Documentation of Questionnaire/Module 'FRS0605B' on 27-04-2006 at 15:04

FRS0605B

FAMILY RESOURCES SURVEY 2006-2007

COMPUTE ALWAYS:

```
NatCen := ONS
```

COMPUTE ALWAYS:

```
Edit := No
```

COMPUTE ALWAYS:

```
Test := No
```

COMPUTE ALWAYS:

```
VerCode := '056_1'
```

COMPUTE ALWAYS:

```
TestVer := '01'
```

COMPUTE ALWAYS:

```
SuppTxt := ('Please record the reasons why you suppressed ' +
'this warning, then press <Alt> + S to save and continue.')
```

COMPUTE ALWAYS:

```
Pd97Txt := ('Please leave a note/remark giving full details '
+ 'then press <Alt> + S to save and continue.')
```

COMPUTE ALWAYS:

```
KeyTxt := ('This is a 'Key Question': It is VERY ' +
'IMPORTANT to get an answer here if possible. If you cannot '
+ 'do so (either now, or later) please make a Note about the '
+ 'circumstances.')
```

COMPUTE IF: NatCen = ONS

```
N := ''
```

COMPUTE IF: NatCen = ONS

```
I := ''
```

COMPUTE IF: NatCen = ONS

```
B := ''
```

COMPUTE IF: NatCen = ONS

```
X := 'H'
```

COMPUTE IF: NatCen = ONS

O1 := '('

COMPUTE IF: NatCen = ONS

O2 := ')'

COMPUTE IF: NatCen = ONS

IC := 'i'

COMPUTE IF: NatCen = ONS

IS := 'N'

COMPUTE IF: NatCen = ONS

BLU := ''

COMPUTE IF: NatCen = ONS

Anyone_Else := 'Anyone else'

COMPUTE IF: NatCen = ONS

Any_Others := 'Any others'

COMPUTE IF: NatCen = ONS

Any_Other := 'Any other'

COMPUTE IF: NOT (NatCen = ONS)

N := ''

COMPUTE IF: NOT (NatCen = ONS)

I := ''

COMPUTE IF: NOT (NatCen = ONS)

B := ''

COMPUTE IF: NOT (NatCen = ONS)

X := '*'

COMPUTE IF: NOT (NatCen = ONS)

O1 := '('

COMPUTE IF: NOT (NatCen = ONS)

O2 := ')'

COMPUTE IF: NOT (NatCen = ONS)
AND: Edit = Yes

IC := 'EDITOR:'

COMPUTE IF: NOT (NatCen = ONS)
    AND: NOT (Edit = Yes)

IC := 'INTERVIEWER:'

COMPUTE IF: NOT (NatCen = ONS)

IS := 'SHOW CARD'

COMPUTE IF: NOT (NatCen = ONS)

BLU := ''

COMPUTE IF: NOT (NatCen = ONS)

Anyone_Else := 'Who else'

COMPUTE IF: NOT (NatCen = ONS)

Any_Others := 'Which others'

COMPUTE IF: NOT (NatCen = ONS)

Any_Other := 'Which other'

RECORD ALWAYS:

FRS0605B.IVers

^I Version code of interview program, eg. I_048_1. I = Interview, 04 = month, 7 = year, 1 = release.^I

STRING[7]

RECORD ALWAYS:

FRS0605B.EVers

^I Version code of edit program, eg. E_047_1. E = Edit, 04 = month, 7 = year, 1 = release.^I

STRING[7]

COMPUTE IF: Test = Yes
    AND: NatCen = NI

IVers := ('NI_' + TestVer)

COMPUTE IF: Test = Yes
    AND: NOT (NatCen = NI)

IVers := ('I_' + VerCode)

COMPUTE IF: Edit = Yes

EVers := ('E_' + VerCode)

COMPUTE IF: NOT (Edit = Yes)

IVers := ('I_' + VerCode)

COMPUTE IF: Edit = Yes

```
Interviewer := 'Editor'
```

COMPUTE IF: Edit = Yes

```
EditVersion := ('Edit version@|@|:@|' + EVers + '
')
```

COMPUTE IF: NOT (Edit = Yes)

```
Interviewer := 'Interviewer'
```

COMPUTE IF: NOT (Edit = Yes)

```
EditVersion := '
'
```

COMPUTE ALWAYS:

```
Days[1] := 'Sunday'
```

COMPUTE ALWAYS:

```
Days[2] := 'Monday'
```

COMPUTE ALWAYS:

```
Days[3] := 'Tuesday'
```

COMPUTE ALWAYS:

```
Days[4] := 'Wednesday'
```

COMPUTE ALWAYS:

```
Days[5] := 'Thursday'
```

COMPUTE ALWAYS:

```
Days[6] := 'Friday'
```

COMPUTE ALWAYS:

```
Days[7] := 'Saturday'
```

COMPUTE ALWAYS:

```
Months[1] := 'January'
```

COMPUTE ALWAYS:

```
Months[2] := 'February'
```

COMPUTE ALWAYS:

```
Months[3] := 'March'
```

COMPUTE ALWAYS:

```
Months[4] := 'April'
```

COMPUTE ALWAYS:

```
Months[5] := 'May'
```

COMPUTE ALWAYS:

```
Months[6] := 'June'
```

COMPUTE ALWAYS:

```
Months[7] := 'July'
```

COMPUTE ALWAYS:

```
Months[8] := 'August'
```

COMPUTE ALWAYS:

```
Months[9] := 'September'
```

COMPUTE ALWAYS:

```
Months[10] := 'October'
```

COMPUTE ALWAYS:

```
Months[11] := 'November'
```

COMPUTE ALWAYS:

```
Months[12] := 'December'
```

COMPUTE ALWAYS:

```
AssDo := No
```

COMPUTE ALWAYS:

```
BookDo := No
```

COMPUTE ALWAYS:

```
NCDVLP := No
```

COMPUTE ALWAYS:

```
NCDVIB := 0
```

COMPUTE ALWAYS:

```
NCDVOB := 0
```

COMPUTE ALWAYS:

```
NCDVDC := No
```

Compute always:

```
NCDVTC := No
```

Compute always:

```
NCDVCP := 0
```

Compute always:

```
NCDVAW := No
```

Compute always:

```
NCDVRT := No
```

Compute always:

```
NCDVAA := No
```

FRS0605B.QSerial

Serial number

ASK IF: Test = Yes

FRS0605B.QSerial.Area

AREA NUMBER.

^B JUST PRESS <Enter>^B.

1..99997

ASK IF: Test = Yes

FRS0605B.QSerial.Address

ADDRESS NUMBER.

^B JUST PRESS <Enter>^B.

1..97

ASK IF: Test = Yes

FRS0605B.QSerial.Hhold

HOUSEHOLD NUMBER.

^B JUST PRESS <Enter>^B.

1..3

CHECK IF: Test = Yes
RESERVECHECK

RESERVECHECK

CHECK IF: Test = Yes
RESERVECHECK

RESERVECHECK

CHECK IF: Test = Yes
RESERVECHECK

RESERVECHECK

COMPUTE IF: Test = Yes
AND: DArea > 0

Area := DArea

COMPUTE IF: Test = Yes
AND: DAddress > 0

Address := DAddress

```
COMPUTE IF: Test = Yes
     AND: DHhold > 0
```

Hhold := DHhold

```
COMPUTE IF: Test = Yes
     AND: DHhold > 0
```

FRS0605B.QSerial

Serial number

ASK IF: *NOT (Test = Yes)*

FRS0605B.QSerial.Area

AREA NUMBER.

^B JUST PRESS <Enter>^B.

1..99997

ASK IF: *NOT (Test = Yes)*

FRS0605B.QSerial.Address

ADDRESS NUMBER.

^B JUST PRESS <Enter>^B.

1..97

ASK IF: *NOT (Test = Yes)*

FRS0605B.QSerial.Hhold

HOUSEHOLD NUMBER.

^B JUST PRESS <Enter>^B.

1..3

CHECK IF: *NOT (Test = Yes)*
RESERVECHECK

RESERVECHECK

CHECK IF: *NOT (Test = Yes)*
RESERVECHECK

RESERVECHECK

CHECK IF: *NOT (Test = Yes)*
RESERVECHECK

RESERVECHECK

COMPUTE IF: *NOT (Test = Yes)*
AND: *DArea > 0*

Area := DArea

COMPUTE IF: *NOT (Test = Yes)*
AND: *DAddress > 0*

Address := DAddress

```
COMPUTE IF: NOT (Test = Yes)
     AND: DHhold > 0
```

Hhold := DHhold

```
COMPUTE IF: NOT (Test = Yes)
     AND: DHhold > 0
```

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

COMPUTE IF: QDataBag.SSTRTReg IN [22 .. 27]

NIreland := No

COMPUTE IF: QDataBag.SSTRTReg IN [22 .. 27]

Scotland := Yes

COMPUTE IF: QDataBag.SSTRTReg IN [22 .. 27]

Wales := No

COMPUTE IF: QDataBag.SSTRTReg IN [20 .. 21]

NIreland := No

COMPUTE IF: QDataBag.SSTRTReg IN [20 .. 21]

Scotland := No

COMPUTE IF: QDataBag.SSTRTReg IN [20 .. 21]

Wales := Yes

COMPUTE IF: QDataBag.SSTRTReg IN [30]

NIreland := Yes

COMPUTE IF: QDataBag.SSTRTReg IN [30]

Scotland := No

COMPUTE IF: *QDataBag.SSTRTReg IN [30]*

Wales := No

COMPUTE IF: *NOT (QDataBag.SSTRTReg IN [30])*

NIreland := No

COMPUTE IF: *NOT (QDataBag.SSTRTReg IN [30])*

Scotland := No

COMPUTE IF: *NOT (QDataBag.SSTRTReg IN [30])*

Wales := No

COMPUTE IF: *Test = Yes*
AND: *NICoun = RESPONSE*

NIDCoun := ORD(NICoun)

COMPUTE IF: *QDataBag.NICoun IN [1 .. 97]*

NIDCoun := QDataBag.NICoun

COMPUTE IF: *QDataBag.NIRate IN [0.01 .. 9997]*

NIRate := QDataBag.NIRate

COMPUTE IF: *NatCen = NI*

SharOwn := 'co-ownership'

COMPUTE IF: *NatCen = NI*

SOwners := 'Co-owners'

COMPUTE IF: *NatCen = NI*

LANIHE := 'NIHE'

COMPUTE IF: *NatCen = NI*

Council1 := 'Northern Ireland Housing Executive'

COMPUTE IF: *NatCen = NI*

Council2 := 'Northern Ireland Housing Executive'

COMPUTE IF: *NatCen = NI*

GOVSSA := 'Social Security Agency'

COMPUTE IF: *NatCen = NI*

GOV1 := 'SSA'

COMPUTE IF: *NatCen = NI*

GOV2 := 'Social Security Agency'

COMPUTE IF: NatCen = NI

```
JobCen := 'a Social Security Office'
```

COMPUTE IF: NatCen = NI

```
RentReb1 := 'rent and/or rates rebate'
```

COMPUTE IF: NatCen = NI

```
RentReb2 := 'rent/rates rebate'
```

COMPUTE IF: NatCen = NI

```
LAuths := 'Social Services'
```

COMPUTE IF: NatCen = NI

```
LAuth1 := 'Social Services'
```

COMPUTE IF: NatCen = NI

```
LAuth2 := 'Social Services'
```

COMPUTE IF: NatCen = NI

```
IncROI1 := '
Include accounts held in the Republic Of Ireland.'
```

COMPUTE IF: NatCen = NI

```
IncROI2 := ('If account held in the Republic of Ireland open
note ' + 'to state if amount recorded in Punts or Euros.')
```

COMPUTE IF: NatCen = NI

```
IncROI3 := ('Include holdings and/or accounts held in the
Republic ' + 'Of Ireland and open note to state if amount
recorded in ' + 'Punts or Euros.')
```

COMPUTE IF: NatCen = NI

```
Dept := 'Department for Social Development'
```

COMPUTE IF: NatCen = NI

```
Mid_Pri := ''
```

COMPUTE IF: NatCen = NI

```
Mid_Sec := ''
```

COMPUTE IF: NatCen = NI

```
Grammar := '/Grammar'
```

COMPUTE IF: NatCen = NI

```
State_run := ''
```

COMPUTE IF: NatCen = NI

assisted := ''

COMPUTE IF: NatCen = NI

Inland_Revenue := 'HM Revenue and Customs (formerly Inland Revenue)'

COMPUTE IF: NOT (NatCen = NI)

SharOwn := 'shared ownership'

COMPUTE IF: NOT (NatCen = NI)

SOwners := 'Shared owners'

COMPUTE IF: NOT (NatCen = NI)

LANIHE := 'local authority'

COMPUTE IF: NOT (NatCen = NI)

Council1 := ('The local authority/council/New Town ' + 'development/Scottish Homes')

COMPUTE IF: NOT (NatCen = NI)

Council2 := 'Local Authority or Council (incl. GLC)'

COMPUTE IF: NOT (NatCen = NI)

GOVSSA := 'DWP (formerly DSS)'

COMPUTE IF: NOT (NatCen = NI)

GOV1 := 'DWP'

COMPUTE IF: NOT (NatCen = NI)

GOV2 := 'DWP (formerly DSS)'

COMPUTE IF: NOT (NatCen = NI)

JobCen := 'an Employment Service local office or Jobcentre'

COMPUTE IF: NOT (NatCen = NI)

RentReb1 := 'rent rebate'

COMPUTE IF: NOT (NatCen = NI)

RentReb2 := 'rent rebate'

COMPUTE IF: NOT (NatCen = NI)

LAuths := 'Local Authorities'

COMPUTE IF: NOT (NatCen = NI)

LAuth1 := 'Local Authority'

COMPUTE IF: NOT (NatCen = NI)

LAuth2 := 'a Local Authority'

COMPUTE IF: NOT (NatCen = NI)

IncROI1 := ''

COMPUTE IF: NOT (NatCen = NI)

IncROI2 := ''

COMPUTE IF: NOT (NatCen = NI)

IncROI3 := ''

COMPUTE IF: NOT (NatCen = NI)

Dept := 'Department for Work and Pensions'

COMPUTE IF: NOT (NatCen = NI)

Mid_Pri := 'Middle-deemed primary school (state run or assisted)'

COMPUTE IF: NOT (NatCen = NI)

Mid_Sec := 'Middle-deemed secondary school (state run or assisted)'

COMPUTE IF: NOT (NatCen = NI)

Grammar := ''

COMPUTE IF: NOT (NatCen = NI)

State_run := 'State run'

COMPUTE IF: NOT (NatCen = NI)

assisted := '(State run or assisted)'

COMPUTE IF: NOT (NatCen = NI)

Inland_Revenue := 'HM Revenue and Customs (formerly Inland Revenue)(or formerly the DSS)'

ASK ALWAYS:

FRS0605B.First

^I^B^BLU^IC^B For your information... You are in the ^B Household ^B Schedule for
Area No:@|@|^StrArea
Address No:@|@|^StrAddr
Household No:@|^QSerial.Hhold

- To go directly to 'Admin', press <Ctrl + Enter>.
- To continue with interview press '1' and <Enter>.

Interview version@|:@|^IVers
^EditVersion^I

(1)	Continue	Continue

ASK IF: *NatCen = Yes*

FRS0605B.AdrCheck

^I^BLU Refer to address label:
Check that label gives respondent's full current address.

If not, amend address and code 'Address changed'.^BLU^I

(1)	Conf	Address confirmed
(2)	Chang	Address changed

COMPUTE IF: *(QDataBag.SampYear = RESPONSE) AND (QDataBag.SampMnth = RESPONSE)*

ThisYear := QDataBag.SampYear

COMPUTE IF: *(QDataBag.SampYear = RESPONSE) AND (QDataBag.SampMnth = RESPONSE)*

ThisMnth := QDataBag.SampMnth

ASK IF: *Test = Yes*

FRS0605B.ThisYear

^I Enter FRS Survey Year (eg. 2006 = April 2006 - March 2007).^I

2006..2020

ASK IF: *Test = Yes*

FRS0605B.ThisMnth

^I Enter survey month.^I

1..12

COMPUTE IF: *ThisMnth IN [1 .. 12]*

MnthOK2 := ThisMnth

COMPUTE IF: *ThisMnth IN [1 .. 12]*
AND: *ThisMnth = 11*

MnthOK1 := 10

COMPUTE IF: ThisMnth IN [1 .. 12]
AND: ThisMnth = 11

MnthOK3 := 12

COMPUTE IF: ThisMnth IN [1 .. 12]
AND: ThisMnth = 11

MnthOK4 := 1

COMPUTE IF: ThisMnth IN [1 .. 12]
AND: ThisMnth = 12

MnthOK1 := 11

COMPUTE IF: ThisMnth IN [1 .. 12]
AND: ThisMnth = 12

MnthOK3 := 1

COMPUTE IF: ThisMnth IN [1 .. 12]
AND: ThisMnth = 12

MnthOK4 := 2

COMPUTE IF: ThisMnth IN [1 .. 12]
AND: ThisMnth = 1

MnthOK1 := 12

COMPUTE IF: ThisMnth IN [1 .. 12]
AND: ThisMnth = 1

MnthOK3 := 2

COMPUTE IF: ThisMnth IN [1 .. 12]
AND: ThisMnth = 1

MnthOK4 := 3

COMPUTE IF: ThisMnth IN [1 .. 12]
AND: ThisMnth = 4

MnthOK1 := 4

COMPUTE IF: ThisMnth IN [1 .. 12]
AND: ThisMnth = 4

MnthOK3 := 5

COMPUTE IF: ThisMnth IN [1 .. 12]
AND: ThisMnth = 4

MnthOK4 := 6

COMPUTE IF: ThisMnth IN [1 .. 12]
AND: NOT (ThisMnth = 4)

MnthOK1 := (ThisMnth - 1)

COMPUTE IF: *ThisMnth IN [1 .. 12]*
    AND: *NOT (ThisMnth = 4)*

MnthOK3 := (ThisMnth + 1)

COMPUTE IF: *ThisMnth IN [1 .. 12]*
    AND: *NOT (ThisMnth = 4)*

MnthOK4 := (ThisMnth + 2)

FRS0605B.QSignIn

RECORD ALWAYS:

FRS0605B.QSignIn.StartDat

^I^BLU^IC The date on which the interview with this household was started.^I

DATE

RECORD ALWAYS:

FRS0605B.QSignIn.DateOK

^I^BLU^IC Today's date according to the laptop is ^AStartD.

Is this the correct date?^I

(1)	Yes	Yes
(2)	No	No

COMPUTE IF: *StartDat = EMPTY*

AStartD := SYSDATE

ASK IF: *StartDat = EMPTY*

FRS0605B.QSignIn.DateOK

^I^BLU^IC Today's date according to the laptop is ^AStartD.

