THEN
    Accounts [1];

IF session[2] = 0 THEN
    Accounts [2];

{q_ADINT2;}
{start of table}

IF session = 0 THEN

    IF NOT(None IN q_AdInt1.accounts) THEN

        IF (curracc in q_adint1.accounts) THEN
            {curracc;}
            IF (AccType = 1) THEN
                AccIntr;

            ELSEIF (AccType = 3) THEN
                InvIntr;

        IF (nsbord in q_adint1.accounts) THEN
            {nsboacc;}
            IF (AccType = 1) THEN
                AccIntr;

            ELSEIF (AccType = 3) THEN
                InvIntr;

        IF (nsbinv in q_adint1.accounts) THEN
            {nsbiacc;}
            IF (AccType = 1) THEN
                AccIntr;

            ELSEIF (AccType = 3) THEN
                InvIntr;
```



```
IF (tessa in q_adint1.accounts) THEN
  {TessaInt;}
  IF (AccType = 1) THEN
    AccIntr;

    ELSEIF (AccType = 3) THEN
      InvIntr;

IF (buildA in q_adint1.accounts) THEN
  {buildsoA;}
  IF (AccType = 1) THEN
    AccIntr;

    ELSEIF (AccType = 3) THEN
      InvIntr;

IF (buildB in q_adint1.accounts) THEN
  {buildsoB;}
  IF (AccType = 1) THEN
    AccIntr;

    ELSEIF (AccType = 3) THEN
      InvIntr;

IF (highstA in q_adint1.accounts) THEN
  {highstbA;}
  IF (AccType = 1) THEN
    AccIntr;

    ELSEIF (AccType = 3) THEN
      InvIntr;

IF (highstB in q_adint1.accounts) THEN
  {highstbB;}
  IF (AccType = 1) THEN
    AccIntr;

    ELSEIF (AccType = 3) THEN
      InvIntr;
```

- 58 -

```
IF (othsavA in q_adint1.accounts) THEN
  {othsavA;}
  IF (Acctype = 1) THEN
    AccIntr;

    ELSEIF (Acctype = 3) THEN
      InvIntr;

  IF (othsavB in q_adint1.accounts) THEN
    {othsavB;}
    IF (Acctype = 1) THEN
      AccIntr;

      ELSEIF (Acctype = 3) THEN
        InvIntr;

{end of table}
```

```
IF (session[1] = 0) OR (session[2] = 0) THEN
  Jump31;
  {q_ADINT3;}
  IF (session[1] = 0) THEN
    invests[1];
  IF (session[2] = 0) THEN
    invests[2];

  {q_ADINT4;}
  {start of table}
  IF (session = 0) THEN

    IF NOT(None IN q_adint3.invests) THEN

      IF (giltA in q_adint3.invests) THEN
{giltedgA;}
        IF (Acctype = 1) THEN
          AccIntr;
        ELSEIF (Acctype = 3) THEN
          InvIntr;

      IF (giltB in q_adint3.invests) THEN
{giltedgB;}
        IF (Acctype = 1) THEN
          AccIntr;
        ELSEIF (Acctype = 3) THEN
          InvIntr;

      IF (unit in q_adint3.invests) THEN
{unitint;}
        IF (Acctype = 1) THEN
          AccIntr;
        ELSEIF (Acctype = 3) THEN
          InvIntr;

      IF (stocksA in q_adint3.invests) THEN
{stockA;}
        IF (Acctype = 1) THEN
          AccIntr;
        ELSEIF (Acctype = 3) THEN
          InvIntr;

      IF (stocksB in q_adint3.invests) THEN
{stockB;}
        IF (Acctype = 1) THEN
          AccIntr;

        ELSEIF (Acctype = 3) THEN
          InvIntr;




```

```
{q_ADINT5;}  
  
IF (session[1] = 0) THEN  
    Otinva[1];  
  
IF (session[2] = 0) THEN  
    Otinva[2];
```

```
IF Childnum > 0 THEN
  Jump32;

  BLOCK: u CHINT;
  CARGO10;

  {QChint1;}
  FOR cno := 1 TO 8 DO
    IF (Chnos[cno] <> 0) THEN
      {Child[cno]}
      IF (session[1] = 0) OR (session[2] = 0) THEN
        Ident;
        chinc;

  {QChint2;}
  FOR cno := 1 TO 8 DO
    IF (Chnos[cno] <> 0) THEN
      {Child[cno]}
      IF (session[1] = 0) OR (session[2] = 0) THEN

        IF (poord in QChint1.Child[cno].chinc) THEN
          chint [1];

        IF (poinv in QChint1.Child[cno].chinc) THEN
          chint [2];

        IF (buildac in QChint1.Child[cno].chinc)
THEN
          chint [3];

        IF (bankac in QChint1.Child[cno].chinc) THEN
          chint [4];

        IF (govgilt in QChint1.Child[cno].chinc)
THEN
          chint [5];

        IF (untrust in QChint1.Child[cno].chinc)
THEN
          chint [6];

        IF (othstock in QChint1.Child[cno].chinc)
THEN
          chint [7];
```

```
{QChint3;}
FOR cno := 1 TO 8 DO
    IF (Chnos[cno] <> 0) THEN
        {Child[cno]}
        IF (session[1] = 0) OR (session[2] = 0) THEN
            otinvc;

    {QChint4;}
    FOR cno := 1 TO 8 DO
        IF (Chnos[cno] <> 0) THEN
            {Child[cno];}
            IF (session[1] = 0) OR (session[2] = 0) THEN
                IF (NOT(None IN QChint1.Child[cno].Chinc)
                    (QChint1.Child[cno].Chinc = RESPONSE)) OR
                    (NOT(None IN QChint3.Child[cno].Otinvc)
                    (QChint3.Child[cno].Otinvc = RESPONSE))
                THEN
                    TotSave;

                    IF TotSave = To20000 THEN
                        ChName;