Is this the correct date?^I

(1)	Yes	Yes
(2)	No	No

COMPUTE IF: *StartDat = EMPTY*
AND: *DateOK = Yes*

StartDat := AStartD

ASK IF: *StartDat = EMPTY*
AND: *DateOK = No*

FRS0605B.QSignIn.BStartD

^I^BLU^IC Enter the date on which the interview with this household was started.^I

DATE

COMPUTE IF: StartDat = EMPTY
AND: DateOK = No
AND: (((BStartD.YEAR = ThisYear) OR (BStartD.YEAR = (ThisYear + 1))) AND (((BStartD.MONTH = MnthOK1) OR (BStartD.MONTH = MnthOK2)) OR (BStartD.MONTH = MnthOK3)) OR (BStartD.MONTH = MnthOK4))) OR (ThisYear <> RESPONSE)) OR (ThisMnth <> RESPONSE)

StartDat := BStartD

CHECK IF: StartDat = EMPTY
AND: DateOK = No
AND: NOT ((((BStartD.YEAR = ThisYear) OR (BStartD.YEAR = (ThisYear + 1))) AND (((BStartD.MONTH = MnthOK1) OR (BStartD.MONTH = MnthOK2)) OR (BStartD.MONTH = MnthOK3)) OR (BStartD.MONTH = MnthOK4))) OR (ThisYear <> RESPONSE)) OR (ThisMnth <> RESPONSE)
((BStartD.YEAR = ThisYear) OR (BStartD.YEAR = (ThisYear + 1))) AND INVOLVING(BStartD)

^I The year is wrong for the current FRS survey (^ThisYear)!
Please re-enter the date from the beginning.^I

CHECK IF: StartDat = EMPTY
AND: DateOK = No
AND: NOT ((((BStartD.YEAR = ThisYear) OR (BStartD.YEAR = (ThisYear + 1))) AND (((BStartD.MONTH = MnthOK1) OR (BStartD.MONTH = MnthOK2)) OR (BStartD.MONTH = MnthOK3)) OR (BStartD.MONTH = MnthOK4))) OR (ThisYear <> RESPONSE)) OR (ThisMnth <> RESPONSE)
((((BStartD.MONTH = MnthOK1) OR (BStartD.MONTH = MnthOK2)) OR (BStartD.MONTH = MnthOK3)) OR (BStartD.MONTH = MnthOK4)) AND INVOLVING(BStartD)

^I The month is wrong for the current FRS survey (^Months[ThisMnth])!
Please re-enter the date from the beginning.^I

ASK IF: (Edit = Yes) AND (Test = Yes)

FRS0605B.QSignIn.CStartD

^I EDITOR: Change the interview start date (currently ^StartDat)?^I

(1)	Yes	Yes
(2)	No	No

ASK IF: (Edit = Yes) AND (Test = Yes)
AND: CStartD = Yes

FRS0605B.QSignIn.StartDat

^I^BLU^IC The date on which the interview with this household was started.^I

DATE

DISPLAY IF: NOT ((Edit = Yes) AND (Test = Yes)

FRS0605B.QSignIn.StartDat

^I^BLU^IC The date on which the interview with this household was started.^I

DATE

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

CHECK ALWAYS:
RESERVECHECK

RESERVECHECK

RECORD ALWAYS:

FRS0605B.QSignIn.IntSTime

^I Interview start time^I

TIME

COMPUTE IF: *IntSTime = EMPTY AND StartDat <> EMPTY*

IntSTime := STARTTIME

ASK IF: *(Edit = Yes) AND (NatCen <> Yes)*

FRS0605B.QSignIn.Editor

^I Editor at HQ: Enter your identification number.^I

1..97

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

COMPUTE ALWAYS:

```
DateNow := QSignIn.StartDat
```

COMPUTE ALWAYS:

```
DatYrAgo := (DateNow + (-1,0,0))
```

COMPUTE ALWAYS:

```
DatMnAgo := (DateNow + (0,-1,0))
```

COMPUTE ALWAYS:

```
DatWkAgo := (DateNow + (0,0,-7))
```

COMPUTE ALWAYS:

```
DatWeek := (Days[DatWkAgo.WEEKDAY] + ' the ' +
STR(DatWkAgo.DAY) + ' ' + Months[DatWkAgo.MONTH])
```

COMPUTE IF: DateNow.WEEKDAY = 1

```
DatLSun := (DateNow + (0,0,-7))
```

COMPUTE IF: NOT (DateNow.WEEKDAY = 1)

```
DatLSun := (DateNow + (0,0,-(DateNow.WEEKDAY) + 1))
```

COMPUTE ALWAYS:

```
DatSun := ('Sunday the ' + STR(DatLSun.DAY) + ' ' +
Months[DatLSun.MONTH])
```

CHECK ALWAYS:

RESERVECHECK

RESERVECHECK

CHECK ALWAYS:

RESERVECHECK

RESERVECHECK

CHECK ALWAYS:

RESERVECHECK

RESERVECHECK

CHECK ALWAYS:

RESERVECHECK

RESERVECHECK

COMPUTE IF: QDataBag.SampMnth IN [1, 2, 3]

CheckYear := 2007

COMPUTE IF: NOT (QDataBag.SampMnth IN [1, 2, 3])

CheckYear := 2006

COMPUTE ALWAYS:

FWDate := TODATE(CheckYear,QDataBag.SampMnth,1)

WARN IF: QDataBag.SampYear <> 0

```
     (QSignIn.StartDat.YEAR = FWDate.YEAR) OR ((QSignIn.StartDat.YEAR =
     (FWDate.YEAR + 1)) AND (QSignIn.StartDat <= (FWDate + (0,3,0))))
```

^I You have accidentally entered the wrong year. It doesn't agree with the fieldwork period. Please check and amend.^I

WARN IF: QDataBag.SampMnth <> 0

```
     (QSignIn.StartDat >= FWDate) OR ((FWDate.MONTH = 12) AND
     (QSignIn.StartDat.MONTH = 11))
```

^I You have accidentally entered the wrong month and/or year. It doesn't agree with the fieldwork period. Please check and amend.^I

WARN IF: QDataBag.SampMnth <> 0

```
     QSignIn.StartDat <= (FWDate + (0,3,0))
```

^I The month of this date is more than 3 months after the fieldwork period, please check and amend.^I

COMPUTE ALWAYS:

DLYear := (QSignIn.StartDat + (-1,0,0))

FRS0605B.QNames

Names of household members

Ask always:

FRS0605B.QNames.WhoHere

^N
Who normally lives at this address?^N

(1) Cont Press <Enter> to continue.

FRS0605B.QNames.M[]

Ask if: *In loop FOR Pers := 1 TO 14*
And: *(Pers = 1) OR (M[Pers - 1].More = Yes)*

FRS0605B.QNames.M[].Name

^BLU^IC^I Enter an identifier for this household member

It doesn't have to be a name - just something that uniquely identifies this person within the household so you can refer to them later in the interview.^I

STRING[15]

FRS0605B.QNames.M[].ProperAdd()

Procedure Call

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
      AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
      AND: Name = RESPONSE
```

OAddLine := AddLine

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
      AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
      AND: Name = RESPONSE
```

PNoChar := NoChar

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
      AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
      AND: Name = RESPONSE
```

NLetter := 0

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
      AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
      AND: Name = RESPONSE
```

NLettID := 0

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
      AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
      AND: Name = RESPONSE
      AND: OAddLine = RESPONSE
      AND: In loop FOR Idx := 1 TO PNoChar
```

AVar[Idx] := LOWERCASE(SUBSTRING(OAddLine,Idx,1))

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
      AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
      AND: Name = RESPONSE
      AND: OAddLine = RESPONSE
      AND: In loop FOR Idx := 1 TO PNoChar
```

ASCIICode[Idx] := SOMETHING(AVar[Idx])

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
      AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
      AND: Name = RESPONSE
      AND: OAddLine = RESPONSE
      AND: In loop FOR Idx := 1 TO PNoChar
      AND: ASCIICode[Idx] IN [48 .. 57, 65 .. 90, 97 .. 122]
```

NLetter := (NLetter + 1)

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
      AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
      AND: Name = RESPONSE
      AND: OAddLine = RESPONSE
      AND: In loop FOR Idx := 1 TO PNoChar
      AND: ASCIICode[Idx] IN [48 .. 57, 65 .. 90, 97 .. 122]
      AND: NLettID = 0
```

NLettID := Idx

```
CHECK IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
     AND: In loop FOR Idx := 1 TO PNoChar
     AND: NOT (ASCIICode[1] IN [32, 48 .. 57, 65 .. 90, 97 .. 122])
     ERROR AND INVOLVING(AddLine)
```

You started the text with an invalid character (^AVar[Idx]).

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
     AND: In loop FOR Idx := 1 TO PNoChar
     AND: ASCIICode[1] IN [97 .. 122]
```

ASCIICode[1] := (ASCIICode[1] - 32)

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
     AND: In loop FOR Idx := 1 TO PNoChar
     AND: ASCIICode[1] IN [97 .. 122]
```

AVar[1] := UPCASE(AVar[1])

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
     AND: In loop FOR Idx := 1 TO PNoChar
     AND: Idx > 1
     AND: (ASCIICode[Idx - 1] IN [32, 45, 46]) AND (ASCIICode[Idx] IN [65 ..
     90, 97 .. 122])
```

ASCIICode[Idx] := (ASCIICode[Idx] - 32)

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
     AND: In loop FOR Idx := 1 TO PNoChar
     AND: Idx > 1
     AND: (ASCIICode[Idx - 1] IN [32, 45, 46]) AND (ASCIICode[Idx] IN [65 ..
     90, 97 .. 122])
```

AVar[Idx] := UPCASE(AVar[Idx])

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
     AND: In loop FOR Idx := 1 TO PNoChar
     AND: Idx > 1
     AND: (ASCIICode[Idx - 2] = 77) AND (ASCIICode[Idx - 1] = 99)
```

ASCIICode[Idx] := (ASCIICode[Idx] - 32)

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
     AND: In loop FOR Idx := 1 TO PNoChar
     AND: Idx > 1
     AND: (ASCIICode[Idx - 2] = 77) AND (ASCIICode[Idx - 1] = 99)
```

AVar[Idx] := UPCASE(AVar[Idx])

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
     AND: In loop FOR Idx := 1 TO PNoChar
```

PAddLine := (PAddLine + AVar[Idx])

```
CHECK IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
     RESERVECHECK
```

RESERVECHECK

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
     AND: PAddLine = RESPONSE
     AND: In loop FOR Idx := 1 TO PNoChar
```

AVar2[Idx] := SUBSTRING(PAddLine,Idx,1)

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
     AND: PAddLine = RESPONSE
     AND: In loop FOR Idx := 1 TO PNoChar
```

ASCIICode2[Idx] := SOMETHING(AVar2[Idx])

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
     AND: PAddLine = RESPONSE
     AND: In loop FOR Idx := 1 TO PNoChar
     AND: (ASCIICode2[Idx] = 32) AND NOT (ASCIICode[Idx + 1] IN [0, 48 ..
     57, 65 .. 90, 97 .. 122])
     AND: In loop FOR Idy := 1 TO PNoChar
     AND: Idy >= Idx
```

ASCIICode2[Idx] := ASCIICode2[Idy + 1]

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
     AND: PAddLine = RESPONSE
     AND: In loop FOR Idx := 1 TO PNoChar
     AND: (ASCIICode2[Idx] = 32) AND NOT (ASCIICode[Idx + 1] IN [0, 48 ..
     57, 65 .. 90, 97 .. 122])
     AND: In loop FOR Idy := 1 TO PNoChar
     AND: Idy >= Idx
```

AVar2[Idx] := AVar2[Idy + 1]

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
     AND: PAddLine = RESPONSE
     AND: In loop FOR Idx := 1 TO PNoChar
     AND: (Idx < NLettID) AND (AVar2[Idx] = )
```

AVar2[Idx] := ''

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
     AND: PAddLine = RESPONSE
     AND: In loop FOR Idx := 1 TO PNoChar
```

PFAddLine := (PFAddLine + AVar2[Idx])

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: Name = RESPONSE
     AND: OAddLine = RESPONSE
```

EFAddLine := PFAddLine

FRS0605B.QNames.M[] (continued)

```
ASK IF: In loop FOR Pers := 1 TO 14
     AND: (Pers = 1) OR (M[Pers - 1].More = Yes)
     AND: PPers < 14
```

FRS0605B.QNames.M[].More

^N
Is there anyone else in this household?^N

(1)	Yes	Yes
(2)	No	No

FRS0605B.QNames (continued)

Names of household members

```
COMPUTE IF: In loop FOR Pers := 1 TO 14
     AND: M[Pers].More = No
```

```
HSize := Pers
```

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

RECORD ALWAYS:

FRS0605B.HHSize

^I Household size including any x-ed out^I

0..14

COMPUTE IF: (QNames.HSize > 0) OR (Edit = No)

```
HHSize := QNames.HSize
```

COMPUTE IF: In loop FOR Loop1 := 1 TO HHSize

```
DMName[Loop1] := QNames.M[Loop1].Name
```

CHECK ALWAYS:

RESERVECHECK

RESERVECHECK

CHECK ALWAYS:

RESERVECHECK

RESERVECHECK

CHECK ALWAYS:

RESERVECHECK

RESERVECHECK

CHECK ALWAYS:

RESERVECHECK

RESERVECHECK

FRS0605B.HHG

Data on household members

COMPUTE IF: HHSize > 0

FHHSize := PHHSize

COMPUTE IF: HHSize > 0
AND: NatCen = Yes

Out772 := '772'

COMPUTE IF: HHSize > 0
AND: NOT (NatCen = Yes)

Out772 := '77'

COMPUTE IF: HHSize > 0
AND: In loop FOR P1 := 1 TO FHHSize

P[P1].Person := P1

COMPUTE IF: HHSize > 0
AND: In loop FOR P1 := 1 TO FHHSize

P[P1].Name := QNames.M[].Name[P1]

FRS0605B.HHG.P[]

RECORD IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*

FRS0605B.HHG.P[].BenUnit

HHG
Benefit Unit number.

0..7

RECORD IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*

FRS0605B.HHG.P[].Person

HHG
Person number in Household Grid.

0..14

RECORD IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*

FRS0605B.HHG.P[].Name

HHG
^N First name.^N

STRING[15]

COMPUTE IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*

LName := Name

COMPUTE IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*

UName := UPCASE(Name)

ASK IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*

FRS0605B.HHG.P[].Sex

HHG

^I^BLU Code ^UName'S sex.^I

(1)	Male	Male
(2)	Female	Female

```
DISPLAY IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
```

FRS0605B.HHG.P[].Name

HHG
^N First name.^N

STRING[15]

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: Sex = Male
```

heshe := 'he'

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: Sex = Male
```

hisher := 'his'

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: NOT (Sex = Male)
```

heshe := 'she'

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: NOT (Sex = Male)
```

hisher := 'her'

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

FRS0605B.HHG.P[].AgeOf

HHG
^N
What was ^LName's age last birthday?^N

^I^BLU If age not given, probe for an estimate.
For later routing, you must know whether:

A) Men are aged 16-64 or 65+
B) Women are aged 16-59 or 60+
^I

0..120

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

DVAge := AgeOf

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: AgeOf IN [16 .. 120]
```

FRS0605B.HHG.P[].MS

HHG
^I
The aim is to obtain legal marital status, irrespective of any de facto arrangement. The only qualification to this aim is that you should not probe the answer 'separated'. Should a respondent query the term, explain that it covers any person whose spouse is living elsewhere because of estrangement (whether the separation is legal or not). Ignore temporary absences, eg. on oil rig.
A person whose spouse has been working away from home for over 6 months, eg. on a contract overseas or in the armed forces, should still be coded as married and living with husband/wife if the separation is not permanent.
Civil partners may only be of the same sex, and must have obtained legal recognition of their partnership; probe whether partnership was registered under the new provisions that came into force as from December 2005. ^I

(1) Single ^N... single, that is, never married or never in a legally recognised Civil Partnership,^N
(2) Marr ^N... married and living with husband/wife or in a legally recognised Civil Partnership and living with civil partner,^N
(3) Sep ^N...married and separated from husband/wife or in a legally recognised Civil Partnership and separated from civil partner,^N
(4) Divorce ^N...divorced or Civil Partnership legally dissolved,^N
(5) Widowed ^N...or widowed?^N

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: AgeOf IN [16 .. 120]
     AND: (FHHSize > 1) AND (MS IN [Single, Sep, Divorce, Widowed])
```

FRS0605B.HHG.P[].CupChk

HHG
^I Ask or record:^I
^N May I just check, are you / is ^LName living with someone in this household as a couple?

^BLU^I^IC Only respondents who are living with their partner should be coded as living together as a couple.
You may code No without asking the question^B only^B if all members of the household are too closely related for any to be living together in a de facto marital relationship.^I

(1)	Yes	Yes
(2)	No	No

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: MS = Widowed
```

FRS0605B.HHG.P[].W1

HHG
^N
What was ^LName's age when widowed?^N

0..120

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: MS = Widowed
     AND: W1 = RESPONSE
     W1 >= 16
```

^I Are you sure? It is not usual to be married before the age of 16 unless you were married outside the UK.^I

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: MS = Widowed
     AND: W1 = RESPONSE
     W1 <= AgeOf
```

^I You've coded that ^LName is ^AgeOf years old, but was widowed at the age of ^W1. Please amend the one or the other.^I

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: MS = Widowed
```

FRS0605B.HHG.P[].W2

HHG
^N
Did ^LName have any children aged under 16 when widowed?^N

(1)	Yes	Yes
(2)	No	No

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: Sex = Male
```

SonDaughter := 'son'

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: Sex = Male
```

BrotherSister := 'brother'

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: Sex = Male
```

FatherMother := 'father'

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: NOT (Sex = Male)
```

SonDaughter := 'daughter'

COMPUTE IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    AND: *SUBSTRING (Name, 1, 2) <> XX*
    AND: *NOT (Sex = Male)*

BrotherSister := 'sister'

COMPUTE IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    AND: *SUBSTRING (Name, 1, 2) <> XX*
    AND: *NOT (Sex = Male)*

FatherMother := 'mother'

CHECK IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    AND: *SUBSTRING (Name, 1, 2) <> XX*
    RESERVECHECK

RESERVECHECK

CHECK IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    AND: *SUBSTRING (Name, 1, 2) <> XX*
    RESERVECHECK

RESERVECHECK

CHECK IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    AND: *SUBSTRING (Name, 1, 2) <> XX*
    RESERVECHECK

RESERVECHECK

CHECK IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    AND: *SUBSTRING (Name, 1, 2) <> XX*
    RESERVECHECK

RESERVECHECK

CHECK IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    AND: *SUBSTRING (Name, 1, 2) <> XX*
    RESERVECHECK

RESERVECHECK

CHECK IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    AND: *SUBSTRING (Name, 1, 2) <> XX*
    RESERVECHECK

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     RESERVECHECK
```

RESERVECHECK

FRS0605B.HHG.P[].QRel[]

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     AND: RPers < PPers
```

FRS0605B.HHG.P[].QRel[].R

HHG

^I^BLU Ask or record ^PName's relationship to ^RName.^BLU^I

(1)	Spouse	spouse / civil partner,
(2)	Cohabit	cohabitee,
(3)	Child	^SonDaughter (incl. adopted) (/legal dependant),
(4)	StChild	step-^SonDaughter,
(5)	FChild	foster child,
(6)	ILChild	^SonDaughter-in-law,
(7)	Parent	^FatherMother (or guardian),
(8)	StParent	step-^FatherMother,
(9)	FParent	foster parent,
(10)	ILParent	^FatherMother-in-law,
(11)	Sib	^BrotherSister (incl. adopted),
(12)	StSib	step-^BrotherSister,
(13)	FSib	foster ^BrotherSister,
(14)	ILSib	^BrotherSister-in-law,
(15)	GChild	grand-^SonDaughter,
(16)	GParent	grand-^FatherMother,
(17)	OthRel	other relative,
(18)	NonRel	or other non-relative?
(97)	Self	

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     AND: RPers < PPers
     R <> Self
```

^I Code 97 is not valid for this question.^I

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     AND: NOT (RPers < PPers)
```

FRS0605B.HHG.P[].QRel[].R

HHG

^I^BLU Ask or record ^PName's relationship to ^RName.^BLU^I

(1)	Spouse	spouse / civil partner,
(2)	Cohabit	cohabitee,
(3)	Child	^SonDaughter (incl. adopted) (/legal dependant),
(4)	StChild	step-^SonDaughter,
(5)	FChild	foster child,
(6)	ILChild	^SonDaughter-in-law,
(7)	Parent	^FatherMother (or guardian),
(8)	StParent	step-^FatherMother,
(9)	FParent	foster parent,
(10)	ILParent	^FatherMother-in-law,
(11)	Sib	^BrotherSister (incl. adopted),
(12)	StSib	step-^BrotherSister,
(13)	FSib	foster ^BrotherSister,
(14)	ILSib	^BrotherSister-in-law,
(15)	GChild	grand-^SonDaughter,
(16)	GParent	grand-^FatherMother,
(17)	OthRel	other relative,
(18)	NonRel	or other non-relative?
(97)	Self	

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     AND: NOT (RPers < PPers)
     AND: RPers = PPers
```

R := Self

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     NOT(IN(R,[???,???,???]))
```

^I Warning: This code must only be used if the foster child is covered by a Local Authority Allowance. Please check that this is the case.^I

FRS0605B.HHG.P[] (continued)

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: In loop FOR R1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

FRS0605B.HHG.P[].Spouses

HHG

0..14

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

FRS0605B.HHG.P[].NumParn

HHG

0..14

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

FRS0605B.HHG.P[].NumPart

HHG

0..14

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

FRS0605B.HHG.P[].NumCohab

HHG

0..14

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

FRS0605B.HHG.P[].Parent1

HHG
^N Person number of parent 1^N

0..14

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

FRS0605B.HHG.P[].Parent2

HHG
^N Person number of parent 2^N

0..14

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

FRS0605B.HHG.P[].Hholder

HHG
^N Is this person coded at QHholder.Hhldr?^N

(1)	Yes	Yes
(2)	No	No

ASK IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*
AND: *SUBSTRING (Name, 1, 2) <> XX*
AND: *AgeOf IN [4 .. 74]*

FRS0605B.HHG.P[].FTEd

HHG
^N
Is ^LName currently in full-time education?^N

^I^BLU^IC Include correspondence courses and open learning as well as other forms of full-time courses.^I

(1)	Yes	Yes
(2)	No	No

WARN IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*
AND: *SUBSTRING (Name, 1, 2) <> XX*
RESERVECHECK

RESERVECHECK

WARN IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*
AND: *SUBSTRING (Name, 1, 2) <> XX*
RESERVECHECK

RESERVECHECK

WARN IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*
AND: *SUBSTRING (Name, 1, 2) <> XX*
RESERVECHECK

RESERVECHECK

COMPUTE IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*
AND: *SUBSTRING (Name, 1, 2) <> XX*
AND: *((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])*
AND: *AgeOf IN [19 .. 23]*

StillEduc := (' - or is ' + heshe + ' still in full-time education')

COMPUTE IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*
AND: *SUBSTRING (Name, 1, 2) <> XX*
AND: *((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])*
AND: *NOT (AgeOf IN [19 .. 23])*

StillEduc := ''

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: AgeOf IN [19 .. 25]
```

**continuous := (''Continuous' can include a break, if less than ' + '18 months.
Code '96' if still in continuous F/T Ed.')**

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: AgeOf IN [26 .. 120]
```

continuous := ('Enter age (or code '96' if still in ' + 'continuous full-time education).')

```
COMPUTE IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: NOT (AgeOf IN [26 .. 120])
```

continuous := ''

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
```

FRS0605B.HHG.P[].TEA

HHG

^I Include the following as part of 'continuous education':
- A ^B'gap year'^B between school and college/university, as long as there is/was some clear intention to continue education.
- A^B holiday job^B during a course, provided they intend to continue with the course.
- The working section of a^B sandwich course^B.
-^B National Service^B, if it occurred between school and college/university.^I

5..97

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: TEA IN [5 .. 95]
     TEA <= AgeOf
```

^I You've coded that ^LName is ^AgeOf years old, but left full-time education at the age of ^TEA. Please amend the one or the other.^I

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: (FTEd = No) AND (AgeOf IN [16 .. 18])
     TEA <> 96
```

^I At the previous question you say that ^LName is not in full-time education. Please amend your answers.^I

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
```

FRS0605B.HHG.P[].TEAEx

HHG
^I This is a 'Key Question': it is very important to get an answer if you possibly can. An ESTIMATE is preferable to Don't know (or refusal).

^SuppTxt^I

OPEN

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: Edit = No
     AND: TEA = NONRESPONSE
     ERROR
```

^I This is a 'Key Question': it is very important to get an answer if you possibly can. An ESTIMATE is preferable to Don't know (or refusal).^I

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: Edit = No
     AND: TEA = NONRESPONSE
```

FRS0605B.HHG.P[].TEAEx

HHG
^I This is a 'Key Question': it is very important to get an answer if you possibly can. An ESTIMATE is preferable to Don't know (or refusal).

^SuppTxt^I

OPEN

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: Edit = No
     AND: TEA = RESPONSE
     AND: AgeOf > 25
     TEA <> 96
```

^I This person is over 25, so is unlikely to still be in CONTINUOUS full-time education (ie. having been OUT of education for less than 18 months). Please check.^I

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: Edit = No
     AND: TEA = RESPONSE
     (TEA >= 14) OR (TEA = 97)
```

^I This value seems low.
Please check that it is correct.^I

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: ((AgeOf IN [16 .. 18]) AND (FTEd = No)) OR (AgeOf IN [19 .. 120])
     AND: Edit = No
     AND: TEA = RESPONSE
     AND: TEA < 96
     TEA < 28
```

^I This value seems high.
Please check that it is correct.^I

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (TEA = 96) OR (FTEd = Yes)
```

FRS0605B.HHG.P[].TypeEd

HHG
^I Secondary Schools include: Secondary Modern, Grammar, Comprehensive and Technical Schools.

Age ranges for Middle-deemed Primary:
8-12
9-12
9-13

Age ranges for Middle-deemed Secondary:
9-13
10-13
10-14.^I

(1)	Nursery	Nursery School/Nursery Class/Playgroup/Pre-school
(2)	Primry	^State_run Primary (including reception classes)
(3)	Special	Special School ^State_run (e.g. for children with disabilities and special educational needs)
(4)	MidPri	^Mid_Pri
(5)	MidSec	^Mid_Sec
(6)	Sec	Secondary^Grammar school ^assisted
(7)	Nonadv	Non-advanced further education/ 6th form/tertiary/further education college
(8)	Private	Any PRIVATE/Independent school (prep, primary, secondary, City Technology Colleges)
(9)	Univ	University/polytechnic/any other higher education
(10)	HomeSch	Home Schooling

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (TEA = 96) OR (FTEd = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (TEA = 96) OR (FTEd = Yes)
     AND: TypeEd IN [Sec .. Nonadv]
     AgeOf >= 8
```

^I This doesn't sound right in relation to ^LName's age:
Please check your entry.^I

```
WARN IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (TEA = 96) OR (FTEd = Yes)
     AND: TypeEd = Nursery
     IN(AgeOf,[2..12])
```

^I This doesn't sound right in relation to ^LName's age:
Please check your entry.^I

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (TEA = 96) OR (FTEd = Yes)
     AND: (TypeEd = Special) AND (AgeOf IN [16 .. 18])
```

FRS0605B.HHG.P[].SchChk

HHG

^I^BLU^IC Please check: Is^B Child Benefit^B still received for this person? (IF yes, this confirms they still belong to someone else's benefit unit).^I

(1)	YesCB	Yes, child benefit still received
(2)	No	No

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: NatCen = NI
     NOT(IN(TypeEd,[???,???]))
```

^I Interviewer, do not use this code at TypeEd.^I

```
ASK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (AgeOf IN [0 .. 19]) AND (AgeOf = RESPONSE)
```

FRS0605B.HHG.P[].DoB

HHG
^N
May I check, what is ^LName's date of birth?^N

^I^BLU If day not known, enter 15th.^I

DATE

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (AgeOf IN [0 .. 19]) AND (AgeOf = RESPONSE)
     AND: DoB = RESPONSE
     DoB <= QSignIn.StartDat
```

^I You've entered a future date!^I

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
     AND: (AgeOf IN [0 .. 19]) AND (AgeOf = RESPONSE)
     AND: DoB = RESPONSE
     AND: AgeOf IN [1 .. 19]
     AGE(DoB,QSignIn.StartDat) = AgeOf
```

^I This date doesn't agree with the age and the date of interview. Please check.^I

CHECK IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*
AND: *SUBSTRING (Name, 1, 2) <> XX*
AND: *(AgeOf IN [0 .. 19]) AND (AgeOf = RESPONSE)*
AND: *DoB = RESPONSE*
AND: *(AgeOf = 0) AND (AgeOf = RESPONSE)*
((QSignIn.StartDat.JULIAN - DoB.JULIAN) <= 365) AND
INVOLVING(AgeOf,DoB)

^I This date doesn't agree with the age and the date of interview. Please check.^I

ASK IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*
AND: *SUBSTRING (Name, 1, 2) <> XX*
AND: *((AgeOf IN [16 .. 19]) AND (FTEd = No)) AND (TEA < 96)*

FRS0605B.HHG.P[].Trainee

HHG
^N
Is ^LName currently on a government scheme for employment training?^N

(1)	Yes	Yes
(2)	No	No

RECORD IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*
AND: *SUBSTRING (Name, 1, 2) <> XX*

FRS0605B.HHG.P[].Depend

HHG
^N Status indicator of whether this adult is treated as dependent.^N

(1)	Adult	Independent adult
(2)	DepAd	16-19 years old AND in F/T education, OR 16-19 in government employment training
(3)	Child	0-15 years old

RECORD IF: *HHSize > 0*
AND: *In loop FOR P1 := 1 TO FHHSize*
AND: *SUBSTRING (Name, 1, 2) <> XX*

FRS0605B.HHG.P[].LiveWith

HHG
^N Cohabitee?^N

(1)	Yes	Yes
(2)	No	No

```
RECORD IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: SUBSTRING (Name, 1, 2) <> XX
```

FRS0605B.HHG.P[].DVMarDF

HHG
^N De facto marital status^N

(1)	Married	Married or in a legally recognised Civil Partnership
(2)	Cohab	Cohabiting (including same sex couples)
(3)	DFSingle	Single
(4)	DFWidow	Widowed (including surviving civil partner)
(5)	DFDivor	Divorced or civil partnership dissolved
(6)	DFSepar	Separated

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

CHECK IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    RESERVECHECK

RESERVECHECK

CHECK IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    RESERVECHECK

RESERVECHECK

CHECK IF: *HHSize > 0*
    AND: *In loop FOR P1 := 1 TO FHHSize*
    RESERVECHECK

RESERVECHECK

FRS0605B.HHG (continued)

Data on household members

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

LegGuard[[P1] := 2

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

P[P1].NumParn := 0

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

P[P1].NumPart := 0

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

P[P1].NumCohab := 0

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

P[P1].Spouses := 0

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

P[P1].Parent1 := 1

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

P[P1].Parent2 := 1

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
```

P[P1].LiveWith := No

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P2 > P1
     AND: P[P2].QRel[P1].R IN [Spouse .. Cohabit, Sib .. ILSib, OthRel,
     NonRel]
```

P[P1].QRel[P2].R := P[P2].QRel[P1].R

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P2 > P1
     AND: P[P2].QRel[P1].R IN [Child .. ILChild]
```

P[P1].QRel[P2].R := (ORD(P[P2].QRel[P1].R) + 4)

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P2 > P1
     AND: P[P2].QRel[P1].R IN [Parent .. ILParent]
```

P[P1].QRel[P2].R := (ORD(P[P2].QRel[P1].R) - 4)

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P2 > P1
     AND: P[P2].QRel[P1].R IN [GChild]
```

P[P1].QRel[P2].R := GParent

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P2 > P1
     AND: P[P2].QRel[P1].R IN [GParent]
```

P[P1].QRel[P2].R := GChild

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = Cohabit
```

P[P1].LiveWith := Yes

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = Cohabit
```

P[P1].DVMarDF := Cohab

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].MS = Marr
```

P[P1].DVMarDF := Married

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].CupChk = Yes
```

P[P1].DVMarDF := Cohab

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].AgeOf < 16
```

P[P1].DVMarDF := DFSingle

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].MS = Single) AND (P[P1].LiveWith <> Yes)
```

P[P1].DVMarDF := DFSingle

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].MS = Widowed
```

P[P1].DVMarDF := DFWidow

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].MS = Divorce
```

P[P1].DVMarDF := DFDivor

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].MS = Sep
```

P[P1].DVMarDF := DFSepar

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = Spouse
     P[P1].Sex <> P[P2].Sex AND INVOLVING(P[P1].Sex,P[P2].Sex)
```

^I Civil partners must have obtained legal recognition of their partnership. Please check whether partnership was registered under the new provisions that came into force as from December 2005.^I

```
CHECK IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = Spouse
     ((P[P2].MS = Marr) OR P[P2].MS=EMPTY) AND
     INVOLVING(P[P2].QRel[P1].R,P[P2].MS)
```

^I You've recorded ^P[P1].Name as the spouse / civil partner of ^P[P2].Name who is NOT 'Married & living with spouse / is NOT in a legally recognised Civil Partnership and living with civil partner'. Please amend one or the other.^I

```
CHECK IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = Cohabit
     (P[P2].MS <> Marr) AND INVOLVING(P[P2].QRel[P1].R)
```

^I You've recorded ^P[P1].Name as 'cohabiting' with ^P[P2].Name, who is 'MARRIED & living with^B spouse^B / in a legally recognised Civil Partnership and living with^B civil partner^B'. Please amend one or the other^I

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = Cohabit
     P[P1].Sex <> P[P2].Sex AND INVOLVING(P[P2].QRel[P1].R)
```

^I A cohabiting partner has been found in our data to usually be of the opposite sex. Are you sure this is a same sex cohabiting couple? ^I

```
CHECK IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P2].QRel[P1].R = Spouse
     ((P[P1].MS = Marr) OR P[P1].MS=EMPTY) AND INVOLVING(P[P2].QRel[P1].R)
```

^I You've recorded ^P[P2].Name as the spouse / civil partner of ^P[P1].Name, who is not coded as being 'Married & living with spouse / in a legally recognised Civil Partnership and living with civil partner'. Please amend one or the other^I

```
CHECK IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P2].QRel[P1].R = Cohabit
     (P[P1].MS <> Marr) AND INVOLVING(P[P2].QRel[P1].R)
```

^I You've recorded ^P[P2].Name as 'cohabiting' with ^P[P1].Name, who is coded as 'MARRIED & living with^B spouse^B / in a legally recognised Civil Partnership and living with^B civil partner^B'. Please amend one or the other^I

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Parent .. ILParent, GParent]
     (P[P1].AgeOf > 15) AND INVOLVING(P[P2].QRel[P1].R,P[P1].AgeOf)
```

^I You've coded ^P[P1].Name as a parent (inc. foster/in-law/step) or grandparent, but he/she is less than 16 years old. Please check ^P[P1].Name's age.^I

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Child .. ILChild, GChild]
     (P[P1].AgeOf < P[P2].AgeOf) AND INVOLVING(P[P2].QRel[P1].R)
```

^I Children (inc. foster/in-law/step) and grandchildren should normally be younger than their parents/grandparents/ step-parents. Please check the ages you have entered.^I

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Parent .. ILParent, GParent]
     (P[P1].AgeOf > P[P2].AgeOf) AND INVOLVING(P[P2].QRel[P1].R)
```

^I Parents (inc. foster/in-law/step) or grandparents, are normally older than their child/grandchild/step-child. Please check the ages and relationships you've entered.^I

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Child .. ILChild]
     (P[P1].AgeOf < (P[P2].AgeOf - 12)) AND INVOLVING(P[P2].QRel[P1].R)
```

^I Children (inc. foster/in-law/step) are normally at least 12 years younger than parents/step-parents. Please check the ages you have entered.^I

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Parent .. ILParent]
     (P[P1].AgeOf > (P[P2].AgeOf + 12)) AND INVOLVING(P[P2].QRel[P1].R)
```

^I Parents (inc. foster/in-law/step) are normally at least 12 years older than their child/step-child. Please check the ages and relationships you've entered.^I

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = GChild
     (P[P1].AgeOf < (P[P2].AgeOf - 24)) AND INVOLVING(P[P2].QRel[P1].R)
```

^I Children are normally at least 24 years younger than their grandparents. Please check the ages you have entered.^I

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = GParent
     (P[P1].AgeOf > (P[P2].AgeOf + 24)) AND INVOLVING(P[P2].QRel[P1].R)
```

^I Grandparents are normally at least 24 years older than their grandchildren. Please check the ages and relationships you've entered.^I

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Spouse]
```

P[P1].Spouses := (P[P1].Spouses + 1)

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Spouse .. Cohabit]
```

P[P1].NumPart := (P[P1].NumPart + 1)

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R = Cohabit
```

P[P1].NumCohab := (P[P1].NumCohab + 1)

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].AgeOf IN [0 .. 15]) AND (P[P1].AgeOf = RESPONSE)
```

P[P1].Depend := Child

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].NumPart > 0
```

P[P1].Depend := Adult

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].AgeOf IN [16 .. 18]) AND ((P[P1].TypeEd = Special) AND
     (P[P1].SchChk = No))
```

P[P1].Depend := Adult

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].AgeOf IN [16 .. 19]) AND ((P[P1].TypeEd IN [Special ..
     Private]) OR (P[P1].Trainee = Yes))
```

P[P1].Depend := DepAd

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: NOT ((P[P1].AgeOf IN [16 .. 19]) AND ((P[P1].TypeEd IN [Special ..
     Private]) OR (P[P1].Trainee = Yes))
```

P[P1].Depend := Adult

```
CHECK IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     (IN(P[P1].NumPart,[0..1])) AND
     INVOLVING(P[P2].QRel[P1].R,P[P1].QRel[P2].R)
```

^I^P[P1].Name has more than one spouse/cohabitee. Establish who is principal partner, & re-code the other as '17' or '18'.^I

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Child .. FChild]
     AND: P[P1].Parent1 = EMPTY
```

P[P1].Parent1 := P2

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Child .. FChild]
     AND: P[P1].Parent2 = EMPTY AND (P[P1].Parent1 <> P2)
     P[P[P1].Parent1].Sex <> P[P2].Sex AND INVOLVING(P[P2].QRel[P1].R)
```

^I The parents of ^P[P1].Name are of the same sex. Please check.^I

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Child .. FChild]
     AND: P[P1].Parent2 = EMPTY AND (P[P1].Parent1 <> P2)
```

P[P1].Parent2 := P2

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: P[P1].QRel[P2].R IN [Child .. FChild]
```

P[P1].NumParn := (P[P1].NumParn + 1)

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].Depend = Adult) AND INVOLVING (P[P2].QRel[P1].R,
     P[P1].QRel[P2].R)
     P[P1].NumParn <= 2
```

^I This suggests that ^P[P1].Name has more than two parents. Please check the relationship codes for ^P[P1].Name and select which one to alter.^I

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].Depend = DepAd) AND INVOLVING (P[P2].QRel[P1].R,
     P[P1].QRel[P2].R)
     P[P1].NumParn <= 2
```

^I You've given ^P[P1].Name more than two parents (inc step/foster). To calculate Benefit Units properly you must reduce this to a maximum of two. Select which one to alter, and re-code as '17'. (Check who receives Child Benefit for ^P[P1].Name).^I

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].Depend = Child) AND (P[P1].QRel[P2].R IN [Child .. FChild])
```

LegGuard[[P1] := 1

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: In loop FOR P2 := 1 TO FHHSize
     AND: (P[P1].Depend = DepAd) AND (P[P1].QRel[P2].R IN [Spouse ..
     FChild])
```

LegGuard[[P1] := 1

```
WARN IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: (P[P1].Depend = DepAd) AND (P[PHHSize].QRel[PHHSize - 1].R =
     RESPONSE)
     (LegGuard[[P1] = 1) AND INVOLVING(P[P1].QRel[1].R)
```

^I Who in the household is responsible for ^P[P1].Name - is there a legal guardian, or does anyone get Child Benefit for ^P[P1].Name? If so, recode ^P[P1].Name as that person's legal dependent (Code 3) or that person as ^P[P1].Name's parent (code 7). If not, suppress warning and continue.^I

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: P[P1].Sex = Male
```

himher := 'him'

```
COMPUTE IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: P[P1].Sex = Female
```

himher := 'her'

```
CHECK IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     AND: (P[P1].Depend = Child) AND (P[PHHSize].QRel[PHHSize - 1].R =
     RESPONSE)
     (LegGuard[[P1] = 1) AND INVOLVING(P[P1].AgeOf)
```

^I^P[P1].Name is under 16, so you MUST recode ^himher as the child (relationship code 3) of an adult: in order of priority, the person receiving Child Benefit for ^himher, or the legal guardian, or whoever is responsible for ^himher.

(NB. If no-one in the household is over age 15, the household is ineligible - outcome code ^Out772).^I

```
CHECK IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     AND: P[FHHSize].Sex = RESPONSE
     AND: In loop FOR P1 := 1 TO FHHSize
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     RESERVECHECK
```

RESERVECHECK

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

```
CHECK IF: HHSize > 0
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHSize > 0
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: (HHG.P[Loop1].MS = Marr) AND (HHG.P[Loop1].Spouses = 0)
     AND: SpOut[Loop1] = Amend
     HHG.P[Loop1].Sex <> RESPONSE
```

^I Press <Enter> to return to the household grid.^I

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: (HHG.P[Loop1].MS = Marr) AND (HHG.P[Loop1].Spouses = 0)
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: (HHG.P[Loop1].MS = Marr) AND (HHG.P[Loop1].Spouses = 0)
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: (HHG.P[Loop1].MS = Marr) AND (HHG.P[Loop1].Spouses = 0)
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: (HHG.P[Loop1].MS = Marr) AND (HHG.P[Loop1].Spouses = 0)
     RESERVECHECK
```

RESERVECHECK

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
```

AllNameNo := ''

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
```

AdNameNo := ''

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
```

AllAd := 0

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
```

AllCh := 0

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
```

ChUnder1 := 0

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
```

ChRegis := 0

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
```

Over75 := 0

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
```

Over60 := 0

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
```

Under19 := 0

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

```
AllNameNo := (AllNameNo + '
' + STR(Loop1,2,0) + '. ' + QNames.M[Loop1].Name)
```

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: HHG.P[Loop1].AgeOf IN [16 .. 120]
```

```
AdNameNo := (AdNameNo + '
' + STR(Loop1,2,0) + '. ' + QNames.M[Loop1].Name)
```

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: HHG.P[Loop1].AgeOf IN [16 .. 120]
```

OneHRP := Loop1

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: HHG.P[Loop1].AgeOf IN [16 .. 120]
```

AllAd := (AllAd + 1)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: NOT (HHG.P[Loop1].AgeOf IN [16 .. 120])
```

AllCh := (AllCh + 1)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: HHG.P[Loop1].AgeOf IN [0 .. 1]
```

ChUnder1 := (ChUnder1 + 1)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: HHG.P[Loop1].AgeOf >= 75
```

Over75 := (Over75 + 1)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: HHG.P[Loop1].AgeOf >= 60
```

Over60 := (Over60 + 1)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: HHG.P[Loop1].AgeOf IN [0 .. 18]
```

Under19 := (Under19 + 1)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

DMAge[Loop1] := HHG.P[Loop1].AgeOf

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

DMDoB[Loop1] := HHG.P[Loop1].DoB

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

DMTEA[Loop1] := HHG.P[Loop1].TEA

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

DMParent1[Loop1] := HHG.P[Loop1].Parent1

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

DMParent2[Loop1] := HHG.P[Loop1].Parent2

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

DMNumParn[Loop1] := HHG.P[Loop1].NumParn

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

PRec[Loop1].Sex := HHG.P[Loop1].Sex

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

PRec[Loop1].MS := HHG.P[Loop1].MS

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

PRec[Loop1].W1 := HHG.P[Loop1].W1

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

PRec[Loop1].W2 := HHG.P[Loop1].W2

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

PRec[Loop1].FtEd := HHG.P[Loop1].FTEd

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
```

PRec[Loop1].TypeEd := HHG.P[Loop1].TypeEd

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: HHG.P[Loop1].Depend IN [DepAd .. Child]
```

NCDVCP := (NCDVCP + 1)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: (HHG.P[Loop1].Depend IN [DepAd .. Child]) AND
     ((HHG.P[Loop1].NumPart > 0) OR (LegGuard[Loop1] = 2))
```

PRec[Loop1].Depend := Adult

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: NOT ((HHG.P[Loop1].Depend IN [DepAd .. Child]) AND
     ((HHG.P[Loop1].NumPart > 0) OR (LegGuard[Loop1] = 2))
```

PRec[Loop1].Depend := HHG.P[Loop1].Depend

```
WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: (HHG.P[Loop1].CupChk = Yes) AND HHG.P[HHSize].QRel[HHSize].R <>
     EMPTY
     (HHG.P[Loop1].NumCohab > 0) AND
     INVOLVING(HHG.P[HHSize].QRel[HHSize].R,HHG.P[Loop1].CupChk)
```

^I (^HHG.P[Loop1].Name) Can I just check, you said earlier that you were living with someone in this household as a couple, is this correct?

If yes: go back and correct the code at Relationship.
IF no: go back and correct CupChk to No (2).^I

```
WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].Sex = RESPONSE
     AND: (HHG.P[Loop1].CupChk = No) AND HHG.P[HHSize].QRel[HHSize].R <>
     EMPTY
     (HHG.P[Loop1].NumCohab = 0) AND
     INVOLVING(HHG.P[HHSize].QRel[HHSize].R,HHG.P[Loop1].CupChk)
```

^I(^HHG.P[Loop1].Name) Can I just check, are you living with anyone in the household as a couple?