    IF (session[1] = 0) AND ((session[2] = 0) OR (Person2 =
97)) THEN
        BLOCK: v_SAVE;

        {Only ask totsav if some savings, ie something in
iadint1,
iadint3, iadint5 (apart from private loan)
(ie something in savfils 1 or 2) }

        IF (len(savfil1) > 2) OR (len(savfil2) > 2) THEN
            Totsav;

    IF (session[1] = 1) OR ((session[2] = 1) AND (Person2 <>
97)) THEN
        HaltDisp;
```

BLOCK: w_ASSETS;

Appropriate parts of the Assets block are asked only iuf the adult(s) in the Benefit Unit reported having between £1500 and £20000 at the question 'TotSav', and/or if one or more child was reported as having between £1500 and £20000.

Where there is more than one child in the Benefit Unit, each child is assessed separately for entry in to the Assets block, at 'TotSav'.

Each section of the Assets block is asked of each qualifying person as appropriate, depending on the liquid assets they are coded as having in the blocks t_Adint and U-Chint.

IF (assets = 1) OR (chasset = 1) THEN

CARGO11;

{Any adult with a bank or building society current account}

Anymon;

IF (Anymon=yes) OR (Countp > 1) THEN

MuchLeft;

AccName;

IF (Countp < MaxNo) THEN

More;

{Any adult with NSB ordinary /investment account; TESSA; high street bank or building society savings account; other bank or building society account}

IF (asset[PersCont,2]=1) OR (asset[PersCont,3]=1) OR
(asset[PersCont,4]=1) OR (asset[PersCont,5]=1) OR
(asset[PersCont,6]=1) OR (asset[PersCont,7]=1)

THEN

{Children with NSB ordinary /investment account; bank or building society account}

{QCSaveAC[PersCont];}
IF (chass[PersCont,2]=1) OR (chass[PersCont,3]=1) OR
(chass[PersCont,5]=1) OR (chass[PersCont,6]=1)

THEN

Intro;

IF (Intro = Still) OR (countp > 1) THEN

Kindof;

AccDisp;

AccName;

MuchAcc;

IF (Countp < MaxNo) THEN

More;

FOR PersCont:= 1 TO 2 DO

{Adults with gilts; unit trust/investment trust; stocks and shares}

IF (asset[PersCont,8]=1) OR (asset[PersCont,9]=1) OR
(asset[PersCont,10]=1) THEN

{Children with gilts; unit trust/investment trust; stocks and shares}

IF (chass[PersCont,8]=1) OR (chass[PersCont,9]=1) OR
(chass[PersCont,10]=1) THEN

Intro;

IF (Intro = Still) OR (countp > 1) THEN

NameOf;

HowMany;

HowHold;

IF (Edit = 'Yes') THEN

Spare;

IF (Countp < MaxNo) THEN

More;

FOR PersCont:= 1 TO 2 DO

{Adults with National Savings Certificates - index-linked and fixed interest}

{QACertif[PersCont];}

IF (asset[PersCont,13]=1) OR (asset[PersCont,14]=1)

THEN

{Children with National Savings Certificates - index-linked and fixed interest}

{QCCertif[PersCont];}

IF (chass[PersCont,13]=1) OR (chass[PersCont,14]=1)

THEN

Intro;

IF (Intro = Still) OR (countp > 1) THEN

Issue;

Issdate;

Issval;

IF (Edit = 'Yes') THEN

Spare;

IF (Countp < MaxNo) THEN

More;

```
FOR pcount:= 1 TO 2 DO
{Adults with Pensioners Guaranteed Income Bond}
    IF (asset[pcount,11]=1) THEN
        {Adult[pcount]}
        Intro;
        IF (Intro = Still) THEN
            PGIBVal;

{Adults with Save As You Earn (SAYE) schemes}
    FOR PersCont:= 1 TO 2 DO
        {QSAYE[PersCont];}
        IF (asset[PersCont,16]=1) THEN
            I;
            IF (I = Still) OR (countp > 1) THEN
                T;
                IF (T=NatSav) OR (T=DONTKNOW) THEN
                    IssHold;
                    SAYEDat;
                    Amount;
                    IF (Amount = RESPONSE) AND (Amount > 0)
THEN
                        Pd;
                        AmtNow;
                        IF (Edit = 'Yes') THEN
                            Spare;
                            IF (Countp < 5) THEN
                                More;
```

{QPremium;}

{Adults with Premium Bonds}

FOR pcount:= 1 TO 2 DO
IF (asset[pcount,17]=1) THEN

{Children with Premium Bonds}

FOR pcount:= 1 TO 8 DO
IF (chass[pcount,17]=1) THEN

Prem;

{Adults with National Savings Income Bonds}

FOR pcount:= 1 TO 2 DO
IF (asset[pcount,18]=1) THEN

{Children with National Savings Income Bonds}

FOR pcount:= 1 TO 8 DO
IF (chass[pcount,18]=1) THEN

Intro;

IF (Intro = Still) THEN

NSIB;

NSIBVal;

{Adults with National Savings Capital Bonds / Deposit Bonds}

```
FOR PersCont:= 1 TO 2 DO
  {QABonds[PersCont];}
  IF (asset[PersCont,12]=1) OR (asset[PersCont,19]=1)
THEN
```

{Children with National Savings Capital Bonds / Deposit Bonds}

```
FOR PersCont:= 1 TO 8 DO
  {QCBonds1[PersCont];}
  IF (chass[PersCont,12]=1) OR (chass[PersCont,19]=1)
THEN
  FOR chno:= 1 TO 8 DO
```

```
    Intro;
    IF (Intro = Still) OR (countp > 1) THEN
      BondDat;
      BondVal;
      IF (Edit = 'Yes') THEN
        Spare;
      IF (Countp < MaxNo) THEN
        More;
```

{Children with Childrens Bonus Bonds}