If yes: go back and correct CupChk to Yes (1).
If no: go back and correct the code at Relationship.^I

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: AllAd = 1
```

you := 'you'

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: NOT (AllAd = 1)
```

you := 'ANY of you'

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (AllAd + AllCh) = 1
```

any_of_you := 'you'

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: NOT ((AllAd + AllCh) = 1)
```

any_of_you := 'ANY of you'

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
```

RentName := ''

FRS0605B.QHholder

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[1] := ' 1. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[2] := ' 2. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[3] := ' 3. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[4] := ' 4. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[5] := ' 5. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[6] := ' 6. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[7] := ' 7. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[8] := ' 8. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[9] := ' 9. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[10] := '10. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[11] := '11. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[12] := '12. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[13] := '13. '

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE

LPad[14] := '14. '

ASK IF: *HHG.P[HHSize].AgeOf = RESPONSE*

FRS0605B.QHholder.HHldr

^N
In whose name is the accommodation owned or rented?
^Anyone_else?^N

^I^BLU Code all that apply.^I

SET [15] OF

(1)	Per1	^DMName[1]
(2)	Per2	^DMName[2]
(3)	Per3	^DMName[3]
(4)	Per4	^DMName[4]
(5)	Per5	^DMName[5]
(6)	Per6	^DMName[6]
(7)	Per7	^DMName[7]
(8)	Per8	^DMName[8]
(9)	Per9	^DMName[9]
(10)	Per10	^DMName[10]
(11)	Per11	^DMName[11]
(12)	Per12	^DMName[12]
(13)	Per13	^DMName[13]
(14)	Per14	^DMName[14]
(97)	NotHH	Not a household member

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*

HhldList := ''

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*

HhldNum := 0

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    AND: *NotHH IN HHldr*

HhldCard := (HHldr.CARDINAL - 1)

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    AND: *NOT (NotHH IN HHldr)*

HhldCard := HHldr.CARDINAL

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
    AND: *In loop FOR Loop := 1 TO 14*

HhldName[Loop] := ''

FRS0605B.QHholder.PadString()

Procedure Call

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN HHldr
```

OutString := InString

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN HHldr
     AND: ResLngth > LEN (OutString)
```

Fin := (ResLngth - LENGTH(OutString))

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN HHldr
     AND: ResLngth > LEN (OutString)
     AND: In loop FOR PLoop := 1 TO Fin
```

OutString := (OutString + ' ')

FRS0605B.QHholder (continued)

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  AND: *In loop FOR Loop := 1 TO 14*
  AND: *Loop IN HHldr*

**HhldName[Loop] := ('
' + LPad[Loop] + PadName + ' (Age: ' + STR(DMAge[[Loop],3,0) + ')')**

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  AND: *In loop FOR Loop := 1 TO 14*
  AND: *Loop IN HHldr*

HhldNum := (HhldNum + 1)

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  AND: *In loop FOR Loop := 1 TO 14*
  AND: *Loop IN HHldr*
  AND: *HhldNum = 1*

HhldList := DMName[[Loop]

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  AND: *In loop FOR Loop := 1 TO 14*
  AND: *Loop IN HHldr*
  AND: *HhldNum < HhldCard*

HhldList := (HhldList + ', ' + DMName[[Loop])

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  AND: *In loop FOR Loop := 1 TO 14*
  AND: *Loop IN HHldr*
  AND: *HhldNum = HhldCard*

HhldList := (HhldList + ' and ' + DMName[[Loop])

WARN IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  RESERVECHECK

RESERVECHECK

WARN IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  RESERVECHECK

RESERVECHECK

WARN IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  RESERVECHECK

RESERVECHECK

WARN IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  RESERVECHECK

RESERVECHECK

WARN IF: *HHG.P[HHSize].AgeOf = RESPONSE*
RESERVECHECK

RESERVECHECK

ASK IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*

FRS0605B.QHholder.WhoResp

^N
Although you have mentioned that the rent or mortgage for this accommodation is paid for by someone outside the household, there needs to be someone within the household who is responsible for the property. Who then in this household is responsible for this accommodation?
Anyone else?^N

^I^BLU Code all that apply.

SET [14] OF

(1)	Per1	^DMName[1]
(2)	Per2	^DMName[2]
(3)	Per3	^DMName[3]
(4)	Per4	^DMName[4]
(5)	Per5	^DMName[5]
(6)	Per6	^DMName[6]
(7)	Per7	^DMName[7]
(8)	Per8	^DMName[8]
(9)	Per9	^DMName[9]
(10)	Per10	^DMName[10]
(11)	Per11	^DMName[11]
(12)	Per12	^DMName[12]
(13)	Per13	^DMName[13]
(14)	Per14	^DMName[14]

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*

HhldList := ''

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*

HhldNum := 0

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*

HhldCard := WhoResp.CARDINAL

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
AND: *(HHldr.CARDINAL = 1) AND (NotHH IN HHldr)*
AND: *In loop FOR Loop := 1 TO 14*

HhldName[Loop] := ''

FRS0605B.QHholder.PadString()

Procedure Call

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (HHldr.CARDINAL = 1) AND (NotHH IN HHldr)
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN WhoResp
```

OutString := InString

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (HHldr.CARDINAL = 1) AND (NotHH IN HHldr)
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN WhoResp
     AND: ResLngth > LEN (OutString)
```

Fin := (ResLngth - LENGTH(OutString))

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (HHldr.CARDINAL = 1) AND (NotHH IN HHldr)
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN WhoResp
     AND: ResLngth > LEN (OutString)
     AND: In loop FOR PLoop := 1 TO Fin
```

OutString := (OutString + ' ')

FRS0605B.QHholder (continued)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (HHldr.CARDINAL = 1) AND (NotHH IN HHldr)
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN WhoResp
```

HhldName[Loop] := ('
' + LPad[Loop] + PadName + ' (Age: ' + STR(DMAge[[Loop],3,0) + ')')

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (HHldr.CARDINAL = 1) AND (NotHH IN HHldr)
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN WhoResp
```

HhldNum := (HhldNum + 1)

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (HHldr.CARDINAL = 1) AND (NotHH IN HHldr)
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN WhoResp
     AND: HhldNum = 1
```

HhldList := DMName[[Loop]

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (HHldr.CARDINAL = 1) AND (NotHH IN HHldr)
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN WhoResp
     AND: HhldNum < HhldCard
```

HhldList := (HhldList + ', ' + DMName[[Loop])

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (HHldr.CARDINAL = 1) AND (NotHH IN HHldr)
     AND: In loop FOR Loop := 1 TO 14
     AND: Loop IN WhoResp
     AND: HhldNum = HhldCard
```

HhldList := (HhldList + ' and ' + DMName[[Loop])

```
WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (HHldr.CARDINAL = 1) AND (NotHH IN HHldr)
     RESERVECHECK
```

RESERVECHECK

```
RECORD IF: HHG.P[HHSize].AgeOf = RESPONSE
```

FRS0605B.QHholder.HRPPrtnr

^N Person number of HRP's spouse/partner.^N

1..15

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
AND: (HHldr <> EMPTY AND (AllAd = 1)) AND (OneHRP = RESPONSE)

HRPPrtnr := 15

WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
RESERVECHECK

RESERVECHECK

WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
RESERVECHECK

RESERVECHECK

WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
RESERVECHECK

RESERVECHECK

WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
RESERVECHECK

RESERVECHECK

WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
RESERVECHECK

RESERVECHECK

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
AND: WhoResp.CARDINAL > 1

own := 'are responsible for'

COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
AND: NOT (WhoResp.CARDINAL > 1)

own := 'own or rent'

ASK IF: HHG.P[HHSize].AgeOf = RESPONSE
AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)

FRS0605B.QHholder.HiHNum

^I If respondent asks for period to average over - one year.

Prompt as necessary for joint householders:
Is one of them the sole person with paid work or occupational pension?

1..15

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: WhoResp = EMPTY
     AND: HiHNum IN [1 .. 14]
     (IN(HiHNum,HHldr)) AND INVOLVING(HiHNum)
```

^I This person is not recorded as a householder (at HHldr).^I

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: NOT (WhoResp = EMPTY)
     AND: HiHNum IN [1 .. 14]
     (IN(HiHNum,WhoResp)) AND INVOLVING(HiHNum)
```

^I This person is not recorded as responsible for the household (at WhoResp).^I

```
ASK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = 15
```

FRS0605B.QHholder.JntEldA

^I Ask or record.
Enter Person Number of the^B eldest^B joint householder from those with the same highest income.

^HhldName[1]^HhldName[2]^HhldName[3]^HhldName[4]^HhldName[5]
^HhldName[6]^HhldName[7]^HhldName[8]^HhldName[9]^HhldName[10]
^HhldName[11]^HhldName[12]^HhldName[13]^HhldName[14]^I

0..14

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = 15
     AND: JntEldA = RESPONSE
     JntEldA <> 0
```

Zero (0) is not a valid code.

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = 15
     AND: JntEldA IN [1 .. 14]
     AND: WhoResp = EMPTY
     (IN(JntEldA,HHldr)) AND INVOLVING(HiHNum)
```

^I This person is not recorded as a householder (at HHldr).^I

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = 15
     AND: JntEldA IN [1 .. 14]
     AND: NOT (WhoResp = EMPTY)
     (IN(JntEldA,WhoResp)) AND INVOLVING(HiHNum)
```

^I This person is not recorded as responsible for the household (at WhoResp).^I

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = 15
     AND: JntEldA IN [1 .. 14]
```

DVHRPNum := JntEldA

```
ASK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = NONRESPONSE
```

FRS0605B.QHholder.JntEldB

^I Ask or record.
Enter Person Number of the eldest joint householder.

^HhldName[1]^HhldName[2]^HhldName[3]^HhldName[4]^HhldName[5]
^HhldName[6]^HhldName[7]^HhldName[8]^HhldName[9]^HhldName[10]
^HhldName[11]^HhldName[12]^HhldName[13]^HhldName[14]^I

0..14

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = NONRESPONSE
     AND: JntEldB = RESPONSE
     JntEldB <> 0
```

^I Zero (0) is not a valid code.^I

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = NONRESPONSE
     AND: JntEldB IN [1 .. 14]
     AND: WhoResp = EMPTY
     (IN(JntEldB,HHldr)) AND INVOLVING(HiHNum)
```

^I This person is not recorded as a householder (at HHldr).^I

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = NONRESPONSE
     AND: JntEldB IN [1 .. 14]
     AND: NOT (WhoResp = EMPTY)
     (IN(JntEldB,WhoResp)) AND INVOLVING(HiHNum)
```

^I This person is not recorded as responsible for the household (at WhoResp).^I

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: (((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR
     (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)
     AND: HiHNum = NONRESPONSE
     AND: JntEldB IN [1 .. 14]
```

DVHRPNum := JntEldB

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  AND: *(((HHldr.CARDINAL = 2) AND NOT (NotHH IN HHldr)) OR (HHldr.CARDINAL > 2)) OR (WhoResp.CARDINAL > 1)*
  AND: *HiHNum IN [1 .. 14]*

DVHRPNum := HiHNum

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  AND: *(HHldr.CARDINAL = 1) AND NOT (NotHH IN HHldr)*

DVHRPNum := ORD(HHldr[1])

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  AND: *(HHldr.CARDINAL = 2) AND (NotHH IN HHldr)*
  AND: *HHldr[1] = NotHH*

DVHRPNum := ORD(HHldr[2])

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  AND: *(HHldr.CARDINAL = 2) AND (NotHH IN HHldr)*
  AND: *NOT (HHldr[1] = NotHH)*

DVHRPNum := ORD(HHldr[1])

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  AND: *WhoResp.CARDINAL = 1*

DVHRPNum := ORD(WhoResp[1])

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  AND: *NOT (WhoResp.CARDINAL = 1)*

DVHRPNum := 0

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  AND: *DVHRPNum IN [1 .. 14]*

LName := DMName[[DVHRPNum]

COMPUTE IF: *HHG.P[HHSize].AgeOf = RESPONSE*
  AND: *NOT (DVHRPNum IN [1 .. 14])*

LName := 'Non-HH Member'

ASK IF: *HHG.P[HHSize].AgeOf = RESPONSE*

FRS0605B.QHholder.HRP

^I^BLU
The Household Reference Person is:

(^DVHRPNum) ^LName

Press 1 and <Enter> to continue.^I

(1) Continue Continue

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     AND: Loop1 IN QHholder.HHldr
     PRec[Loop1].Sex = RESPONSE
```

^I Code ^Loop1 is not valid for this question.^I

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     AND: Loop1 IN QHholder.HHldr
     PRec[Loop1].Depend = Adult
```

^I Person ^Loop1 is a child or a dependent adult. Please amend.^I

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     AND: Loop1 IN QHholder.HHldr
```

HHG.P[Loop1].Hholder := Yes

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     AND: NOT (Loop1 IN QHholder.HHldr)
```

HHG.P[Loop1].Hholder := No

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     AND: Loop1 IN QHholder.WhoResp
     PRec[Loop1].Sex = RESPONSE
```

^I This figure exceeds the number of household members. Please check and amend your answer.^I

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     AND: Loop1 IN QHholder.WhoResp
     PRec[Loop1].Depend = Adult
```

^I The person responsible for the property must be an adult household member. Please check and amend your answer.^I

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: In loop FOR Loop1 := 1 TO 14
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     PRec[QHholder.DVHRPNum].Depend = Adult
```

^I Code ^QHholder.DVHRPNum is not valid for this question.^I

```
WARN IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     NOT((PRec[QHholder.DVHRPNum].Sex = Female) AND
     (PRec[QHholder.DVHRPNum].MS = Marr))
```

^I For a married couple the man is always Head of household.
Please amend your coding. (But if he is away for more than 6 months, suppress check and move on.)^I

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
```

HRPNames := DMName[QHholder.DVHRPNum]

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
```

PRel.PR[Loop1].R := HHG.P[Loop1].QRel[QHholder.DVHRPNum].R

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].QRel[QHholder.DVHRPNum].R IN [Spouse, Cohabit]
```

HRPNames := (HRPNames + ' and ' + DMName[Loop1])

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: HHG.P[Loop1].QRel[QHholder.DVHRPNum].R IN [Spouse, Cohabit]
```

QHholder.HRPPrtnr := Loop1

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: DMAge[Loop1] >= 16
     AND: Loop1 = QHholder.DVHRPNum
```

QHholder.QPerId[Loop1].HOHID := HOH

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: DMAge[Loop1] >= 16
     AND: Loop1 = QHholder.DVHRPNum
```

QHholder.QPerId[Loop1].HRPID := HRP

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: DMAge[Loop1] >= 16
     AND: Loop1 = QHholder.DVHRPNum
```

QHholder.QPerId[Loop1].CombID := HOHHRP

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: DMAge[Loop1] >= 16
     AND: NOT (Loop1 = QHholder.DVHRPNum)
```

QHholder.QPerId[Loop1].HOHID := NotHOH

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: DMAge[Loop1] >= 16
     AND: NOT (Loop1 = QHholder.DVHRPNum)
```

QHholder.QPerId[Loop1].HRPID := NotHRP

```
COMPUTE IF: HHG.P[HHSize].AgeOf = RESPONSE
     AND: QHholder.DVHRPNum = RESPONSE
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: DMAge[Loop1] >= 16
     AND: NOT (Loop1 = QHholder.DVHRPNum)
```

QHholder.QPerId[Loop1].CombID := HOHonly

FRS0605B.QEthnic

Ethnic data on adults in household

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
    AND: *In loop FOR Personnr := 1 TO HHSize*
    AND: *PRec[Personnr].Depend = Adult*

P[Personnr].PersId := Personnr

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
    AND: *In loop FOR Personnr := 1 TO HHSize*
    AND: *PRec[Personnr].Depend = Adult*

P[Personnr].BenUnit := DMBU[[Personnr]

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
    AND: *In loop FOR Personnr := 1 TO HHSize*
    AND: *PRec[Personnr].Depend = Adult*

P[Personnr].EName := DMName[[Personnr]

FRS0605B.QEthnic.P[]

RECORD IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
 AND: *In loop FOR Personnr := 1 TO HHSize*
 AND: *PRec[Personnr].Depend = Adult*

FRS0605B.QEthnic.P[].BenUnit

^I QEthnic
^I

^N Benefit Unit number.^N

1..7

RECORD IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
 AND: *In loop FOR Personnr := 1 TO HHSize*
 AND: *PRec[Personnr].Depend = Adult*

FRS0605B.QEthnic.P[].PersId

^I QEthnic
^I

^N Person identifier.^N

0..14

DISPLAY IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
 AND: *In loop FOR Personnr := 1 TO HHSize*
 AND: *PRec[Personnr].Depend = Adult*

FRS0605B.QEthnic.P[].EName

^I QEthnic
^I

STRING[15]

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
 AND: *In loop FOR Personnr := 1 TO HHSize*
 AND: *PRec[Personnr].Depend = Adult*

LName := EName

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
 AND: *In loop FOR Personnr := 1 TO HHSize*
 AND: *PRec[Personnr].Depend = Adult*
 AND: *PRec[PersId].Sex = Male*

he_she := 'he'

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: NOT (PRec[PersId].Sex = Male)
```

he_she := 'she'

```
ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: NatCen <> NI
```

FRS0605B.QEthnic.P[].NatID

^I QEthnic
^I

^IS^I B^I
^X(^LName)^N What do you consider your^B national identity^B to be? Please choose your answer from this card, choose as many or as few as apply.^N
^I^BLU Probe:^Blu ^Any_other?

^I

SET [6] OF

(1)	English	English
(2)	Scottish	Scottish
(3)	Welsh	Welsh
(4)	Irish	Irish
(5)	British	British
(6)	Other	Other answer

```
ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: NatCen <> NI
     AND: Other IN NatID
```

FRS0605B.QEthnic.P[].NatOth

^I QEthnic
^I

^N (^LName)
How would you describe your national identity?^N

^I^BLU^IC If someone describes themselves as being half English and half Irish or any combination of Welsh, Scottish, Irish or English, code them as 'Mixed British' and then record the mix they specify.^I

(1)	Mixed	Mixed British
(2)	Describe	ENTER DESCRIPTION OF ETHNIC GROUP

```
ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: NatCen <> NI
     AND: Other IN NatID
```

FRS0605B.QEthnic.P[].XNatOth

^I QEthnic
^I

^I^BLU (^LName)
Enter description of ethnic group.^I

STRING[100]

```
ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: NatCen <> NI
```

FRS0605B.QEthnic.P[].EthGrp

^I QEthnic
^I

^IS^I C^I
^X^N To which of these ethnic groups does ^LName consider ^he_she belongs?^N
^I^BLU^IC This is a question of respondent's (or proxy's) opinion.^I

(1)	WhtBrit	White - British
(2)	WhtOth	Any other white background (please describe)
(3)	MixedWBC	Mixed - White and Black Caribbean
(4)	MixedWBA	Mixed - White and Black African
(5)	MixedWAs	Mixed - White and Asian
(6)	MixedOth	Any other mixed background (please describe)
(7)	Indian	Asian or Asian British - Indian
(8)	Pakistan	Asian or Asian British - Pakistani
(9)	Bngldesh	Asian or Asian British - Bangladeshi
(10)	AsianOth	Any other Asian/Asian British background (please describe)
(11)	BlackCrb	Black or Black British - Caribbean
(12)	BlackAfr	Black or Black British - African
(13)	BlackOth	Any other Black/Black British background (please describe)
(14)	Chinese	Chinese
(15)	Other	Any other (please describe)

```
ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: NatCen <> NI
     AND: EthGrp IN [WhtOth, MixedOth, AsianOth, BlackOth, Other]
```

FRS0605B.QEthnic.P[].EthOth

^I QEthnic
^I

^N (^LName)
Please can you describe your ethnic group?^N

^I^BLU Enter description of ethnic group.^I

STRING[100]

```
ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: NOT (NatCen <> NI)
```

FRS0605B.QEthnic.P[].NINatID

^I QEthnic
^I

^IS^I B^I
^N *^X^LName) What do you consider your^B national identity^B to be? Please choose your answer from this card, choose as many or as few as apply.^N
^I^BLU Probe: Any other?^I

SET [8] OF

(1)	British	British
(2)	Irish	Irish
(3)	Ulster	Ulster
(4)	NIrish	Northern Irish
(5)	English	English
(6)	Scottish	Scottish
(7)	Welsh	Welsh
(8)	Other	Other answer

```
ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Personnr := 1 TO HHSize
     AND: PRec[Personnr].Depend = Adult
     AND: NOT (NatCen <> NI)
     AND: Other IN NINatID
```

FRS0605B.QEthnic.P[].NINatOth

^I QEthnic
^I

^N (^LName)
How would you describe your national identity?^N

^I^BLU^IC - If someone describes themselves as being half English and half Irish or any combination of Welsh, Scottish, Irish or English, code them as 'Mixed British' AND then record the mix they specify.^I

(1)	Mixed	Mixed British
(2)	Describe	ENTER DESCRIPTION OF ETHNIC GROUP

ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
AND: In loop FOR Personnr := 1 TO HHSize
AND: PRec[Personnr].Depend = Adult
AND: NOT (NatCen <> NI)
AND: Other IN NINatID

FRS0605B.QEthnic.P[].NIXNatOth

^I QEthnic
^I

^I (^LName)
Enter description of ethnic GROUP.^I

STRING[100]

ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
AND: In loop FOR Personnr := 1 TO HHSize
AND: PRec[Personnr].Depend = Adult
AND: NOT (NatCen <> NI)

FRS0605B.QEthnic.P[].NIEthGrp

^I QEthnic
^I

^IS^I C^IS
^N To which of these ethnic groups does ^LName consider ^he_she belongs?^N
^I^BLU^IC This is a question of respondent's (or proxy's) opinion.

(1)	WhtBrit	White
(2)	IrishTrv	Irish Traveller
(3)	WhtOth	Any other white background (please describe)
(4)	MixedWBC	Mixed - White and Black Caribbean
(5)	MixedWBA	Mixed - White and Black African
(6)	MixedWAs	Mixed - White and Asian
(7)	MixedOth	Any other mixed background (please describe)
(8)	Indian	Asian - Indian
(9)	Pakistan	Asian - Pakistani
(10)	Bngldesh	Asian - Bangladeshi
(11)	AsianOth	Any other Asian background (please describe)
(12)	BlackCrb	Black - Caribbean
(13)	BlackAfr	Black - African
(14)	BlackOth	Any other Black background (please describe)
(15)	Chinese	Chinese
(16)	Other	Any other (please describe)

ASK IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
  AND: *In loop FOR Personnr := 1 TO HHSize*
  AND: *PRec[Personnr].Depend = Adult*
  AND: *NOT (NatCen <> NI)*
  AND: *NIEthGrp IN [WhtOth, AsianOth, BlackOth, Other]*

FRS0605B.QEthnic.P[].NIEthOth

^I QEthnic
^I

^N (^LName)
Please can you describe your ethnic group?^N

^I Enter description of ethnic group^I.

STRING[100]

WARN IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
  AND: *In loop FOR Personnr := 1 TO HHSize*
  AND: *PRec[Personnr].Depend = Adult*
  RESERVECHECK

RESERVECHECK

WARN IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
  AND: *In loop FOR Personnr := 1 TO HHSize*
  AND: *PRec[Personnr].Depend = Adult*
  RESERVECHECK

RESERVECHECK

WARN IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
  AND: *In loop FOR Personnr := 1 TO HHSize*
  AND: *PRec[Personnr].Depend = Adult*
  RESERVECHECK

RESERVECHECK

WARN IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
  AND: *In loop FOR Personnr := 1 TO HHSize*
  AND: *PRec[Personnr].Depend = Adult*
  RESERVECHECK

RESERVECHECK

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

```
WARN IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     RESERVECHECK

     RESERVECHECK
```

```
WARN IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     RESERVECHECK

     RESERVECHECK
```

```
WARN IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     RESERVECHECK

     RESERVECHECK
```

```
WARN IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     RESERVECHECK

     RESERVECHECK
```

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: PRec[Loop1].Depend IN [DepAd .. Child]
     AND: ((DMParent1[Loop1] IN [1 .. 14]) AND (DMParent2[Loop1] IN [1 ..
     14])) AND NOT (HHG.P[DMParent1[Loop1]].QRel[DMParent2[Loop1]].R IN
     [Spouse .. Cohabit])

DepParnt := (STR(DMParent1[Loop1],1,0) + ' : ' +
DMName[DMParent1[Loop1]] + '
' + STR(DMParent2[Loop1],1,0) + ' : ' +
DMName[DMParent2[Loop1]])
```

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: PRec[Loop1].Depend IN [DepAd .. Child]
     AND: ((DMParent1[Loop1] IN [1 .. 14]) AND (DMParent2[Loop1] IN [1 ..
     14])) AND NOT (HHG.P[DMParent1[Loop1]].QRel[DMParent2[Loop1]].R IN
     [Spouse .. Cohabit])
     AND: PRec[Loop1].Sex = Male

HeShe := 'he'
```

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: PRec[Loop1].Depend IN [DepAd .. Child]
     AND: ((DMParent1[Loop1] IN [1 .. 14]) AND (DMParent2[Loop1] IN [1 ..
     14])) AND NOT (HHG.P[DMParent1[Loop1]].QRel[DMParent2[Loop1]].R IN
     [Spouse .. Cohabit])
     AND: PRec[Loop1].Sex = Male

HisHer := 'his'
```

COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
AND: In loop FOR Loop1 := 1 TO HHSize
AND: PRec[Loop1].Depend IN [DepAd .. Child]
AND: ((DMParent1[Loop1] IN [1 .. 14]) AND (DMParent2[Loop1] IN [1 .. 14])) AND NOT (HHG.P[DMParent1[Loop1]].QRel[DMParent2[Loop1]].R IN [Spouse .. Cohabit])
AND: NOT (PRec[Loop1].Sex = Male)

HeShe := 'she'

COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
AND: In loop FOR Loop1 := 1 TO HHSize
AND: PRec[Loop1].Depend IN [DepAd .. Child]
AND: ((DMParent1[Loop1] IN [1 .. 14]) AND (DMParent2[Loop1] IN [1 .. 14])) AND NOT (HHG.P[DMParent1[Loop1]].QRel[DMParent2[Loop1]].R IN [Spouse .. Cohabit])
AND: NOT (PRec[Loop1].Sex = Male)

HisHer := 'her'

ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
AND: In loop FOR Loop1 := 1 TO HHSize
AND: PRec[Loop1].Depend IN [DepAd .. Child]
AND: ((DMParent1[Loop1] IN [1 .. 14]) AND (DMParent2[Loop1] IN [1 .. 14])) AND NOT (HHG.P[DMParent1[Loop1]].QRel[DMParent2[Loop1]].R IN [Spouse .. Cohabit])

FRS0605B.LegDep

^I^BLU^IC ^DMName[Loop1] is classified as a^B dependant adult^B or a^B child^B, ie. ^HeShe will not form a Benefit Unit of ^HisHer own.
To properly assess to which Benefit Unit ^HeShe belongs, please code which of the parents receive Child Benefit for ^DMName[Loop1].

^DepParnt^I

1..97

CHECK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
AND: In loop FOR Loop1 := 1 TO HHSize
AND: PRec[Loop1].Depend IN [DepAd .. Child]
AND: ((DMParent1[Loop1] IN [1 .. 14]) AND (DMParent2[Loop1] IN [1 .. 14])) AND NOT (HHG.P[DMParent1[Loop1]].QRel[DMParent2[Loop1]].R IN [Spouse .. Cohabit])
AND: LegDep[Loop1] = RESPONSE
(LegDep[Loop1] = DMParent1[Loop1]) OR (LegDep[Loop1] = DMParent2[Loop1])

^I Code ^LegDep[Loop1] is not valid for this question.^I

COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
AND: In loop FOR Loop1 := 1 TO HHSize

ABen[Loop1] := 1

CHECK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
RESERVECHECK

RESERVECHECK

```
CHECK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     RESERVECHECK
```

RESERVECHECK

```
RECORD IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
```

FRS0605B.NewBU

^I Total number of BUs^I

0..7

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
```

ABen[QHholder.DVHRPNum] := 1

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
```

Last := 1

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [Adult])
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: ABen[Loop2] <> EMPTY AND (HHG.P[Loop1].QRel[Loop2].R IN [Spouse,
     Cohabit])
```

ABen[Loop1] := ABen[Loop2]

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend = Adult)
     AND: Last < 7
```

Last := (Last + 1)

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend = Adult)
     AND: Last < 7
```

ABen[Loop1] := Last

```
COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend = Adult)
     AND: NOT (Last < 7)
```

ABen[Loop1] := 0

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [DepAd .. Child])*
AND: *LegDep[Loop1] = RESPONSE*

ABen[Loop1] := ABen[LegDep[Loop1]]

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [DepAd .. Child])*
AND: *DMParent1[Loop1] <> 0*

ABen[Loop1] := ABen[DMParent1[Loop1]]

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [DepAd .. Child])*
AND: *DMParent2[Loop1] <> 0*

ABen[Loop1] := ABen[DMParent2[Loop1]]

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [DepAd .. Child])*
AND: *NOT (DMParent2[Loop1] <> 0)*
AND: *Last < 7*

Last := (Last + 1)

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [DepAd .. Child])*
AND: *NOT (DMParent2[Loop1] <> 0)*
AND: *Last < 7*

ABen[Loop1] := Last

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *ABen[Loop1] = EMPTY AND (PRec[Loop1].Depend IN [DepAd .. Child])*
AND: *NOT (DMParent2[Loop1] <> 0)*
AND: *NOT (Last < 7)*

ABen[Loop1] := 0

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*

NewBU := Last

CHECK IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

CHECK IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

CHECK IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

CHECK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
RESERVECHECK

RESERVECHECK

COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
AND: In loop FOR Loop1 := 1 TO HHSize

NameInBU[Loop1] := ''

COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
AND: In loop FOR Loop1 := 1 TO HHSize
AND: PRec[Loop1].Depend IN [Adult]

NameInBU[ABen[Loop1]] := (NameInBU[ABen[Loop1]] + UPCASE(DMName[Loop1]) + ' ')

COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
AND: In loop FOR Loop1 := 1 TO HHSize
AND: PRec[Loop1].Depend IN [DepAd .. Child]

NameInBU[ABen[Loop1]] := (NameInBU[ABen[Loop1]] + DMName[Loop1] + ' ')

COMPUTE IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
AND: In loop FOR Loop1 := 1 TO NewBU

NameInBU[Loop1] := ('
' + STR(Loop1,1,0) + ': ' + NameInBU[Loop1])

ASK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)

FRS0605B.ShowBen

^I If you think that the computer has made a mistake in allocating the individuals to Benefit Units, go back to the household grid and check the relationship codes of each person.^I

1..1

CHECK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
RESERVECHECK

RESERVECHECK

CHECK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
RESERVECHECK

RESERVECHECK

CHECK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
RESERVECHECK

RESERVECHECK

CHECK IF: (QHholder.HHldr = RESPONSE) OR (Edit = Yes)
RESERVECHECK

RESERVECHECK

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*

hhchu11 := No

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *PRec[Loop1].Sex = RESPONSE*

DMBU[Loop1] := ABen[Loop1]

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *PRec[Loop1].Sex = RESPONSE*

HHG.P[Loop1].BenUnit := ABen[Loop1]

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *(DMBU[Loop1] = 1) AND (DMAge[Loop1] IN [0 .. 10])*

hhchu11 := Yes

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *Loop1 IN QHholder.HHldr*

RentName := (RentName + ' ' + DMName[Loop1])

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *Loop1 IN QHholder.HHldr*
AND: *DMBU[Loop1] <> 1*

NotHRPBU := 1

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *PRec[Loop1].Depend = Adult*
AND: *BUAdName[DMBU[Loop1]] =*

BUAdName[DMBU[Loop1]] := DMName[Loop1]

COMPUTE IF: *(QHholder.HHldr = RESPONSE) OR (Edit = Yes)*
AND: *In loop FOR Loop1 := 1 TO HHSize*
AND: *PRec[Loop1].Depend = Adult*
AND: *NOT (BUAdName[DMBU[Loop1]] =)*

BUAdName[DMBU[Loop1]] := (BUAdName[DMBU[Loop1]] + ' and ' + DMName[Loop1])

FRS0605B.QAccomdat

Questions about accommodation

RECORD IF: HHG.P[HHSize].BenUnit = RESPONSE

FRS0605B.QAccomdat.Ten1Ex

^I QAccomDat
^I
^BLU^IC^I^KeyTxt

^SuppTxt^I^BLU

OPEN

RECORD IF: HHG.P[HHSize].BenUnit = RESPONSE

FRS0605B.QAccomdat.Ten2Rs

^I QAccomDat
^I
^NCan I just check do you live rent free because you receive 100% housing benefit?^N
^I^BLU^IC If Yes recode tenure to code 4 (rent it).
Please check their rent/mortgage is not paid by^B benefits^B. Only accommodation provided by someone else (employer, relative, etc) is rent-free.^I

(1)	Passed	Passed
(2)	Hard	Hard
(3)	Soft	Soft
(4)	Suppress	Suppressed

RECORD IF: HHG.P[HHSize].BenUnit = RESPONSE

FRS0605B.QAccomdat.Ten2Ex

^I QAccomDat
^I
^I^BLU^IC ^SuppTxt^I

OPEN

ASK IF: HHG.P[HHSize].BenUnit = RESPONSE

FRS0605B.QAccomdat.Tenure

^I QAccomDat
^I
^IS^I D^I ^N
In which of these ways do you occupy this accommodation?^N

(1)	Outright	Own it outright
(2)	Mortgage	Buying it with the help of a mortgage or loan
(3)	Part	Pay part rent and part mortgage (^SharOwn)
(4)	Rents	Rent it
(5)	RentFree	Live here rent-free (including in a relative's/friend's property; excluding squatting)
(6)	Squattin	Squatting

WARN IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *Tenure = NONRESPONSE*
ERROR

^I^KeyTxt^I

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *Tenure = NONRESPONSE*

FRS0605B.QAccomdat.Ten1Ex

^I QAccomDat
^I
^BLU^IC^I^KeyTxt

^SuppTxt^I^BLU

OPEN

WARN IF: *HHG.P[HHSize].BenUnit = RESPONSE*
Tenure <> RentFree

Tenure <> RentFree

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *Ten2Rs = Suppressed*

FRS0605B.QAccomdat.Ten2Ex

^I QAccomDat
^I
^I^BLU^IC ^SuppTxt^I

OPEN

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *Tenure = Part*

FRS0605B.QAccomdat.SOBuy

^I QAccomDat
^I
^I^BLU^IC Ask or record:
^SOwners:^I Are you still buying your share in this (house/flat), or have you now paid off that mortgage or loan?

(1)	StillM	Still buying
(2)	Paid	Mortgage is paid off

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.SubLet

^I QAccomDat
^I
^N Do you have a formal arrangement to let, or sub-let, any part of this accommodation to someone who is^B not^B a member of your household?^N

(1)	Yes	Yes
(2)	No	No

COMPUTE IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *SubLet = Yes*

**How := ('Thinking just of the accommodation occupied ' + 'by your household,
how')**

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *SubLet = Yes*

FRS0605B.QAccomdat.SubLetY

^I QAccomDat
^I
^N Who is that?··^N^I^BLU Code first that applies.

^BLU^IC Close relatives = Householder's partner,
parent (incl.step-), son or daughter (incl. step-),
brother or sister, or spouse of any of these.^N

(1)	ClosRel	Close relative
(2)	OthRel	Other relative
(3)	NonRel	Non-relative

COMPUTE IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *NOT (SubLet = Yes)*

How := 'How'

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.Rooms

^I QAccomDat
^I
^I
In general, include any room which is habitable or usable by the household all year round.

If a room is open-plan count it as 2 rooms if it is divided by a fixed sliding or folding partition.^I

0..20

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.RoomShar

^I QAccomDat
^I
^N Are any of these rooms shared with anyone who is^B not^B a member of your household?^N

^I^BLU^IC If 'No' enter '0'.
If 'Yes',^B ask:^B How many? and enter number.^I

0..10

WARN IF: *HHG.P[HHSize].BenUnit = RESPONSE*
RoomShar <> 2

^I The answer you have entered means two rooms are shared. If you intended to answer 'No' to this question, please change the code to '0' (zero). Otherwise, suppress this warning.^I

Ask if: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.Bedroom

^I QAccomDat
^I
^N^How many bedrooms do you have in this accommodation?^N

^I^BLU^IC Include any room used for sleeping.

1..10

Compute if: *HHG.P[HHSize].BenUnit = RESPONSE*

ChkTxt := ('cannot be greater than total number of rooms: ' + 'please check your answers and amend as necessary.')

Check if: *HHG.P[HHSize].BenUnit = RESPONSE*
RoomShar <= Rooms

^I Number of shared rooms ^ChkTxt^I

Check if: *HHG.P[HHSize].BenUnit = RESPONSE*
Bedroom <= Rooms

^I Number of bedrooms ^ChkTxt^I

Ask if: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.MainAcc

^I QAccomDat
^I
^I Note that this is accommodation occupied by the household. If the household occupies a flat in a converted house, code as a flat.^I^Blu

(1)	HseBun	a house or bungalow
(2)	FltMas	a flat or maisonette
(3)	ARoom	a room or rooms
(4)	Oth	or something else?

Ask if: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.Shelter

^I QAccomDat
^I
^N Is this sheltered accommodation?^N

^I^BLU^IC Housing with a warden and/or alarms.^I

(1)	Yes	Yes
(2)	No	No

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

Detach := (N + 'detached')

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

SemiDetach := (N + 'semi-detached')

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

Terrace := (N + 'or terraced/end of terrace?')

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

PurposeBuilt := ''

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

ConvertedHouse := ''

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

MobileHome := ''

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

OtherKind := ''

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = HseBun
```

Accommodation := 'the house or bungalow'

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

Detach := ''

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

SemiDetach := ''

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

Terrace := ''

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

PurposeBuilt := (N + 'a purpose-built block')

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

ConvertedHouse := (N + 'or a converted house/some other kind of building?')

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

MobileHome := ''

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

OtherKind := ''

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: MainAcc = FltMas
```

Accommodation := (N + 'the flat/maisonette')

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

Detach := ''

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

SemiDetach := ''

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

Terrace := ''

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

PurposeBuilt := ''

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

ConvertedHouse := ''

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

MobileHome := (N + 'a caravan, mobile home or houseboat')

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

OtherKind := (N + 'or some other kind of accommodation?')

```
COMPUTE IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: NOT (MainAcc = FltMas)
```

Accommodation := (N + 'the accommodation')

```
ASK IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
```

FRS0605B.QAccomdat.TypeAcc

^I QAccomDat
^I
^I
Houses which are joined only by a garage (link-detached) should be coded detached.^I

(1)	Detached	^Detach
(2)	Semi_det	^SemiDetach
(3)	Terraced	^Terrace
(4)	Purpose_	^PurposeBuilt
(5)	Converte	^ConvertedHouse
(6)	Mobile_h	^MobileHome
(7)	Other_ki	^OtherKind

```
CHECK IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: TypeAcc = RESPONSE
     (((IN(TypeAcc,[???])) AND (MainAcc = HseBun)) OR
     ((IN(TypeAcc,[???,???])) AND (MainAcc = FltMas))) OR ((MainAcc = Oth)
     AND (IN(TypeAcc,[???])))
```

^I This code is not valid for this accommodation.^I