```
IF (chass[PersCont,15]=1) THEN
  FOR chno:= 1 TO 8 DO
```

```
    Intro;
    IF (Intro = Still) OR (countp > 1) THEN
      BondDat;
      BondVal;
      IF (Edit = 'Yes') THEN
        Spare;
      IF (Countp < MaxNo) THEN
        More;
```

```
IF (session[1] = 0) AND ((session[2] = 0) OR (Person2 = 97))
THEN
  BLOCK: x_END;

  Address;
  Telephon;

  IF Telephon = Yes THEN
    TelNo1;
    TelNo2;

  Future1;

  IF AdultNum = 2 THEN
    Future2;

  Thank;

  Finish;
```

BLOCK NONRESPONSE (ADMIN)

```
Ar03;      { HIDDEN }
Ad03      { HIDDEN }
HH03;     { HIDDEN }
BU03;     { HIDDEN }
IssueNo;  { HIDDEN }
IntNo;    { HIDDEN }
CARGO;
AdmNote;
IStatus;  { HIDDEN }

{CodeSOC;}
IF ((Edit = 'Yes') OR (SCPR <> 'Yes')) AND
  ((c_JOBDES.QJobdes.Subj1.Empee = RESPONSE) OR
   (d_JOBDES.QJobdes.Subj1.Empee = RESPONSE)) THEN
  SOCNow;

IF (SOCNow = Now) AND ((Edit = 'Yes') OR (SCPR <> 'Yes'))
THEN

  IF (c_JOBDES.QJobdes.Subj1.Empee = RESPONSE) THEN
    SOC1;

  IF (Edit = 'Yes') THEN
    RevEmp1;
    Disp1;
    SocCls1;
    SEG1;

  IF (d_JOBDES.QJobdes.Subj1.Empee = RESPONSE) THEN
    SOC2;

  IF (Edit = 'Yes') THEN
    RevEmp2;
    Disp2;
    SocCls2;
    SEG2;      Choice;
vChoice;    { HIDDEN }
vThank;    { HIDDEN, = x_END.Thank,
            to check outcome against last question }
```

```
IF (Choice = Admin) OR (vChoice = Admin) THEN
  Iout1; { code as DONTKNOW if not really finished }

  IF (Iout1 <> Full) THEN
    Iout1Txt;

  IF (Person2 <> 97) THEN
    Iout2;

    IF (Iout2 <> Full) THEN
      Iout2Txt;

  IntDone;

  IF (Edit = 'Yes') THEN
    PABDone;