```
CHECK IF: HHG.P[HHSize].BenUnit = RESPONSE
     AND: MainAcc IN [HseBun .. FltMas, Oth]
     AND: TypeAcc IN [Detached .. Purpose_built]
     (SubLet <> Yes) AND INVOLVING(MainAcc)
```

^I As part of this accommodation is sub-let, this household cannot be coded as occupying a whole house, flat etc.
Use another code at MainAcc instead.^I

Ask if: *HHG.P[HHSize].BenUnit = RESPONSE*
And: *(TypeAcc IN [Purpose_built, Converted_house]) OR (MainAcc = ARoom)*

FRS0605B.QAccomdat.Floor

^I QAccomDat
^I
^N What is the floor level of this household's accommodation?^N

(1)	Basement	Basement/semi-basement
(2)	Ground	Ground floor/street level
(3)	First	1st floor (floor above street level)
(4)	Second	2nd floor
(5)	Third	3rd floor
(6)	Fourth	4th Floor
(7)	Fifthup	5th to 9th floor
(8)	TenthUp	10th floor or higher
(9)	DNK	Don't Know

Ask if: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.Entry

^I QAccomDat
^I
^N Are there any physical barriers to entry to the house/flat/accommodation? ^I

^BLU^IC Code all that apply^I

SET [6] OF

(1)	None	No physical impediments or barriers
(2)	Entrance	Locked common entrance
(3)	Gates	Locked gates
(4)	Staff	Security staff, concierge or other gatekeeper
(5)	Phone	Entry phone access, intercom
(6)	Animal	Guard dog/patrol animal
(7)	Warden	Warden controlled

Check if: *HHG.P[HHSize].BenUnit = RESPONSE*
And: *None IN Entry*
Entry.CARDINAL = 1

^I'None' is an exclusive code.^I

Ask if: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.YearLive

^I QAccomDat
^I
^N For how many years have you, ^N(that is ^PHRPName),^N lived at this address?^N

^I^BLU^IC Probe to classify.^I

(1)	Less12m	Less than 12 months
(2)	Fr1yr	12 months but less than 2 years
(3)	Fr2yr	2 years but less than 3 years
(4)	Fr3yr	3 years but less than 5 years
(5)	Fr5yr	5 years but less than 10 years
(6)	Fr10yr	10 years but less than 20 years
(7)	Fr20yr	20 years or longer

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *YearLive = Less12m*

FRS0605B.QAccomdat.MonLive

^I QAccomDat
^I
^N For how many months have you, (that is ^PHRPName), lived at this address?^N

^I^BLU^IC Enter number of months, to nearest whole month.^I

0..11

COMPUTE IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *NewBU = 1*

HHStat := Conv

ASK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *NOT (NewBU = 1)*

FRS0605B.QAccomdat.HHStat

^I QAccomDat
^I
^I^BLU^IC Classify this household as one of the following:
NOTE:^B Conventional Households^B include:
- tenure is owner occupier and a 2nd or 3rd benefit unit is paying rent
- tenure is rent free but 2nd or 3rd BU receives Housing Benefit.
- 2rd or 3rd BU members paying rent to the householder(s) in BU1 are also named as householders (this is similar to part owned/part rented tenure)^I.

(1) Conv Conventional household: ie. single person or couple - with other family and/or boarder(s) and/or lodger(s)
(2) Shared 'Shared' household arrangements: identity of HRP is unclear or arbitrary - eg. students, nurses, unrelated adults etc, sharing ON EQUAL BASIS

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.AnyVeh

^I QAccomDat
^I
^N Do you at present own or have continuous use of any motor vehicles?^N

(1) Yes Yes
(2) No No

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.VehNumb

^I QAccomDat
^I
^N Number of vehicles.^N

0..8

Record if: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.AdultH

^I QAccomDat
^I
^N Actual number of adults in household.^N

0..14

Record if: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.DepChldH

^I QAccomDat
^I
^N Actual number of children in household.^N

0..14

Record if: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.DatYrAgo

^I QAccomDat
^I
^N Date one year ago^N

DATE

Record if: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.BenUnits

^I QAccomDat
^I
^N Actual number of Benefit Units in household.^N

0..7

Record if: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.Dentist

^I QAccomDat
^I
^N Anyone having NHS visits to the dentist?^N

(1)	Yes	Yes
(2)	No	No

Record if: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.EyeTest

^I QAccomDat
^I
^N Anyone having NHS eyetests?^N

(1)	Yes	Yes
(2)	No	No

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.Specs

^I QAccomDat
^I
^N Anyone having NHS glasses/lenses^N?

(1)	Yes	Yes
(2)	No	No

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.Hospital

^I QAccomDat
^I
^N Anyone having NHS hospital treatment?^N

(1)	Yes	Yes
(2)	No	No

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.Pres

^I QAccomDat
^I
^B Anyone having NHS prescriptions?^N

(1)	Yes	Yes
(2)	No	No

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.SchMeal

^I QAccomDat
^I
^N Anyone having school meals?^N

(1)	Yes	Yes
(2)	No	No

RECORD IF: *HHG.P[HHSize].BenUnit = RESPONSE*

FRS0605B.QAccomdat.SchMilk

^I QAccomDat
^I
^N Anyone having school milk?^N

(1)	Yes	Yes
(2)	No	No

RECORD IF: HHG.P[HHSize].BenUnit = RESPONSE

FRS0605B.QAccomdat.WelfMilk

^I QAccomDat
^I
^N Anyone having welfare milk?^N

(1)	Yes	Yes
(2)	No	No

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

CHECK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
RESERVECHECK

RESERVECHECK

CHECK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
RESERVECHECK

RESERVECHECK

CHECK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
RESERVECHECK

RESERVECHECK

CHECK IF: *HHG.P[HHSize].BenUnit = RESPONSE*
RESERVECHECK

RESERVECHECK

COMPUTE IF: *HHG.P[HHSize].BenUnit = RESPONSE*
AND: *QAccomdat.HHStat = Shared*
AND: *In loop FOR Loop1 := 1 TO NewBU*

BUHBElig[Loop1] := Yes

COMPUTE ALWAYS:

QAccomdat.AdultH := AllAd

COMPUTE ALWAYS:

QAccomdat.DepChldH := AllCh

COMPUTE ALWAYS:

QAccomdat.DatYrAgo := DLYear

COMPUTE ALWAYS:

QAccomdat.BenUnits := NewBU

FRS0605B.QRenting

Questions about renters

***ASK IF:** QAccomdat.Tenure IN [Part .. Squatting]*

FRS0605B.QRenting.Landlord

^I QRenting
^I
^I
If property is let through a letting agent or estate agent, the question refers to the owner not the agent, so please probe to try to find out who actually owns the property.

If the respondent does not know who the landlord is, use code 7 (other private individual) rather than coding 'Don't know'.

Code 1 (^LANIHE) includes people renting from Housing Action Trusts.

Code 2 (housing association etc.) includes Registered Social Landlords. Nearly all housing associations are now Registered Social Landlords but continue to be known as housing associations.

^I

(1)	Council	^Council1
(2)	Assocn	A housing association, charitable trust or Local Housing Company
(3)	OrgEmpl	Employer (organisation) of a household member
(4)	OrgOth	Another organisation
(5)	FrndRel	Relative/friend (before you lived here) of household member
(6)	IndEmpl	Employer (individual) of a household member
(7)	OthIndiv	Another individual, private landlord or Letting Agency

***RECORD IF:** QAccomdat.Tenure IN [Part .. Squatting]*

FRS0605B.QRenting.LLEx

^I QRenting
^I
^I^BLU^IC^KeyTxt

^SuppTxt^I

OPEN

***WARN IF:** QAccomdat.Tenure IN [Part .. Squatting]*
***AND:** Edit = No*
***AND:** Landlord = NONRESPONSE*
ERROR

^I^BLU^IC^KeyTxt^I

ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
AND: Edit = No
AND: Landlord = NONRESPONSE

FRS0605B.QRenting.LLEx

^I QRenting
^I
^I^BLU^IC^KeyTxt

^SuppTxt^I

OPEN

COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
AND: Landlord = Council

Allowance := 'Rebate'

COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
AND: NOT (Landlord = Council)

Allowance := 'Allowance'

ASK IF: QAccomdat.Tenure IN [Part .. Squatting]

FRS0605B.QRenting.Furnish

^I QRenting
^I
^N Is this accomodation provided...
^N

(1)	Furnishe	furnished,
(2)	PartFurn	partly furnished (eg. curtains and carpets only),
(3)	Unfurnis	or unfurnished?

ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
AND: Landlord IN [FrndRel .. OthIndiv]

FRS0605B.QRenting.ResLL

^I QRenting
^I
^N Does the landlord live in the building?
^N

(1)	Yes	Yes
(2)	No	No

ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
AND: Landlord IN [FrndRel .. OthIndiv]
AND: (ResLL = Yes) AND (QAccomdat.TypeAcc = Purpose_built)

FRS0605B.QRenting.ResLL2

^I QRenting
^I
^N Does the landlord live in the same flat as you or not?@^N/

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: (ResLL = No) OR (ResLL2 = No)
     AND: NatCen <> NI
```

FRS0605B.QRenting.YStart

^I QRenting
^I
^I^BLU^IC Ask or record^I
In which year did you first become a tenant of this accommodation?

^I^IC 'You'=Person(s) named at 'Hhldr', that is... ^RentName.
^I

(1)	Bef1988	1988 or earlier
(2)	ToFeb97	From 1989 to February 1997
(3)	AftMar97	March 1997 or later

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: (ResLL = No) OR (ResLL2 = No)
     AND: NOT (NatCen <> NI)
```

FRS0605B.QRenting.NIYstart

^I QRenting
^I
^I^BLU^IC Ask or record^I
In which year did you first become a tenant of this accommodation?

^I^IC 'You'=Person(S) named AT 'Hhldr', that is... ^RentName.
^I

(1)	Bef1978	1978 or earlier
(2)	Aft1979	1979 or later

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: (ResLL = No) OR (ResLL2 = No)
     AND: ((NatCen <> NI) AND (YStart IN [ToFeb97 .. AftMar97])) OR ((NatCen
     = NI) AND (NIYstart = Aft1979))
```

FRS0605B.QRenting.Ctract

^I QRenting
^I
^N When you started to rent this accommodation ^N ^I^BLU...Running prompt...
^I

(1)	Signed	^N...did you and the landlord sign a written agreement,^N
(2)	NotSign	^N...did you have a written agreement which you didn't sign,^N
(3)	Unwrittn	^N...or did you just have an unwritten agreement?^N

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: NatCen = NI
```

various := 'various'

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Landlord IN [FrndRel .. OthIndiv]*
    AND: *NOT (NatCen = NI)*

various := 'various other'

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Landlord IN [FrndRel .. OthIndiv]*
    AND: *NatCen <> NI*
    AND: *(YStart IN [ToFeb97 .. AftMar97]) OR ((ResLL = Yes) AND (ResLL2 = EMPTY OR (ResLL2 = Yes)))*
    AND: *Scotland = Yes*

AssuredSH := 'Short Assured'

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Landlord IN [FrndRel .. OthIndiv]*
    AND: *NatCen <> NI*
    AND: *(YStart IN [ToFeb97 .. AftMar97]) OR ((ResLL = Yes) AND (ResLL2 = EMPTY OR (ResLL2 = Yes)))*
    AND: *NOT (Scotland = Yes)*

AssuredSH := 'Assured Shorthold'

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Landlord IN [FrndRel .. OthIndiv]*
    AND: *NatCen <> NI*
    AND: *(YStart IN [ToFeb97 .. AftMar97]) OR ((ResLL = Yes) AND (ResLL2 = EMPTY OR (ResLL2 = Yes)))*

FRS0605B.QRenting.TenType

^I QRenting
^I
^IS^I F^I ^N
Can you tell me what kind of tenancy you have?^N

^I^BLU^IC If tenancy type written on contract/notice ask respondent to read out.^I

(1)	AssSHold	^AssuredSH
(2)	Assured	Assured
(3)	Regulate	Regulated (tenancy must have started in 1988 or earlier)
(4)	ResLL	Resident landlord
(5)	LetEduc	Let by educational institution
(6)	OthLet	Other type of let

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: NatCen <> NI
     AND: (YStart IN [ToFeb97 .. AftMar97]) OR ((ResLL = Yes) AND (ResLL2 =
     EMPTY OR (ResLL2 = Yes)))
     AND: TenType = OthLet
```

FRS0605B.QRenting.OthType

^I QRenting
^I
^IS^I G^I ^N
There are ^various ways in which landlords can let accommodation. Will you please look at this card and tell me if the letting is one of these?^N

(1)	Crown	Crown tenancy/licence (includes H.M Forces)
(2)	Servive	Service occupancy (excludes H.M. Forces)
(3)	BusAgr	Business or agricultural tenancy
(4)	AssAgr	Assured agricultural occupancy
(5)	Asylum	Asylum seeker let (issued by National Asylum Support Service NASS)
(6)	Holiday	Holiday let
(7)	OthLet	Other type of let

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: NatCen <> NI
     AND: (YStart IN [ToFeb97 .. AftMar97]) OR ((ResLL = Yes) AND (ResLL2 =
     EMPTY OR (ResLL2 = Yes)))
     AND: TenType = AssSHold
```

FRS0605B.QRenting.LowShort

^I QRenting
^I
^N Is this a low season let?^N
^I^BLU^IC This refers to an out of season let^I

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: NOT (NatCen <> NI)
```

FRS0605B.QRenting.OthType

^I QRenting
^I
^IS^I G^I ^N
There are ^various ways in which landlords can let accommodation. Will you please look at this card and tell me if the letting is one of these?^N

(1)	Crown	Crown tenancy/licence (includes H.M Forces)
(2)	Servive	Service occupancy (excludes H.M. Forces)
(3)	BusAgr	Business or agricultural tenancy
(4)	AssAgr	Assured agricultural occupancy
(5)	Asylum	Asylum seeker let (issued by National Asylum Support Service NASS)
(6)	Holiday	Holiday let
(7)	OthLet	Other type of let

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: Ctract IN [Signed .. NotSign]
     AND: ((((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = ToFeb97))
     OR (((Scotland = Yes) AND (ResLL2 <> Yes)) AND (YStart IN [ToFeb97 ..
     AftMar97]))) OR (((NatCen = NI) AND (ResLL2 <> Yes)) AND (NIYstart =
     Aft1979))
     AND: NatCen = NI
```

Assured := 'a Protected'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: Ctract IN [Signed .. NotSign]
     AND: ((((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = ToFeb97))
     OR (((Scotland = Yes) AND (ResLL2 <> Yes)) AND (YStart IN [ToFeb97 ..
     AftMar97]))) OR (((NatCen = NI) AND (ResLL2 <> Yes)) AND (NIYstart =
     Aft1979))
     AND: NatCen = NI
```

Tenancy := 'a Protected Shorthold Tenancy'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: Ctract IN [Signed .. NotSign]
     AND: ((((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = ToFeb97))
     OR (((Scotland = Yes) AND (ResLL2 <> Yes)) AND (YStart IN [ToFeb97 ..
     AftMar97]))) OR (((NatCen = NI) AND (ResLL2 <> Yes)) AND (NIYstart =
     Aft1979))
     AND: NatCen = NI
```

Shorthold := 'a Protected Shorthold'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: Ctract IN [Signed .. NotSign]
     AND: ((((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = ToFeb97))
     OR (((Scotland = Yes) AND (ResLL2 <> Yes)) AND (YStart IN [ToFeb97 ..
     AftMar97]))) OR (((NatCen = NI) AND (ResLL2 <> Yes)) AND (NIYstart =
     Aft1979))
     AND: NOT (NatCen = NI)
```

Assured := 'an Assured'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: Ctract IN [Signed .. NotSign]
     AND: ((((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = ToFeb97))
     OR (((Scotland = Yes) AND (ResLL2 <> Yes)) AND (YStart IN [ToFeb97 ..
     AftMar97]))) OR (((NatCen = NI) AND (ResLL2 <> Yes)) AND (NIYstart =
     Aft1979))
     AND: NOT (NatCen = NI)
```

Tenancy := 'for an Assured Shorthold Tenancy'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Landlord IN [FrndRel .. OthIndiv]
     AND: Ctract IN [Signed .. NotSign]
     AND: ((((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = ToFeb97))
     OR (((Scotland = Yes) AND (ResLL2 <> Yes)) AND (YStart IN [ToFeb97 ..
     AftMar97]))) OR (((NatCen = NI) AND (ResLL2 <> Yes)) AND (NIYstart =
     Aft1979))
     AND: NOT (NatCen = NI)
```

Shorthold := 'an Assured Shorthold'

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *Landlord IN [FrndRel .. OthIndiv]*
AND: *Ctract IN [Signed .. NotSign]*
AND: *((((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = ToFeb97)) OR (((Scotland = Yes) AND (ResLL2 <> Yes)) AND (YStart IN [ToFeb97 .. AftMar97]))) OR (((NatCen = NI) AND (ResLL2 <> Yes)) AND (NIYstart = Aft1979))*
AND: *NatCen <> NI*
AND: *TenType = NONRESPONSE OR OthType = NONRESPONSE*

FRS0605B.QRenting.Short1

^I QRenting
^I
^N There is a form of tenancy called ^Assured Shorthold. It had to be initially for a fixed period and you had to be given a special notice in writing by the landlord that told you it was ^Tenancy. Here is an example of a notice to a tenant saying that the tenancy is ^Shorthold.
^BLU SHOW EXAMPLE OF NOTICE.^BLU

Does your notice state that it is ^Assured Shorthold or not?^I

(1)	Assured	Yes, an Assured Shorthold
(2)	Other	Other agreement

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *Landlord IN [FrndRel .. OthIndiv]*
AND: *Ctract IN [Signed .. NotSign]*
AND: *((NatCen <> NI) AND (Scotland <> Yes)) AND (YStart = AftMar97)*
AND: *TenType = NONRESPONSE OR OthType = NONRESPONSE*

FRS0605B.QRenting.Short2

^I QRenting
^I
^N Most tenancies are Assured Shortholds. There are others, just called 'Assured'. For these you have to be given a notice, in writing by the landlord, that tells you it is NOT an Assured Shorthold agreement.

^BLU SHOW EXAMPLE OF NOTICE.^BLU

Does your agreement or notice state that it is NOT an Assured Shorthold?^N

^I^BLU^IC The law changed in March 1997 to the effect that by default all tenancy agreements are assured shortholds, unless the landlord gave written notice to the contrary.^I

(1)	NotAss	Not an Assured Shorthold
(2)	Other	No, other agreement

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *Landlord IN [FrndRel .. OthIndiv]*
AND: *((((Scotland <> Yes) AND (YStart = Bef1988)) AND ((ResLL = No) OR (ResLL2 = No))) OR ((Scotland = Yes) AND ((ResLL = No) OR (ResLL2 = No)))) OR ((NatCen = NI) AND ((ResLL = No) OR (ResLL2 = No)))*
AND: *NatCen = NI*

```
assessed := 'assessed'
```

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *Landlord IN [FrndRel .. OthIndiv]*
AND: *((((Scotland <> Yes) AND (YStart = Bef1988)) AND ((ResLL = No) OR (ResLL2 = No))) OR ((Scotland = Yes) AND ((ResLL = No) OR (ResLL2 = No)))) OR ((NatCen = NI) AND ((ResLL = No) OR (ResLL2 = No)))*
AND: *NatCen = NI*

rent_officer := ('rent officer for Northern Ireland who will set up an independent ' + 'committee to assess and fix an appropriate rent')

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *Landlord IN [FrndRel .. OthIndiv]*
AND: *((((Scotland <> Yes) AND (YStart = Bef1988)) AND ((ResLL = No) OR (ResLL2 = No))) OR ((Scotland = Yes) AND ((ResLL = No) OR (ResLL2 = No)))) OR ((NatCen = NI) AND ((ResLL = No) OR (ResLL2 = No)))*
AND: *NOT (NatCen = NI)*

assessed := 'registered'

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *Landlord IN [FrndRel .. OthIndiv]*
AND: *((((Scotland <> Yes) AND (YStart = Bef1988)) AND ((ResLL = No) OR (ResLL2 = No))) OR ((Scotland = Yes) AND ((ResLL = No) OR (ResLL2 = No)))) OR ((NatCen = NI) AND ((ResLL = No) OR (ResLL2 = No)))*
AND: *NOT (NatCen = NI)*

rent_officer := ('local rent officer or rent assessment committee to decide on a ' + 'fair rent which is then registered')

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *Landlord IN [FrndRel .. OthIndiv]*
AND: *((((Scotland <> Yes) AND (YStart = Bef1988)) AND ((ResLL = No) OR (ResLL2 = No))) OR ((Scotland = Yes) AND ((ResLL = No) OR (ResLL2 = No)))) OR ((NatCen = NI) AND ((ResLL = No) OR (ResLL2 = No)))*

FRS0605B.QRenting.FairRent

^I QRenting
^I
^N Has the rent been registered by the local rent officer or rent committee?^N

(1)	Yes	Yes
(2)	No	No

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*

FRS0605B.QRenting.AccJob

^I QRenting
^I
^N Does this accommodation go with the present job of anyone in your household?^N

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccJob = Yes
```

FRS0605B.QRenting.AccJbPer

^I QRenting
^I
^N Who is that?^N

^I^BLU^IC Code all that apply.^I

SET [14] OF

(1)	Per1	^DMName[1]
(2)	Per2	^DMName[2]
(3)	Per3	^DMName[3]
(4)	Per4	^DMName[4]
(5)	Per5	^DMName[5]
(6)	Per6	^DMName[6]
(7)	Per7	^DMName[7]
(8)	Per8	^DMName[8]
(9)	Per9	^DMName[9]
(10)	Per10	^DMName[10]
(11)	Per11	^DMName[11]
(12)	Per12	^DMName[12]
(13)	Per13	^DMName[13]
(14)	Per14	^DMName[14]

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccJob = Yes
     AND: In loop FOR Index := 1 TO 14
     AND: Index IN AccJbPer
     PRec[].Depend[Index] = Adult
```

^I Code ^Index is not valid for this question.^I

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: QAccomdat.HHStat = Shared
```

es_household := (' you, that is, just ^B' + HRPNames + '^B,')

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: QAccomdat.HHStat = Shared
```

IsAre := 'Are'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: NOT (QAccomdat.HHStat = Shared)
```

es_household := 'es your household'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: NOT (QAccomdat.HHStat = Shared)
```

IsAre := 'Is'

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *PTenure IN [Rents, Part]*

FRS0605B.QRenting.RentDoc

^I QRenting
^I
^N Do you have a rent book, rent card, Housing Benefit statement or some other rent document that you could consult?^N

^I^BLU^IC If HB statement available please consult this.^I

(1)	HBStmt	Housing Benefit Statement
(2)	Oth	Some other document
(3)	None	None

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *PTenure IN [Rents, Part]*
AND: *RentDoc IN [HBStmt, Oth]*

Consult_the_document := ' Please consult the document.'

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *PTenure IN [Rents, Part]*
AND: *NOT (RentDoc IN [HBStmt, Oth])*

Consult_the_document := ''

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *PTenure IN [Rents, Part]*

FRS0605B.QRenting.Rent

^I QRenting
^I
^I If in arrears, enter amount last paid but open a Note and give the date of payment (and say if an extra amount is included to pay towards the arrears).
If 100% rent rebate/HB is received and water/sewerage rates and other services etc are^B not^B included in the rent, then the amount at 'Rent' should be zero. But if water, etc rates ARE included in the rent, then the amount paid for these rates should be entered at the question 'Rent'.

If rent includes an element for the business part of the property (eg a shop beneath a flat), and the amount for the residential part cannot be determined, enter 'DK' at 'Rent'.^I

0.00..999997.00

RECORD IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *PTenure IN [Rents, Part]*

FRS0605B.QRenting.RentEx

^I QRenting
^I
^I^BLU^IC ^KeyTxt

^SuppTxt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Edit = No
     AND: Rent = NONRESPONSE
     ERROR
```

^I^KeyTxt^I

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Edit = No
     AND: Rent = NONRESPONSE
```

FRS0605B.QRenting.RentEx

^I QRenting
^I
^I^BLU^IC ^KeyTxt

^SuppTxt^I

OPEN

```
RECORD IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

FRS0605B.QRenting.RentPx

^I QRenting
^I
^I^BLU^IC ^Pd97Txt^I

OPEN

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

FRS0605B.QRenting.RentPd

^I QRenting
^I
^N How long does this cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
     AND: RentPd = Note
```

FRS0605B.QRenting.RentPx

^I QRenting
^I
^I^BLU^IC ^Pd97Txt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
     AND: Edit = Yes
     RentPd <> Note
```

^I Editor: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^I

FRS0605B.QRenting.Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0

PdConW[1] := 1
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0

PdConW[2] := 2
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0

PdConW[3] := 3
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0

PdConW[4] := 4
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0

PdConW[5] := 4.333
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0

PdConW[7] := 8.67
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0

PdConW[8] := 6.5
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0

PdConW[9] := 5.78
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0

PdConW[10] := 5.2
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0605B.QRenting (continued)

Questions about renters

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
     AND: RentPd IN [OneWeek .. Year]
     AND: LWeekly1 >= 0.01
```

RentWkly := LWeekly1

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
     AND: RentPd IN [OneWeek .. Year]
     AND: LWeekly1 >= 0.01
     AND: Landlord = Council
     (RentWkly < 101) AND INVOLVING(RentPd,Rent)
```

^I This comes to £^RentWkly a week.
Rents for Council tenants are normally below £100 a week.^I

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent > 0
     AND: RentPd IN [OneWeek .. Year]
     AND: LWeekly1 >= 0.01
     AND: Edit = No
     ((RentWkly < 151) OR (Landlord = Council)) AND INVOLVING(RentPd,Rent)
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent = REFUSAL
```

HMissVar := (HMissVar + 1)

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent = DONTKNOW
```

FRS0605B.QRenting.RentDK

^I QRenting
^I
^I^BLU^IC@|Is this 'don't know' because rent is partly for^B business^B,
@|@|@|and you cannot establish a separate amount for the^B domestic^B accommodation?^I

(1)	Yes	Yes (Please give full details in a Note)
(2)	No	No

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: PTenure IN [Rents, Part]
     AND: Rent = DONTKNOW
     AND: RentDK <> Yes
```

HMissVar := (HMissVar + 1)

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Rent <> EMPTY*

FRS0605B.QRenting.RentHol

^I QRenting
^I
^N Do you have a rent holiday?^N

^I^BLU^IC Some people know this as 'Rent free week(s)'.^I

(1)	Yes	Yes
(2)	No	No

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Rent <> EMPTY*
    AND: *RentHol = Yes*

FRS0605B.QRenting.WeekHol

^I QRenting
^I
^N For how many weeks of the year do you have a rent holiday?^N

1..52

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Rent <> EMPTY*
    AND: *RentHol = Yes*
    WeekHol <= 8

^I Rent holidays do not normally exceed 8 weeks per year.^I

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Landlord = Council*

allowed := 'allowed'

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *Landlord = Council*

directly := ''

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *NOT (Landlord = Council)*

allowed := 'receiving'

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *NOT (Landlord = Council)*

directly := ', either directly or by having it paid to your landlord'

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*

FRS0605B.QRenting.HBenefit

^I QRenting
^I
^N Are you ^allowed Housing Benefit or Rent ^Allowance, to help with paying your rent^directly?^N

(1)	Yes	Yes
(2)	No	No

RECORD IF: *QAccomdat.Tenure IN [Part .. Squatting]*

FRS0605B.QRenting.HBenEx

^I QRenting
^I
^I^BLU^IC ^KeyTxt

^SuppTxt^I

OPEN

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *Edit = No*
AND: *HBenefit = NONRESPONSE*
ERROR

^I^KeyTxt^I

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *Edit = No*
AND: *HBenefit = NONRESPONSE*

FRS0605B.QRenting.HBenEx

^I QRenting
^I
^I^BLU^IC ^KeyTxt

^SuppTxt^I

OPEN

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *((Rent = 0) AND (Rent = RESPONSE)) AND (HBenefit = Yes)*

FRS0605B.QRenting.Rebate

^I QRenting
^I
^N You said that you paid no rent last time, is that because you get 100% Housing Benefit?^N

(1)	Yes	Yes
(2)	No	No

Ask if: *QAccomdat.Tenure IN [Part .. Squatting]*
 And: *((Rent = 0) AND (Rent = RESPONSE)) AND ((HBenefit = No) OR (Rebate = No))*

FRS0605B.QRenting.RebateO

^I QRenting
^I
^N Can I just check, what is the reason for your paying no rent last time?^N

STRING[60]

Compute if: *QAccomdat.Tenure IN [Part .. Squatting]*
 And: *HHSize > 1*

you_all := 'you and the other members of your household'

Compute if: *QAccomdat.Tenure IN [Part .. Squatting]*
 And: *NOT (HHSize > 1)*

you_all := 'you'

Compute if: *QAccomdat.Tenure IN [Part .. Squatting]*
 And: *HBenefit = Yes*

NCDVIB := (NCDVIB + 1)

Ask if: *QAccomdat.Tenure IN [Part .. Squatting]*
 And: *HBenefit = Yes*

FRS0605B.QRenting.HBenAmt

^I QRenting
^I
^N How much Housing Benefit/ rent rebate/ allowance are ^you_all allowed?^N

^I^BLU^IC Some respondents may receive more housing benefit than the amount of their rent.^I

0.01..997.00

Compute if: *QAccomdat.Tenure IN [Part .. Squatting]*
 And: *HBenefit = Yes*
 And: *HBenAmt = NONRESPONSE*

HMissVar := (HMissVar + 1)

Record if: *QAccomdat.Tenure IN [Part .. Squatting]*
 And: *HBenefit = Yes*
 And: *HBenAmt > 0*

FRS0605B.QRenting.HBenPx

^I QRenting
^I
^I^BLU^IC^Pd97Txt^I

OPEN

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
```

FRS0605B.QRenting.HBenPd

^I QRenting
^I
^N How long does this cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
     AND: HBenPd = Note
```

FRS0605B.QRenting.HBenPx

^I QRenting
^I
^I^BLU^IC^Pd97Txt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
     AND: Edit = Yes
     HBenPd <> Note
```

^I Editor: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^I

FRS0605B.QRenting.Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
```

PdConW[10] := 5.2

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *HBenefit = Yes*
    AND: *HBenAmt > 0*

PdConW[13] := 13

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *HBenefit = Yes*
    AND: *HBenAmt > 0*

PdConW[26] := 26

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *HBenefit = Yes*
    AND: *HBenAmt > 0*

PdConW[52] := 52

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *HBenefit = Yes*
    AND: *HBenAmt > 0*
    AND: *(PAmount > 0) AND (PPeriod IN [OneWeek .. Year])*

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

COMPUTE IF: *QAccomdat.Tenure IN [Part .. Squatting]*
    AND: *HBenefit = Yes*
    AND: *HBenAmt > 0*
    AND: *NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])*

PWeekly := 0

FRS0605B.QRenting (continued)

Questions about renters

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
     AND: HBenPd IN [OneWeek .. Year]
     AND: LWeekly1 >= 0.01
```

HBenWkly := LWeekly1

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
     AND: HBenPd IN [OneWeek .. Year]
     AND: LWeekly1 >= 0.01
     AND: Edit = No
     (HBenWkly < 200) AND INVOLVING(HBenPd,HBenAmt)
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
     AND: HBenPd IN [OneWeek .. Year]
     AND: LWeekly1 >= 0.01
     AND: Edit = Yes
     AND: (HBenWkly = RentWkly) OR (Rent = HBenAmt)
     ERROR AND INVOLVING(Rent,HBenAmt)
```

^I Editor: The respondent has given exactly the same figure for rent and housing benefit. Please check that there is no double counting.^I

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
     AND: Rent > 0
```

FRS0605B.QRenting.HBenChk

^I QRenting
^I
^N Can I just check, is the amount of ^rent for rent that you mentioned earlier, BEFORE or AFTER taking off the Housing Benefit?^N

(1)	Befor	Before
(2)	Aftr	After

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBenAmt > 0
     AND: Rent > 0
     AND: ((HBenWkly = RESPONSE) AND (RentWkly = RESPONSE)) AND (HBenChk =
     Befor)
     (HBenWkly <= RentWkly) AND INVOLVING(HBenAmt,Rent)
```

^I Housing Benefit is not normally more than rent. However from October 2003 in some areas Housing Benefit may exceed rent. Please double check the figure with the respondent.^I

Ask if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *HBenefit = Yes*
And: *(HBenAmt = DONTKNOW) AND ((Rent = DONTKNOW) OR (Rent > 0))*

FRS0605B.QRenting.RentFull

^I QRenting
^I
^N How much is your FULL rent - that is, BEFORE Housing Benefit or Rent ^Allowance?^N

0.00..999997.00

Record if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *HBenefit = Yes*
And: *RentFull > 0*

FRS0605B.QRenting.RentPx1

^I QRenting
^I
^I^BLU^IC ^Pd97Txt^I

OPEN

Ask if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *HBenefit = Yes*
And: *RentFull > 0*

FRS0605B.QRenting.RentPd1

^I QRenting
^I
^N How long does this cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

Ask if: *QAccomdat.Tenure IN [Part .. Squatting]*
And: *HBenefit = Yes*
And: *RentFull > 0*
And: *RentPd1 = Note*

FRS0605B.QRenting.RentPx1

^I QRenting
^I
^I^BLU^IC ^Pd97Txt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentFull > 0
     AND: Edit = Yes
     RentPd1 <> Note
```

^I Editor: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^I

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
```

FRS0605B.QRenting.HBWeeks

^I QRenting
^I
^N For how long have you been on Housing Benefit or Rent ^Allowance (this time)?^N

(1)	Less2Y	Up to 2 years
(2)	Less3Y	2 years but less than 3
(3)	Less4Y	3 years but less than 4
(4)	Less5Y	4 years but less than 5
(5)	More5Y	5 or more years

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBWeeks = Less2Y
```

FRS0605B.QRenting.HBWeeks2

^I QRenting
^I
^N Please tell me how many weeks you have been on Housing Benefit or Rent ^Allowance (this time)?^N

0..997

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: HBWeeks = Less2Y
     AND: HBWeeks2 = RESPONSE
     IN(HBWeeks2,[1..104])
```

Enter a value between 1 and 104

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
```

FRS0605B.QRenting.EligAmt

^I QRenting
^I
^N On the (rent book/ card/ statement), what is the amount shown for eligible rent?^N

^I^BLU^IC This must be the ^B eligible rent ^B (may not be same as the amount of benefit)
Eligible rent = after deductions.^I

1.00..1000.00

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
RECORD IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

FRS0605B.QRenting.EligPx

^I QRenting
^I
^I^BLU^IC ^Pd97Txt^I

OPEN

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

FRS0605B.QRenting.EligPd

^I QRenting
^I
^N What period does that cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
     AND: EligPd = Note
```

FRS0605B.QRenting.EligPx

^I QRenting
^I
^I^BLU^IC ^Pd97Txt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
     AND: Edit = Yes
     EligPd <> Note
```

^I Editor: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^I

FRS0605B.QRenting.Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0

PdConW[1] := 1
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0

PdConW[2] := 2
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0

PdConW[3] := 3
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0

PdConW[4] := 4
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0

PdConW[5] := 4.333
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0

PdConW[7] := 8.67
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0

PdConW[8] := 6.5
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0

PdConW[9] := 5.78
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0605B.QRenting (continued)

Questions about renters

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
     AND: EligPd IN [OneWeek .. Year]
     AND: LWeekly1 >= 0.01
```

EligWkly := LWeekly1

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = Yes
     AND: RentDoc = HBStmt
     AND: EligAmt > 0
     AND: EligPd IN [OneWeek .. Year]
     AND: LWeekly1 >= 0.01
     AND: Edit = No
     (EligWkly < 200) AND INVOLVING(EligPd,EligAmt)
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: HBenefit = No
```

FRS0605B.QRenting.HBenWait

^I QRenting
^I
^N Are you awaiting the outcome of a claim for Housing Benefit - that is, either Rent Rebate or Rent Allowance?^N

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
```

FRS0605B.QRenting.WSInc

^I QRenting
^I
^N Were water or sewerage charges (rates) included in the rent which you mentioned?^N^I^Consult_the_document^I

(1)	Both	Both water & sewerage
(2)	Water	Water only
(3)	Sewer	Sewerage only
(4)	Neith	Neither

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: WSInc = Both
```

COMBINED_AMOUNT := '

Enter combined amount.'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: WSInc = Water
```

water_sewerage := 'water'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: WSInc = Sewer
```

water_sewerage := 'sewerage'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (WSInc = Sewer)
```

water_sewerage := 'water/sewerage'

FRS0605B.QRenting.PdTxt1()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]

LInThat := ' in that '
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = OneWeek

PPdTxt := 'one week period'
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = TwoWeek

PPdTxt := 'two week period'
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = ThrWeek

PPdTxt := 'three week period'
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = Fourweek

PPdTxt := 'four week period'
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = Month

PPdTxt := 'calendar month'
```

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = ThrMonth
```

PPdTxt := 'three month period'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = SixMonth
```

PPdTxt := 'six month period'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [OneWeek .. Month, ThrMonth .. Year]
     AND: PPeriod = Year
```

PPdTxt := 'year'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [TwoMonth .. TenYear]
```

LInThat := ' in those '

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [TwoMonth .. TenYear]
     AND: PPeriod = TwoMonth
```

PPdTxt := 'two calendar months'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [TwoMonth .. TenYear]
     AND: PPeriod = EighYear
```

PPdTxt := 'eight payments'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [TwoMonth .. TenYear]
     AND: PPeriod = NineYear
```

PPdTxt := 'nine payments'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod IN [TwoMonth .. TenYear]
     AND: PPeriod = TenYear
```

PPdTxt := 'ten payments'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod = LessWeek
```

LInThat := ' in that '

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: PPeriod = LessWeek
```

PPdTxt := 'week'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: NOT (PPeriod = LessWeek)
```

LInThat := ''

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
     AND: NOT (PPeriod = LessWeek)
```

PPdTxt := ''

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd IN [OneWeek .. LessWeek]
```

PPdTxt := (LInThat + PPdTxt)

FRS0605B.QRenting (continued)

Questions about renters

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd IN [OneWeek .. LessWeek])
```

in_that_period := ''

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
```

FRS0605B.QRenting.WSIncAmt

^I QRenting
^I
^N How much was included for ^water_sewerage ^in_that_period?^N
^I^BLU^IC ^COMBINED_AMOUNT^Consult_the_document^I

1.00..100.00

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: Rent >= 0
     NOT(WSIncAmt > Rent)
```

^I The amount included in rent for water/sewerage is greater than the rent!^I

```
RECORD IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: WSIncAmt > 0
```

FRS0605B.QRenting.RentPx2

^I QRenting
^I
^I^BLU^IC ^Pd97Txt^IC

OPEN

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
 AND: *(PTenure IN [Rents, Part]) AND Rent <> EMPTY*
 AND: *(Scotland <> Yes) AND (NatCen <> NI)*
 AND: *WSInc IN [Both, Water, Sewer]*
 AND: *WSIncAmt > 0*

FRS0605B.QRenting.RentPd2

^I QRenting
^I
^N How long did this cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
 AND: *(PTenure IN [Rents, Part]) AND Rent <> EMPTY*
 AND: *(Scotland <> Yes) AND (NatCen <> NI)*
 AND: *WSInc IN [Both, Water, Sewer]*
 AND: *WSIncAmt > 0*
 AND: *RentPd2 = Note*

FRS0605B.QRenting.RentPx2

^I QRenting
^I
^I^BLU^IC ^Pd97Txt^IC

OPEN

FRS0605B.QRenting.Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: RentPd2 = RESPONSE
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0605B.QRenting.Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: NOT (RentPd2 = RESPONSE)
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0605B.QRenting (continued)

Questions about renters

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: (WSIncAmt > 0) AND ((RentPd IN [OneWeek .. Year]) OR (RentPd2 IN
     [OneWeek .. Year]))
     AND: LWeekly1 >= 1
```

WSIWkly := LWeekly1

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: (WSIncAmt > 0) AND ((RentPd IN [OneWeek .. Year]) OR (RentPd2 IN
     [OneWeek .. Year]))
     AND: LWeekly1 >= 1
     AND: Edit = No
     (WSIWkly < 20) AND INVOLVING(WSIncAmt)
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: (Scotland <> Yes) AND (NatCen <> NI)
     AND: WSInc IN [Both, Water, Sewer]
     AND: WSIncAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: (PTenure IN [Rents, Part]) AND Rent <> EMPTY
     AND: Rent <> 0
```

FRS0605B.QRenting.SerInc

^I QRenting
^I
^IS^I M^I
^N Does the rent which you mentioned include any of the services shown on this card?^N^I^Consult_the_document^I

^I^BLU^IC code all that apply.^I

SET [5] OF

(1)	Heating	Heating
(2)	Lighting	Lighting
(3)	HotWat	Hot water
(4)	FuelCook	Fuel for cooking
(5)	TVLic	TV licence fees
(6)	None	None of these services

CHECK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *(PTenure IN [Rents, Part]) AND Rent <> EMPTY*
AND: *Rent <> 0*
AND: *None IN SerInc*
SerInc.CARDINAL = 1

^I 'None of these' is an exclusive code for this question.^I

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *(PTenure IN [Rents, Part]) OR ((PTenure IN [RentFree, Squatting]) AND (AccJob <> Yes))*

FRS0605B.QRenting.AccNonHH

^I QRenting
^I
^N (Apart from Housing Benefit) does anyone outside your household pay any rent on this accommodation on your behalf?^N

^I^BLU^IC Exclude Housing Benefit - ie. Rent Rebate or Rent Allowance.^I

(1)	Yes	Yes
(2)	No	No

ASK IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *AccNonHH = Yes*

FRS0605B.QRenting.AccPay

^I QRenting
^I
^N Who is that?^N

^I^BLU^IC Code all that apply.^I

SET [5] OF

(1)	GOV	^GOVSSA
(2)	Emp	Employer
(3)	Org	Other organisation
(4)	Rel	Friend or relative
(5)	Other	Other

WARN IF: *QAccomdat.Tenure IN [Part .. Squatting]*
AND: *AccNonHH = Yes*
NOT(IN(GOV,AccPay))

^I Are you sure? ^GOV1 only ever pay ^B arrears ^B of rent. Double-check, that respondent is not thinking of Housing Benefit.
If genuine arrears, suppress this warning.^I

FRS0605B.QRenting.QAccPay[]

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

Payer[1] := GOV1

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

Payer[2] := 'employer'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

Payer[3] := 'other organisation'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

Payer[4] := 'relative or friend'

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

Payer[5] := '

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

LRent := PRent

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

RentSeq := PSeq

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

AccPay := PSeq

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
```

FRS0605B.QRenting.QAccPay[].AccAmt

^I QRenting
^I

^N How much rent did the ^Payer[AccPay] pay for you last time?^N

0.01..999997.00

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt = NONRESPONSE
```

```
HMissVar := (HMissVar + 1)
```

```
RECORD IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

FRS0605B.QRenting.QAccPay[].AccPx

^I QRenting
^I

^I^BLU^IC ^Pd97Txt^I

OPEN

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

FRS0605B.QRenting.QAccPay[].AccPd

^I QRenting
^I

^N How long did that cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
     AND: AccPd = Note
```

FRS0605B.QRenting.QAccPay[].AccPx

^I QRenting
^I

^I^BLU^IC ^Pd97Txt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
     AND: Edit = Yes
     AccPd <> Note
```

^I Editor: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^I

FRS0605B.QRenting.QAccPay[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0605B.QRenting.QAccPay[] (continued)

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
     AND: AccPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
```

AccWkly := LWeekly

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: AccAmt > 0
     AND: AccPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
     AND: Edit = No
     (AccWkly < 151) AND INVOLVING(AccPd,AccAmt)
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
ASK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: PRent > 0
```

FRS0605B.QRenting.QAccPay[].AccChk

^I QRenting
^I

^N Can I just check, is the amount of £^LRent for rent, that you mentioned earlier, BEFORE or AFTER deducting this payment?^N

(1)	Befor	Before
(2)	Aftr	After

FRS0605B.QRenting (continued)

Questions about renters

```
COMPUTE IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: ((QAccPay[Index].AccWkly = RESPONSE) AND (RentWkly = RESPONSE))
     AND (QAccPay[Index].AccChk = Befor)
```

AccC := (AccC + QAccPay[Index].AccWkly)

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     AND: Index IN AccPay
     AND: ((QAccPay[Index].AccWkly = RESPONSE) AND (RentWkly = RESPONSE))
     AND (QAccPay[Index].AccChk = Befor)
     (AccC <= RentWkly) AND
     INVOLVING(QAccPay[Index].AccPd,QAccPay[Index].AccAmt)
```

^I The amount recorded for help with your rent is greater than the rent recorded.^I

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: AccNonHH = Yes
     AND: In loop FOR Index := 1 TO 5
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Edit = Yes
     AND: Rent = NONRESPONSE OR RentPd = NONRESPONSE
     NOT(IN(Landlord,[???])) AND
     INVOLVING(QAccomdat.Rooms,QAccomdat.TypeAcc)
```

^I
Missing information for rent amount or period.
Follow edit instructions for 'Rent'^I

```
CHECK IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: Edit = Yes
     AND: HBenAmt = NONRESPONSE OR (HBenPd = NONRESPONSE AND (HBenefit =
     Yes))
     NOT(IN(Landlord,[???]))
```

^I
Missing information FOR Housing Benefit amount and/or period.
Follow edit instructions for 'Housing Benefit'^I

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     AND: QAccomdat.Tenure = RentFree
     IN(QRenting.Landlord,[???])
```

^I It is very unusual for Local Authority or Housing Association tenants to be living rent-free. Please check with respondent. Change 'Tenure' to renting if 100% Housing Benefit received, or somebody else pays the rent.^I

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Part .. Squatting]
     RESERVECHECK
```

RESERVECHECK

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: In loop FOR Loop1 := 1 TO HHSize
     AND: Loop1 IN QHholder.HHldr
```

```
Oldest := (MAX(DMAge[Loop1],Oldest))
```

FRS0605B.QOwner1

Questions about mortgages

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
```