  IF (Edit <> 'Yes') THEN
    INFO;
```

**ROUTING DOCUMENTATION: FRS HOUSEHOLD
QUESTIONNAIRE**

VERSION 31: APRIL 1994

BLOCK: I_044_1

**ArNum;
AdNum;
HHNum;
IssueNo;
First;
IntNum;
LAC;
MnthCode;
YearCode;
cargo1;**

**MaxRef;
CurrRef;
MaxDK;
CurrDK;**

BLOCK: SetOne

IntDate;
OrigDate;
IntSTime;
Adult;
VAdult; {protects Adult}

BLOCK: a_HHA

{For each adult - maximum 10}

Person;
Name;
Vname; {protects Name}

IF (SUBSTRINT(Name,1,2)<>'XX') AND
(SUBSTRING(Name,1,2)<>'xx') THEN
{If person has not been xx'd out; that is, removed from Adult grid}

Sex;
Age;

FOR RelNr:= 1 to 9 DO
IF (RelNr < PersonNr) THEN
{Code relationship of each adult to those in the grid before them}
R[RelNr];

FOR RelNr:= 1 to 10 DO
CR[RelNr];

MS;

IF (MS = widowed) AND (Sex = female) THEN
W1;
W2;

IF (Age) IN [16..18] THEN
FtEd;

IF (Age > 18) OR (FtEd = No) THEN
TEA;

IF (TEA = 96) OR (FtEd = Yes) THEN
{if still in continuous full-time education,
TEA = 96}
TypeEd;

Depend;

BLOCK: b_Eth

{For each adult that has not been xx'd out}
FOR PersonNr := 1 TO 10 DO

IF (SetOne.Adult >= PersonNr) AND
 (SUBSTRING(a_HHA.Ad[PersonNr].Name,1,2) <> 'XX')
 AND
 (SUBSTRING(a_HHA.Ad[PersonNr].Name,1,2) <> 'xx')
 THEN
 EthGrp;

BLOCK: Children

NumChil;
VChil; {protects NumChil}

IF (Children.NumChl > 0) THEN

BLOCK: c_HHC

{for each child - maximum 10}

Person;

Name;

VName; {protects Name}

IF (SUBSTRING(Name,1,2) <> 'XX')

AND (SUBSTRING(Name,1,2) <> 'xx') THEN

{if child has not been xx'd out; that is, removed from Child grid}

Sex;

Age;

IF (SetOne.Adult > 1) THEN

Par1;

IF (Par1 IN [1,10]) THEN

{if Parent 1 is a member of the household}

Par2;

IF (ParPart = 1) OR

((ParPart = 2) AND (Par2 <> 97)) THEN

{if parents outside h'hold or parents not partners}

Resp1;

IF (ParPart = 1) THEN

{if parents outside h'hold}

Resp2;

Foster;

FTEd;

IF (FTEd = Yes) THEN

School

BLOCK: BenUnit

FOR Index1:=1 TO 10 DO

IF (Index1 <= SetOne.Adult) AND
(a_HHA.Ad[Index1].Depend = DepAd) AND
(SUBSTRING(a_HHA.Ad[Index1].name,1,2) <> 'XX') AND
(SUBSTRING(a_HHA.Ad[Index1].name,1,2) <> 'xx')
THEN

IF (Parent1[Index1] <> 0) AND
(Parent2[Index1] <> 0) AND
(a_HHA.Ad[Parent1[Index1]].R[Parent2[Index1]]
<> Partner) AND
(a_HHA.Ad[Parent2[Index1]].R[Parent1[Index1]]
<> Partner) THEN

LegDep[Index1];

FOR Index1:=1 TO 20 DO

{Assign adult/child to each person}

AC[Index1];

FOR Index1:=1 TO 10 DO

IF (Index1 <= SetOne.Adult) AND
(SUBSTRING(a_HHA.Ad[Index1].Name,1,2) <> 'XX') AND
(SUBSTRING(a_HHA.Ad[Index1].Name,1,2) <> 'xx')
THEN

{Assign benefit unit for each adult}

ABen[Index1];

IF Children.NumChil > 0 THEN

FOR Index1:=11 TO 20 DO

IF (Index1 <= (Children.NumChil+10)) AND
(SUBSTRING(c_HHC.Ch[Index1].Name,1,2) <> 'XX')
AND
(SUBSTRING(c_HHC.Ch[Index1].Name,1,2) <> 'xx')
THEN

{Assign benefit unit for each child}

CBen[Index1];

NewBU; {amended number of B.U.s (only confirmed h'hold members)}

IF (Changed = 1) THEN

NewInfo;

IF (NewInfo <> RESPONSE) THEN

BenUInfo;

ShowBen;

BLOCK: TF (TRANSFER PARAGRAPH)

{Details transferred from Household Questionnaire to
Benefit Unit Questionnaire, for purposes of routing,
textfills, administering checks & warnings, etc.}

cargo2;
Tissue;

FOR Ind1:= 1 TO 10 DO
 DepChild[Ind1];

FOR Ind1:= 1 TO 10 DO
 APerson[Ind1];
 AName[Ind1];
 ASex[Ind1];
 AAge[Ind1];
 AMS[Ind1];
 TE[Ind1];
 DC[Ind1];
 ABU[Ind1];
 ATEA[Ind1];

FOR Ind1:= 1 TO 20 DO
 CPerson[Ind1];
 CName[Ind1];
 CSEX[Ind1];
 CAGE[Ind1];
 CBU[Ind1];

FOR Ind1:= 1 TO 20 DO
 Guard1[Ind1];
 Guard2[Ind1];

FOR Ind1:= 1 TO 20 DO

DPerson[Ind1];
DName[Ind1];
DSex[Ind1];
DAge[Ind1];
DBU[Ind1];
DGuard1[Ind1];
DGuard2[Ind1];

FOR Ind1:= 1 TO 20 DO

EName[Ind1];

OrigAd;
NewAd;
OrigCh;
NewCh;
ODate;
IDate;
DatYrAgo;
INum;
BUNew;
TOutsPay;
TOutsAmt;
TOutsPd;
TAccPay;
TAccAmt;
TAccPd;

BLOCK: d_ADDINF

cargo3;
HHolder;
Tenure;

IF (Tenure = Shared) THEN
 OwnType;

SubLet;

IF (SubLet = Yes) THEN
 SubLetY;

Rooms;
RoomShar;
Bedroom;
BusRoom;

IF (BusRoom = Yes) THEN
 OnBsRoom;
 PtBsRoom;

AllAdult;
AllChild;
{the real number of adults and children are stored,
triggers the computations for variables ValidAd
and ValidChild}

TypeAcc;

IF TypeAcc = Other THEN
 TypeAcco;

ELSEIF TypeAcc IN [Flat,PartHse] THEN
 Floor;

SewSupp;
YearLive;

IF (YearLive = 0) THEN
 MonLive;

IF (BenUnit.NewBU > 1) THEN
 HHStat;

vHHStat; {equals HHStat, but with imputed value if
not on route}

IF (d_ADDINF.Tenure IN [CoOwns..RentFree]) THEN
BLOCK: e_RENTER

Cargo4;
Landlord;

IF (Landlord IN [FrndRel..OthIndiv]) THEN
Furnish;
ResLL;

IF (ResLL = Yes) AND (d_ADDINF.TypeAcc = Flat)
THEN
ResLL2;

IF (ResLL = No) OR (ResLL2 = No) THEN
YStart;

IF (YStart = Aft1988) THEN
Ctract;

IF (YStart = Aft1988) AND
(Ctract IN [Signed..NotSign]) THEN
Short;

IF (YStart = Bef1988) THEN
FairRent;

IF ((ResLL = Yes) AND (ResLL2 = EMPTY)) OR
(ResLL2 = Yes) OR (Short = NotShort) OR
(Short = DONTKNOW) OR (Short = REFUSAL) THEN
OthWay;

AccJob;

IF (d_AddInf.Tenure IN [Rents,Part,Shared,CoOwns])
THEN
Rent;

IF (Rent = RESPONSE) AND (Rent > 0) THEN
RentPd;

IF (Rent > 0) OR (Rent = DONTKNOW) OR
(Rent = REFUSAL) THEN
CommInc;

IF (CommInc=Yes) THEN
CommAmt;

IF (CommAmt = RESPONSE) AND (CommAmt > 0) THEN
CommPd;

IF (Area >= 15921) THEN {If in Scotland}
CWatAmt;

IF (CWatAmt = RESPONSE) AND (CWatAmt > 0)
THEN
CWatPd;

RentHol;

IF (RentHol=Yes) THEN
WeekHol;

HBenefit;

IF (Rent<0.01) AND (HBenefit = Yes) THEN
Rebate;

IF ((Rent<0.01) AND (HBenefit = No)) OR
((Rent<0.01) AND (HBenefit = Yes) AND
(Rebate=No)) THEN
RebateO;

IF (HBenefit = Yes) THEN
IF NOT(Landlord IN [Council]) THEN
HBenFuRd;

IF (HBenFuRd = Reduced) THEN
HBenRAmt;

IF (HBenRAmt = RESPONSE) AND
(HBenRAmt > 0) THEN
HBenRPd;

HBenAmt;

IF (HBenAmt = RESPONSE) AND (HBenAmt > 0.00) THEN
HBenPd;

IF (HBenAmt = RESPONSE) THEN
HBenDed;

IF (HBenDed = Deduct) AND (Rent = RESPONSE)
THEN **HBenChk;**

HBStmt;

IF (HBStmt = Yes) THEN

HBWater;

HBSEwer;

IF (HBWater = Yes) OR (HBSEwer = Yes) THEN

HBWSAmt;

SpareSer; {HIDDEN, spare columns for poss
more deductions}

HBServ;

IF NOT(None IN HBServ) THEN

FOR Index:= 1 TO 11 DO

IF (Index IN HBServ) THEN

HBSerAmt[Index];

HBsvOth;

IF (HBsvOth = Yes) THEN

HBSOther;

IF (HBSOther = RESPONSE) THEN

HBSOAmt;

EligAmt;

IF (EligAmt = RESPONSE) AND (EligAmt > 0) THEN

EligPd;

ELSEIF (HBenefit = No) THEN

HBenWait

```
IF (d_AddInf.Tenure IN [Rents,Part,Shared,CoOwns])
THEN

IF (HBenefit <> Yes) OR ((HBStmt = RESPONSE) AND
(HBStmt <> Yes)) THEN

IF (Area < 15921) THEN {Not in Scotland}

IF NOT(d_ADDINF.SewSupp IN
[Sewerage,Neither]) THEN
WaterInc;

IF NOT(d_ADDINF.SewSupp IN
[Water,Neither]) THEN
SewerInc;

IF (WaterInc = Yes) OR (SewerInc = Yes) THEN
WSIncAmt;

IF (HBServ = EMPTY) THEN
SerInc;

IF (SerInc = Yes) THEN
SerIncW

IF NOT (SerInc = No) THEN
ServAmt;

IF (d_AddInf.Tenure IN [Rents,Part,Shared,CoOwns]) OR
((d_AddInf.Tenure = RentFree) AND
(AccJob <> Yes)) THEN
AccNonHH;

IF (AccNonHH = Yes) THEN
AccPay;

FOR Index:= 1 TO 4 DO

IF (Index <= CARDINAL(AccPay)) THEN
AccAmt[Index];

IF (AccAmt[Index] = RESPONSE) AND
(AccAmt[Index] > 0) THEN
AccPd[Index];

IF (Rent = RESPONSE) AND
(AccAmt[Index] <> 0) THEN
AccChk[Index];
```

IF (d_ADDINF.Tenure IN [Owns..Part]) THEN
BLOCK: f_OWNER1

LastPay[1];
LastPay[2];
{variables for textfill (hidden)}

IF (d_AddInf.Tenure IN [Owns..Part]) THEN
OwnHow;

IF (OwnHow = Mortgage) THEN
Lender;

IF (Lender = Other) THEN
LenderO;

BuyYear;

{if mortgage is for purchase}
IF (Ask = 1) THEN
PurcAmt;

BorrAmt;

IF (Worth = 1) THEN
{EstWorth not asked/answered before}
EstWorth;

RMort;

IF (RMort = Yes) THEN
RMortYr;
RMAMt;
RMPur;

MortType;

TopUp;

IF (TopUp = Yes) THEN for each Top Up - Maximum 3
TopAmt;
TopYr;
TopPur;
TopMore;

MortEnd;
MortLeft;
Discount;

```
IF (MortType In [Endow,Pension..None] THEN
{payments cover interest only}
MorInPay;

IF (MorInPay = RESPONSE) AND (MorInPay > 0)
THEN
MorInPd;

Menpol;

IF (Menpol = Yes) THEN
{for each endowment policy - maximum 4}
MenPolAm;

IF (MenPolAm = RESPONSE) AND
(MenPolAm > 0) THEN
MenPolPd;

InclnInt;
Menstyr;

IF (Count < 4) THEN
MpMore;

ELSE
EndwPrin;

ELSE {payments cover interest & loan}
IntPrPay;

IF (IntPrPay = RESPONSE) AND (IntPrPay > 0)
THEN IntPrPd;

IntL12M;

IF (IntL12M = RESPONSE) AND (IntL12M > 0) THEN
IntPerB;

IF (IntPerB = RESPONSE) THEN
IntPerE;

TaxRelf;
MortProt;

IF (Mortprot=Yes) THEN
IncMPAmt;

IF (IncMPAmt = RESPONSE) AND (IncMPAmt > 0)
THEN IncMPPd;
```

IncMStYr;
IncMP;

OutsMort;

IF (OutsMort = Yes) THEN
OutsPay;

FOR Count:=1 to 4 DO

IF (Count <= CARDINAL(OutsPay)) THEN
OutsAmt[Count];

IF (OutsAmt[Count] = RESPONSE)
AND (OutsAmt[Count] > 0) THEN
OutsPd[Count];

IF (OutsAmt[Count] <> 0) THEN
OutsIncl[Count];

OthMort1;

ELSEIF (OwnHow = Outright) THEN
OthMort2;

IF (OthMort1 = Yes) OR (OthMort2 = Yes) THEN
... repeat from Lender to OutsIncl ...

OthMort3;

EstValue;

BLOCK: g_Insur

IF (f_Owner1.OwnHow = Mortgage) OR
(f_Owner1.OthMort2 = Yes) THEN
StrMort;

IF (StrMort = Yes) THEN
StrCov;

StrAmt; {MortStr}

IF (StrAmt = RESPONSE) AND (StrAmt > 0) THEN
StrPd;

IF (StrAmt = DONTKNOW) OR (StrPd = DONTKNOW) THEN

IF (StrCov = Struct) OR
(StrCov = Combine) OR
(CovOths = Struct) OR
(CovOths = Combine) THEN
StrIns

IF (StrCov=Furn) OR (StrCov=Combine) OR
(CovOths=Combine) THEN
FrnIns

{All except Council renters and those answering yes to
StrMort - i.e. include non-Council renters, outright
owners, unknown tenure, people answering no or don't
know to StrMort}

{OR those who only contents insurance tied with mortgage}

IF ((d_ADDINF.Tenure = DONTKNOW) OR
(d_ADDINF.Tenure = REFUSAL)) OR
((e_RENTER.Landlord IN [NewTown..OthIndiv]) OR
(e_RENTER.Landlord = DONTKNOW) OR
(e_RENTER.Landlord = REFUSAL)) OR
((f_Owner1.OwnHow = Outright) OR
(f_Owner1.OwnHow = DONTKNOW) OR
(f_Owner1.OwnHow = REFUSAL)) THEN

IF (StrMort = EMPTY) OR (StrMort = No) OR
(StrCov = Furn) THEN
StrOths;

IF (StrOths = Yes) THEN
CovOths;

StrAmt; {OthStr}

IF (StrAmt = RESPONSE) AND (StrAmt > 0) THEN
StrPd;

IF (StrAmt = DONTKNOW) OR (StrPd = DONTKNOW) THEN

IF (StrCov = Struct) OR (StrCov = Combine) OR
(CovOths = Struct) OR (CovOths = Combine)
THEN
StrIns

IF (StrCov=Furn) OR (StrCov=Combine) OR
(CovOths=Combine) THEN
FrnIns

{Questions about Council Tax}

CTBand;

IF (CTBand IN [BandB..BandH]) OR (CTBand = DONTKNOW) OR
(CTBand = REFUSAL) THEN

CTLVBand;

IF (CTBand IN [BandA..BandH]) OR (CTBand = DONTKNOW) OR
(CTBand = REFUSAL) THEN

CTAmt;

IF (CTAmt = RESPONSE) AND (CTAmt > 0) THEN

CTTime;

IF (CTAmt = 0) THEN

CTExReb;

IF (CTAmt <> 0) OR (CTExReb = DiscReb) THEN

CTDisc;

IF (CTDisc = Yes) THEN

CT25D50D;

CTReb;

IF (CTReb = Yes) THEN

RebType;

CTRebAmt;

IF (CTRebAmt = RESPONSE) AND
(CTRebAmt > 0) THEN

CTRebPd;

CTRed;

IF (CTRed = Yes) THEN

CTRedAmt;

IF (CTRedAmt = RESPONSE) AND
(CTRedAmt > 0) THEN

CTRedPd;

IF (d_AddInf.Tenure IN [Owns..Part]) THEN
BLOCK: h_Owner2

Cargo5;

Charge;

IF (Other IN Charge) THEN

ChargeO;

IF {xxxxxxx} IN Charge THEN

{for each item coded at Charge, (except None)}

ChAmt;

If (ChAmt = RESPONSE) AND (ChAmt > 0) THEN

ChargePd;

BLOCK: i_Sewer

Cargo6;

IF (d_ADDINF.SewSupp IN [Both..Water]) THEN

{If not in Scotland}

IF (Area < 15921) THEN

IF (e_Renter.WaterInc <> Yes) AND
(e_Renter.HBWater <> Yes) AND
(d_ADDINF.SewSupp IN [Both,Water]) THEN
WaterPay;

IF (e_Renter.SewerInc <> Yes) AND
(e_Renter.HBSewer <> Yes) AND
(d_ADDINF.SewSupp IN [Both,Sewerage]) THEN
SewerPay;

IF (WaterPay = Yes) AND (SewerPay = Yes) THEN
SewSep;

IF (SewSep = Separate) OR ((WaterPay = Yes) AND
(SewerPay <> Yes)) THEN
WatTime;
WatAmt;

IF (SewSep = Separate) OR ((SewerPay = Yes) AND
(WaterPay <> Yes)) THEN
SewTime;
SewAmt;

IF (SewSep = Combined) THEN
WSewTime;
WSewAmt;

```
{if more than one benefit unit}
IF (BenUnit.NewBU > 1) THEN
  BLOCK: j_BENGIV

      {if 'conventional' household}
  IF (d_AddInf.HHStat = Conv) THEN
    ConvIn;

    FOR Count:= 2 TO 6 DO

      IF (Chosen1[Count] <> 0) THEN

        IF (a_HHA.Ad[Chosen1[Count]].R[1] IN
          [FChild,FParent,FSib,GChild..NonRel])
          THEN
            iID1;
            BUIId;
            ConvBL;

            IF (ConvBL IN [Board..Lodg]) THEN
              CvPay;

              IF (CvPay = RESPONSE) AND
                (CvPay > 0) THEN
                CvPd;

              IF (ConvBL = Lodg) THEN
                CvHt;

              CvHb;

              IF (CvHb = Yes) THEN
                CHBAmt;

                IF (CHBAmt = RESPONSE) AND
                  (CHBAmt > 0) THEN
                  CHBPd;
```

```
IF (a_HHA.Ad[Chosen1[count]].R[1]
  IN [Partner..StChild,ILChild..StParent,
      ILParent..StSib,ILSib])
  {not a boarder or lodger, or DK/refuse}
OR (ConvBL = Neith)
OR (ConvBL = DONTKNOW)
OR (ConvBL = REFUSAL) THEN
Contrib;

IF (Contrib = Yes) THEN
ConAmt;

  IF (ConAmt = RESPONSE) AND
    (ConAmt > 0) THEN
ConPd;

  IF (ConAmt <> 0) THEN
QConlt;
```

{if 'non-conventional', ie. shared household}
IF (d_AddInf.HHStat = Shared) THEN

FOR Count:= 1 TO 10 DO

IF (Count <= SetOne.Adult) AND
(SUBSTRING(a_HHA.Ad[Count].Name,1,2)
<> 'XX') AND
(SUBSTRING(a_HHA.Ad[Count].Name,1,2)
<> 'xx') THEN

{for each adult in the house}

pID2;

BUId;

IF (d_ADDINF.Tenure IN [CoOwns..RentFree])
AND

{All sharing renters}

(BenUnit.ABen[Count] > 1) THEN

{in BU:s 2+ }

SRentAmt;

IF (SRentAmt = RESPONSE) AND

(SRentAmt > 0) THEN

SRentPd;

IF (BenUnit.ABen[Count] > 1) AND

{in BU:s 2+ }

(NOT(Count IN d_ADDINF.Hholder) OR

(d_ADDINF.Tenure IN [CoOwns..RentFree]))

THEN

SCvHb;

IF (SCvHB = Yes) THEN

SCHBAmt;

IF (SCHBAmt = RESPONSE) AND (SCHBAmt > 0)

THEN

SCHBPd;

SharePay;

IF (SharePay = RESPONSE) AND

(SharePay > 0) THEN

SharePd;

IF (SharePay > 0) THEN

ShareIt;

BLOCK: k_prop

IF (d_AddInf.SubLet = Yes) THEN

SubRent;

SubAllow;

OthProRt;

IF (OthProRt = Yes) AND (d_ADDINF.AllAdult > 1) THEN

PropWho;

{for each person receiving any rent from other property}

IF (OthProRt = Yes) THEN

FOR Index:= 1 TO 10 DO

IF (Index IN PropWho) OR

((Index = 1) AND (d_ADDINF.AllAdult = 1)) THEN

jlD1[Index];

PropRent[Index];

PropAllw[Index];

BLOCK: I_pol

Premium;

IF (Premium = Yes) THEN

FOR index:= 1 TO 6 DO
{for each policy, maximum 6}

Cargo7;
NumPols;
PolIns;
PolPay;

IF (index < 6) THEN

PolMore;

BLOCK: m_Mod

FOR Count:=1 TO 13 DO
{for washing machine, tumble dryer, fridge freezer, ...
etc.}

Cons[Count];

CentHeat;

IF (CentHeat = Yes) THEN

CentFuel;

BLOCK: n_VehInt

AnyVeh;

IF (AnyVeh = Yes) THEN

VehNumb;

FOR Index:= 1 TO 9 DO

IF VehNumb >= Index THEN

{for each vehicle}

cargo8;

Vehic;

VehOwn;

IF VehOwn=Own THEN

VehPer

ELSEIF VehOwn=Use THEN

VehUse;

IF Vehic IN [car, van_] THEN

VehProv;

IF VehProv IN [Employer, PartEmp] THEN

VehCC ;

VehFuel;

IF (Vehic=car) AND (VehOwn=RESPONSE) THEN

VehEmp

BLOCK: o_Welf

Prscrpt;

IF Prscrpt=Yes THEN

{for up to 5 people who have had free prescriptions}
FOR Index1:=1 TO 5 DO

IF Index1 <= TotP THEN

IF (Index1=2) OR ((Index1 > 2) AND
(FPIntro[Index1-1]=Yes)) THEN
FPIntro[Index1]

IF (Index1=1) OR (FPIntro[Index1]=Yes) THEN

IF (d_AddInf.AllAdult>1) OR
((d_AddInf.AllAdult=1) AND
(d_AddInfo.AllChild>0)) THEN
PrsPer[Index1];

Prslt[Index1];

{if there are any people in household under 60}

IF (LEN(Under60) > 0) THEN

WelfMilk;

IF WelfMilk=Yes THEN

{for up to 5 people who have had welfare milk}

FOR Index1:=1 TO 5 DO

IF Index1 <= Tot60 THEN

IF (d_AddInf.AllAdult>1) OR
((d_AddInf.AllAdult=1) AND
(d_AddInf.AllChild>0)) THEN

IF (Index1=2) OR ((Index1 > 2) AND
(WMIntro[Index1-1]=Yes)) THEN
WMIntro[Index1]

IF (Index1=1) OR (WMIntro[Index1]=Yes) THEN

IF (d_AddInf.AllAdult>1) OR
((d_AddInf.AllAdult=1) AND
(d_AddInf.AllChild>0)) THEN
WmkPer[Index1];

Wmklf[Index1];

IF Children.NumChil > 0 THEN

{if any of the children are attending a State school}
IF (LEN(SS16) > 0) THEN

SchMilk;

IF SchMilk=Yes THEN

{for up to 5 children who have had free
school milk}

FOR Index1:=1 TO 5 DO

IF Index1 <= Tot16 THEN

IF ((Index1=2) AND
(d_AddInf.AllChild>1)) OR
((Index1 > 2) AND
(SMIntro[Index1-1]=Yes)) THEN
SMIntro[Index1]

IF (Index1=1) OR (SMIntro[Index1]=Yes) THEN

IF d_AddInf.AllChild >= 1 THEN
SmkPer[Index1]
Smklt[Index1];

{if there is anyone under 19 at State school}
IF (LEN(SS19) > 0) THEN

SchMeal;

IF SchMeal=Yes THEN

{for up to 5 people who have had free
school meals}

FOR Index1:=1 TO 5 DO

IF Index1 <= Tot19 THEN

IF (Index1=2) OR
((Index1 > 2) AND
(MLIntro[Index1-1]=Yes)) THEN
MLIntro[Index1];

IF (Index1=1) OR (MLIntro[Index1]=Yes) THEN
MLPer[Index1];
Smlt[Index1];

BLOCK: p_Educ

Grant;

IF (Grant = Yes) THEN

GrtPer;

{for each person receiving a grant, max. 5}

FOR Index:= 1 TO 5 DO

IF (Chosen1[Index] <> 0) THEN

pID1[Index];

GrtNum[Index];

GrtSce1[Index];

IF (GrtSce1[Index] = State) THEN

GrtAmt1[Index]

ELSEIF (GrtSce1[Index] IN [Private,Overseas])

THEN **GrtVal1[Index]**

IF (GrtSce1[Index] = RESPONSE) THEN

GrtDir1[Index]

IF (GrtNum[Index] IN [Two,Three]) THEN

GrtSce2[Index];

IF (GrtSce2[Index] = State) THEN

GrtAmt2[Index]

ELSEIF (GrtSce2[Index]

IN [Private,Overseas]) THEN

GrtVal2[Index]

IF (GrtSce2[Index] = RESPONSE) THEN

GrtDir2[Index]

{if any adult in household is less than 50}

IF (o50 = 1) THEN

TopUp;

IF (TopUp = Yes) THEN

TUPer;

{for each person eligible for a top up loan,
max. 5}

FOR Index:= 1 TO 5 DO

IF (Chosen2[Index] <> 0) THEN

pID2[Index];

TUEnt[Index];

TUBorr[Index];

Loan;

IF Loan=Yes THEN

EDPer;

{for each person receiving an educational loan,
max 5}

FOR Index:= 1 TO 5 DO

IF (Chosen3[Index] <> 0) THEN

pID3[Index];

LoanNum[Index];

Ed1Borr[Index];

Ed1MonYr[Index];

Ed1Sum[Index];

Ed1Amt[Index];

IF (Ed1Amt[Index] = RESPONSE) AND

(Ed1Amt[Index] > 0) THEN

Ed1Pd[Index];

Ed1Int[Index];

IF (Ed1Amt[Index]>0) AND

(Ed1Int[Index]=Interest) THEN

Ln1RpInt[Index]

IF (LoanNum[Index] IN [Two,Three]) THEN

Ed2Borr[Index];

Ed2MonYr[Index];

Ed2Sum[Index];

Ed2Amt[Index];

IF (Ed2Amt[Index] = RESPONSE) AND

(Ed2Amt[Index] > 0) THEN

Ed2Pd[Index];

Ed2Int[Index];

IF (Ed2Amt[Index]>0) AND

(Ed2Int[Index]=Interest) THEN

Ln2RpInt[Index]

{if there is at least one person over 25 in
the household}

IF (u25 > 0) THEN

NHHChild;

{for up to 4 children (16 - 24), outside the h'hold,
in education}

IF (NHHChild = Yes) THEN

FOR Index:=1 TO 4 DO

IF (Index=2) OR ((Index > 2) AND
(NHHIntro[Index-1]=Yes)) THEN
NHHIntro[Index]

IF (Index=1) OR (NHHIntro[Index]=Yes) THEN
NHHName[Index];
NHHPar[Index];
NHHEd[Index];
NHHFee[Index];

IF (NHHFee[Index] = Yes) THEN
NHHAmt[Index];

IF (NHHAmt[Index] = RESPONSE) AND
(NHHAmt[Index] > 0) THEN
NHHPd[Index];

BLOCK: q_Cares

IF d_AddInf.AllChild > 0 THEN

{for each child in the household, max 10}
FOR Index2:=1 TO 10 DO

IF (AskWork[BenUnit.CBen[Index2+10]] = 0) THEN
Wrk;

IF (AskWork[BenUnit.CBen[Index2+10]] = 1) THEN
Disp;
Care;

IF (Care = Yes) THEN
ChLook;

IF (ChMind IN ChLook) THEN
Registrd[1];

IF (Nursery IN ChLook) THEN
Registrd[2];

ChHr1;
ChHr2;
Cost;

IF (Cost = Yes) THEN

IF (ChHr1 > 0) THEN
ChAmt1;

IF (ChHr2 > 0) THEN
ChAmt2;

ChPay;

NeedHelp;
GiveHelp;

IF (NeedHelp = Yes) OR (GiveHelp = Yes) THEN
NeedPer;

IF (Other IN NeedPer) THEN
NeedPerO;

FOR Idx:= 1 TO 8 DO

IF Idx<= CARDINAL(NeedPer) THEN
HelpId;
Freq;

IF (Freq IN [Cont..OWeek]) THEN
WhoLook;
Long;
NeedTask;

FOR Count:= 1 TO 10 DO

IF (Count IN WhoLook) AND (Count <> 11)
THEN **Hour[Count];**

IF (Kids IN WhoLook) THEN
Hour[11];

FOR Count:= 12 TO 14 DO

IF (Count IN WhoLook) AND (Count <> 11)
THEN **Hour[Count];**

EndDisp;

BLOCK: NONRESPONSE (ADMIN)

```
ArNum;      { HIDDEN }
AdNum;      { HIDDEN }
HHNum;      { HIDDEN }
IssNum;     { HIDDEN }
IntNum;     { HIDDEN }
NOFBU;      { HIDDEN }
HAdult;     { HIDDEN }
HHEnd;      { HIDDEN }
CARGO;      { HIDDEN }
HStatus;    { HIDDEN }
AdmNote;
Choice;
vChoice;    { HIDDEN - protects Choice }
```

```
IF (Choice = Admin) OR (vChoice = Admin) THEN
```

```
  IF (HHNum = 1) THEN
    NOFHH;
```

```
      IF (SCPR = 'Yes') AND (NOFHH >= 4) AND
        (NOFHH <= 12) THEN
        HHCode1;
        HHCode2;
        HHCode3;
```

```
{CALLS;   call details table }
{           for each call, max 18}

FOR CallNo := 1 TO 18 DO

    IF ( MORE = 'Y' ) THEN

        {CALL[CallNo];}
        CallDat;
        CallDOW; { SCREEN }
        CallTime;
        CallRes;
        Dummy;

        IF ( CallRes IN [AnyDone,Checking] ) THEN
            CallDur;

        IF ( CallNo < 18 ) THEN
            CallMore;

Hout; { code as DONTKNOW if not really finished }

IF (Hout IN [OthRef,OthSix,HOHRef,HOHSix,ContRef]) THEN
    RefR;

    IF (Other IN RefR) THEN
        RefRTxt;

IF (Hout IN [OthNoC,NoContac]) THEN
    NCR;

    IF (Other IN NCR) THEN
        NCRTxt;

MinsAdm;
Notes;

IntDone; { ensure Hout has value }

IF (Edit = 'Yes') THEN
    PABDone;
```