FRS0605B.QOwner1.BuyYear

^I QOwner1
^I

^I This should be the year^B this^B property was bought. Even if the respondent states that the current mortgage was 'carried over' from a previous property, enter the purchase date for^B this^B property - not the previous one.^I

1901..2007

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: (BuyYear = RESPONSE) AND (POldest > 0)
     BuyYear >= (DLYear.YEAR - POldest)
```

^I This is before the date of birth of the oldest householder. Please check your figures.^I

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: QDataBag.SampMnth IN [4 .. 12]
     BuyYear <> 2007
```

^I Wrong Year!^I

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
```

YearLive := ORD(QAccomdat.YearLive)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])
     AND: YearLive = 4
```

YearLive := 5

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])
     AND: YearLive = 5
```

YearLive := 10

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])
     AND: YearLive = 6
```

YearLive := 20

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])
```

Time[1] := '12 months'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])
```

Time[2] := '2 years'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])
```

Time[3] := '3 years'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])
```

Time[5] := '5 years'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])
```

Time[10] := '10 years'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])
```

Time[20] := '20 years'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])
```

MorgYear := (QSignIn.StartDat.YEAR - BuyYear)

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: (BuyYear = RESPONSE) AND (YearLive IN [1 .. 6])
     (YearLive >= MorgYear) AND INVOLVING(QAccomdat.YearLive,BuyYear)
```

^I The respondent has lived here for less than ^Time[YearLive], but the^B mortgage^B started in ^BuyYear - ^MorgYear years ago. Please check that BuyYear is when the mortgage on THIS PROPERTY was taken out. (If so, suppress & continue)^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
```

FRS0605B.QOwner1.PurcLoan

^I QOwner1
^I
^N Can I just check, did you take out one loan to purchase this accommodation, or more than one?^N

(1)	One	One
(2)	Two	Two (or more) loans for purchase

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: PTenure = Part
```

your_share_in := ' your share in'

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
```

FRS0605B.QOwner1.PurcAmt

^I QOwner1
^I
^N What was the purchase price of^your_share_in your house/flat?^N

-99999999.99..999999999.99

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: PurcAmt = RESPONSE
     PurcAmt >= 0
```

^I^BLU^IC Please enter a positive amount (>=0)^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: Edit = No
     PurcAmt < 500000
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Mortgage) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     StillM))
     AND: PurcAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: (PTenure = Outright) OR ((PTenure = Part) AND (QAccomdat.SOBuy =
     Paid))
```

FRS0605B.QOwner1.OthMort3

^I QOwner1
^I
^N May I just check, are you currently using ^B this house/flat ^B as security for a mortgage or loan of any kind?^N

(1)	Yes	Yes
(2)	No	No

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Outright) OR ((PTenure = Part) AND (QAccomdat.SOBuy = Paid))*

FRS0605B.QOwner1.OPur3Rs

^I QOwner1
^I
^I This should only apply to loans for purchase. Please resolve, or make a Note.^I

(1)	Passed	Passed
(2)	Hard	Hard
(3)	Soft	Soft
(4)	Suppress	Suppressed

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Outright) OR ((PTenure = Part) AND (QAccomdat.SOBuy = Paid))*

FRS0605B.QOwner1.OPur3Ex

^I QOwner1
^I
^I^BLU^IC^SuppTxt^I

OPEN

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Outright) OR ((PTenure = Part) AND (QAccomdat.SOBuy = Paid))*
AND: *OthMort3 = Yes*

FRS0605B.QOwner1.OthPur3

^I QOwner1
^I
^IS^I N^I

^N Which of these items best describe the reasons why you took out this other loan? ^Any_others?^N

^I^BLU^IC Code all that apply.^I

SET [7] OF

(1)	Improve	To make improvements or extensions to this property
(2)	Purcase	To help purchase a major item like a car, boat, caravan or second home
(3)	IntrRate	To get a better, or fixed, interest rate
(4)	Business	In connection with a business
(5)	BuyOut	To buy out another person's share in the property
(6)	Repairs	For essential repairs to make the property fit for occupation
(7)	Other	Some other purpose ^I(Specify in a note.)^I

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Outright) OR ((PTenure = Part) AND (QAccomdat.SOBuy = Paid))*
AND: *OthMort3 = Yes*
NOT(IN(IntrRate,OthPur3))

NOT (IntrRate IN OthPur3)

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *(PTenure = Outright) OR ((PTenure = Part) AND (QAccomdat.SOBuy = Paid))*
AND: *OthMort3 = Yes*
AND: *(OPur3Rs = Suppressed) OR OPur3Ex <> EMPTY*

FRS0605B.QOwner1.OPur3Ex

^I QOwner1
^I
^I^BLU^IC^SuppTxt^I

OPEN

FRS0605B.QOwner1.QMortgage.M[]

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq IN [1 .. 2]
     AND: PPTenure = Part
```

to_buy_this_house := ' to buy your share in this house/flat'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq IN [1 .. 2]
     AND: NOT (PPTenure = Part)
```

to_buy_this_house := ' to buy this house/flat'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq IN [1 .. 2]
```

fill := (', in ' + STR(PBuyYear))

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq = 3
```

to_buy_this_house := ' for essential repairs'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq = 1
```

MORTGAGE := (B + ' main ' + B + ' mortgage
@|for ' + 'the purchase of this accommodation.')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq = 1
```

INSTRUC := ('

@|(Questions about any other, ' + 'separate mortgage will follow.)')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq = 2
```

MORTGAGE := (B + ' second ' + B + ' mortgage
@|for ' + 'the purchase of this accommodation.')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq = 2
```

INSTRUC := ('

@|(Questions about any other, ' + 'separate mortgage will follow.)

')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: PSeq = 3
```

MORTGAGE := 'loan for essential repairs'

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
```

FRS0605B.QOwner1.QMortgage.M[].IntroM

^I QOwner1
^I

^I
@|The next questions are about the ^MORTGAGE ^INSTRUC^I

(1)	Cont	Press <Enter> to continue.

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
```

MortSeq := PSeq

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: PSeq = 2
```

FRS0605B.QOwner1.QMortgage.M[].Loan2Y

^I QOwner1
^I

^I^BLU^IC Check: Do they still have this other mortgage for purchase? (If now repaid, use code 2)^I

(1)	Still	Yes, still have this mortgage
(2)	Repaid	No, mortgage has been repaid

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *PSeq = 3*

FRS0605B.QOwner1.QMortgage.M[].LoanYrRs

^I QOwner1
^I

^I^BLU^IC You have entered that the respondent took out a second loan on this property before they purchased it. Please check your answer. You MUST make a note if you suppress this check.^I

(1)	Passed	Passed
(2)	Hard	Hard
(3)	Soft	Soft
(4)	Suppress	Suppressed

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *PSeq = 3*

FRS0605B.QOwner1.QMortgage.M[].LoanYrEx

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *PSeq = 3*

FRS0605B.QOwner1.QMortgage.M[].LoanYear

^I QOwner1
^I

^N In which year did you take out this mortgage or loan?^N

1901..2007

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PSeq = 3
     AND: QDataBag.SampMnth IN [4 .. 12]
     LoanYear <> 2007
```

Wrong Year!

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PSeq = 3
     AND: (Edit <> Yes) AND ((LoanYear > 0) AND (BuyYear > 0))
     BuyYear <= LoanYear
```

BuyYear <= LoanYear

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PSeq = 3
     AND: (Edit <> Yes) AND ((LoanYear > 0) AND (BuyYear > 0))
     AND: (LoanYrRs = Suppressed) OR LoanYrEx <> EMPTY
```

FRS0605B.QOwner1.QMortgage.M[].LoanYrEx

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid

LPurcAmt := PPurcAmt
```

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
```

FRS0605B.QOwner1.QMortgage.M[].BorAmtRs

^I QOwner1
^I

^I^BLU^IC The amount borrowed is more than the purchase price - this is very unusual. Please check your figures and, if necessary, explain in a Note.^I

(1)	Passed	Passed
(2)	Hard	Hard
(3)	Soft	Soft
(4)	Suppress	Suppressed

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
```

FRS0605B.QOwner1.QMortgage.M[].BorAmtEx

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
```

FRS0605B.QOwner1.QMortgage.M[].BorrAmt

^I QOwner1
^I

^I
This should be the^B original^B amount of this mortgage, as taken out when the property was purchased (in 'BuyYear').

^B Properties/mortgages partly for business^B: If the amount borrowed includes the purchase of non-domestic accommodation or land, eg. a farm, a shop with flat above, try to obtain purchase and mortgage details for the domestic element only.^I

-99999999.99..999999999.99

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: BorrAmt = RESPONSE
     BorrAmt >= 0
```

^I^BLU^IC Please enter a positive amount. Negative amounts (eg. -10) are not allowed.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((Edit <> Yes) AND (BorrAmt > 0)) AND (LPurcAmt > 0)
     BorrAmt <= LPurcAmt
```

BorrAmt <= LPurcAmt

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((Edit <> Yes) AND (BorrAmt > 0)) AND (LPurcAmt > 0)
     AND: (BorAmtRs = Suppressed) OR BorAmtEx <> EMPTY
```

FRS0605B.QOwner1.QMortgage.M[].BorAmtEx

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: BorrAmt = RESPONSE
     BorrAmt < 250000
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: BorrAmt = RESPONSE
     BorrAmt >= 500
```

^I That seems very low - please check your figures.^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: (PSeq = 1) AND (BorrAmt = DONTKNOW)
```

FRS0605B.QOwner1.QMortgage.M[].BorAmtDK

^I QOwner1
^I

^I^BLU^IC Is this 'Don't know' because the^B original^B mortgage was to buy domestic accommodation^B and^B for business purposes, and you cannot get a separate figure for the^B domestic^B part?^I

(1)	Yes	Yes (Please give full details in a Note)
(2)	No	No

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: BorrAmt = REFUSAL
```

HMissVar := (HMissVar + 1)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: (((PSeq = 1) AND (BorrAmt = DONTKNOW)) AND (BorAmtDK <> Yes)) OR
     ((PSeq <> 1) AND (BorrAmt = DONTKNOW))
```

HMissVar := (HMissVar + 1)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
```

FRS0605B.QOwner1.QMortgage.M[].RMort

^I QOwner1
^I

^I 'Re-mortgage': a new mortgage is taken out and is used to REPAY an existing mortgage on a property. Typically this happens when a new lender offers better terms, eg. a lower rate of interest. The new loan can be bigger than the old one.

'Further advance or top-up': the loan is EXTENDED (eg. from £30,000 to £40,000). But, if there are SEPARATE payments to cover the new sum borrowed, this counts as a separate loan - NOT a further advance/top-up. Separate loans are dealt with later, at 'OthMort1'.^I

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = Yes
```

FRS0605B.QOwner1.QMortgage.M[].RMortYr

^I QOwner1
^I

^N In which year did you take out the most recent re-mortgage/further advance?^N

1901..2007

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = Yes
     AND: QDataBag.SampMnth IN [4 .. 12]
     RMortYr <> 2007
```

^I Wrong Year!^I

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = Yes
     AND: (RMortYr = RESPONSE) AND (PBuyYear = RESPONSE)
     RMortYr >= PBuyYear
```

^I The first mortgage was taken out in ^PBuyYear, so the re-mortgage can't have been taken out before that. Please amend your answers.^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = Yes
```

FRS0605B.QOwner1.QMortgage.M[].RMAmt

^I QOwner1
^I

^N What was the total amount of the mortgage, after re-mortgaging/taking out the further advance?^N

^I^BLU^IC Total should be after all re-mortgages and further advances.^I

-99999999.99..999999999.99

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = Yes
     AND: RMAmt = RESPONSE
     RMAmt >= 0
```

^I^BLU^IC Please enter a positive amount. Negative amounts (eg. -10) are not allowed.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = Yes
     AND: Edit = No
     RMAmt < 250000
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *RMort = Yes*
AND: *RMAmt = NONRESPONSE*

HMissVar := (HMissVar + 1)

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *RMort = Yes*

FRS0605B.QOwner1.QMortgage.M[].RMPur

^I QOwner1
^I

^IS^I H^I
^N Which of these items best describe the reasons why you took out a re-mortgage/ further advance?
^Any_others?^N
^I^BLU^IC Code all that apply.^I

SET [8] OF

(1)	Improve	To make improvements or extensions to this property
(2)	Purcase	To help purchase a major item like a car, boat, caravan or second home
(3)	IntrRate	To get a better, or fixed, interest rate
(4)	Business	In connection with a business
(5)	BuyOut	To buy out another person's share in the property
(6)	Repairs	For essential repairs to make the property fit for occupation
(7)	Flexible	To move to a more flexible mortgage
(8)	Other	Some other purpose (SPECIFY IN A NOTE.)

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*

FRS0605B.QOwner1.QMortgage.M[].MortTyEx

^I QOwner1
^I

^I^MortTTxt

^SuppTxt^I

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
```

FRS0605B.QOwner1.QMortgage.M[].MortType

^I QOwner1
^I

^IS^I I^I
^N Looking at this card, please tell me which of these options describe your mortgage or loan?^N
^I^BLU^IC If necessary add 'With a repayment mortgage, by repaying the original loan we mean the original capital sum borrowed.'^I

(1)	Endow	an ENDOWMENT mortgage (where your mortgage payments cover interest only)
(2)	Repay	a REPAYMENT mortgage (where your mortgage payments cover interest and part of the original loan)
(3)	Pension	a PENSION mortgage (where your mortgage payments cover interest only)
(4)	PEP	a PEP, Unit Trust or ISA mortgage
(5)	EndRep	both an endowment (or other interest only) AND a repayment mortgage
(6)	IntLink	an interest only mortgage with more than one linked investment (e.g. pension and unit trust, endowment and ISA)
(7)	IntNoLnk	an interest only mortgage with NO linked investment (e.g. NO endowment, pension, PEP or ISA)
(8)	Other	or another type (not listed above)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: MortType = NONRESPONSE
```

MortTTxt := KeyTxt

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: MortType = NONRESPONSE
     ERROR
```

^MortTTxt

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: MortType = NONRESPONSE
```

FRS0605B.QOwner1.QMortgage.M[].MortTyEx

^I QOwner1
^I

^I^MortTTxt

^SuppTxt^I

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: MortType = Other
```

MortTTxt := ('Interviewer: This is a rather unusual sort of ' + 'mortgage. Are you sure that it is not covered ' + 'by one of the codes above? If it is not, please ' + 'record the title in a note and answer follow-up ' + 'questions as fully as possible.')

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: MortType = Other
     ERROR
```

^I^MortTTxt^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: Edit = No
     AND: MortType = Other
```

FRS0605B.QOwner1.QMortgage.M[].MortTyEx

^I QOwner1
^I

^I^MortTTxt

^SuppTxt^I

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType = Endow
```

this_kind_of := 'an endowment'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType = Repay
```

this_kind_of := 'a repayment'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType = Pension
```

this_kind_of := 'a pension'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType = PEP
```

this_kind_of := 'a PEP, Unit Trust or ISA'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType = EndRep
```

this_kind_of := 'an endowment & repayment'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [IntLink, IntNoLnk]
```

this_kind_of := 'an interest only'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT (MortType IN [IntLink, IntNoLnk])
```

this_kind_of := 'this kind of'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: MortType IN [Endow, EndRep]
```

Is_the := 'Apart from the endowment I mentioned earlier is the'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: NOT (MortType IN [Endow, EndRep])
```

Is_the := 'Is the'

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
```

FRS0605B.QOwner1.QMortgage.M[].EndwPrin

^I QOwner1
^I

^IS^I J^I
^N^Is_the repayment of the original loan covered by any of the things on this card?^N
^I^BLU^IC Code all that apply.^I

SET [4] OF

(1)	Pension	Current payments into a Pension Plan (pension mortgage)
(2)	PEP	Current payments into a PEP or ISA
(3)	UnitT	Current payments into a Unit Trust or Investment Trust scheme
(4)	OthSch	Current payments into any^B other^B savings/investment scheme
(5)	HseSale	Proceeds of sale from existing house only
(6)	None	None of the above.

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: None IN EndwPrin
     EndwPrin.CARDINAL = 1
```

^I None is an exclusive code for this question.^I

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     RESERVECHECK
```

RESERVECHECK

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
```

FRS0605B.QOwner1.QMortgage.M[].EndwP1Ex

^I QOwner1
^I

^I Please leave a note to describe how the respondent will be repaying their mortgage.

^SuppTxt^I

OPEN

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
```

FRS0605B.QOwner1.QMortgage.M[].EndwP2Ex

^I QOwner1
^I

^I^EndP2Txt

^SuppTxt^I

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
```

```
EndwPTxt := ('This method of capital repayment does not match
the ' + 'type of mortgage recorded earlier at MortType. ' + '
Please resolve, or make a note.')
```

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
  AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
  AND: *In loop FOR ii := 1 TO 3*
  AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
  AND: *Loan2Y <> Repaid*
  AND: *MortType IN [Endow, Pension .. Other]*
  AND: *Edit = No*
  AND: *None IN EndwPrin*
  AND: *NOT (MortType IN [Endow, EndRep])*
  ERROR AND INVOLVING(MortType,EndwPrin)

^I Please leave a note to describe how the respondent will be repaying their mortgage.^I

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
  AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
  AND: *In loop FOR ii := 1 TO 3*
  AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
  AND: *Loan2Y <> Repaid*
  AND: *MortType IN [Endow, Pension .. Other]*
  AND: *Edit = No*
  AND: *None IN EndwPrin*
  AND: *NOT (MortType IN [Endow, EndRep])*

FRS0605B.QOwner1.QMortgage.M[].EndwP1Ex

^I QOwner1
^I

^I Please leave a note to describe how the respondent will be repaying their mortgage.

^SuppTxt^I

OPEN

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
  AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
  AND: *In loop FOR ii := 1 TO 3*
  AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
  AND: *Loan2Y <> Repaid*
  AND: *MortType IN [Endow, Pension .. Other]*
  AND: *Edit = No*
  AND: *MortType = Endow*
  AND: *(((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*

EndP2Txt := EndwPTxt

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Endow
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
     ERROR AND INVOLVING(MortType,EndwPrin)
```

^I^EndP2Txt^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Endow
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

FRS0605B.QOwner1.QMortgage.M[].EndwP2Ex

^I QOwner1
^I

^I^EndP2Txt

^SuppTxt^I

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Pension
     AND: (((PEP IN EndwPrin) OR (UnitT IN EndwPrin)) OR (OthSch IN
     EndwPrin)) OR (None IN EndwPrin)
```

EndP2Txt := EndwPTxt

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Pension
     AND: (((PEP IN EndwPrin) OR (UnitT IN EndwPrin)) OR (OthSch IN
     EndwPrin)) OR (None IN EndwPrin)
     ERROR AND INVOLVING(MortType,EndwPrin)
```

^I^EndP2Txt^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Pension
     AND: (((PEP IN EndwPrin) OR (UnitT IN EndwPrin)) OR (OthSch IN
     EndwPrin)) OR (None IN EndwPrin)
```

FRS0605B.QOwner1.QMortgage.M[].EndwP2Ex

^I QOwner1
^I

^I^EndP2Txt

^SuppTxt^I

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = PEP
     AND: ((Pension IN EndwPrin) OR (OthSch IN EndwPrin)) OR (None IN
     EndwPrin)
```

EndP2Txt := EndwPTxt

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = PEP
     AND: ((Pension IN EndwPrin) OR (OthSch IN EndwPrin)) OR (None IN
     EndwPrin)
     ERROR AND INVOLVING(MortType,EndwPrin)
```

^I^EndP2Txt^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = PEP
     AND: ((Pension IN EndwPrin) OR (OthSch IN EndwPrin)) OR (None IN
     EndwPrin)
```

FRS0605B.QOwner1.QMortgage.M[].EndwP2Ex

^I QOwner1
^I

^I^EndP2Txt

^SuppTxt^I

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = IntNoLnk
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

EndP2Txt := ('You described your mortgage as an interest ' + 'only with NO linked investments, can I just ' + 'check is this savings/investment scheme linked ' + 'to your mortgage? Please amend the answer at ' + 'MortType as appropriate.')

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = IntNoLnk
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
     ERROR AND INVOLVING(EndwPrin,MortType)
```

^I^EndP2Txt^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = IntNoLnk
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

FRS0605B.QOwner1.QMortgage.M[].EndwP2Ex

^I QOwner1
^I

^I^EndP2Txt

^SuppTxt^I

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Other
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

EndP2Txt := ('If there is a pension, ISA or other ' + 'savings/investment scheme to cover the ' + 'repayment of the original loan, please ' + 'amend the answer at MortType as appropriate.')

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Other
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
     ERROR AND INVOLVING(EndwPrin,MortType)
```

^I^EndP2Txt^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType = Other
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

FRS0605B.QOwner1.QMortgage.M[].EndwP2Ex

^I QOwner1
^I

^I^EndP2Txt

^SuppTxt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     RESERVECHECK
```

RESERVECHECK

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Pension IN EndwPrin
```

DMAEndwPrin := Pension

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*

payment1 := 'contribution to the'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*

payment2 := '(pension plan/PEP/ISA/Unit Trust)'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *MortType = IntLink*

payment1 := 'premium/payment on the'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *MortType = IntLink*

payment2 := ('endowment policy / pension / unit trust /' + 'ISA / PEP / investment trust / other policy')

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *NOT (MortType = IntLink)*

payment1 := 'premium on the'

Compute if: *QAccomdat.Tenure IN [Outright .. Part]*
  And: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
  And: *In loop FOR ii := 1 TO 3*
  And: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
  And: *Loan2Y <> Repaid*
  And: *NOT (MortType = IntLink)*

payment2 := 'endowment policy'

Compute if: *QAccomdat.Tenure IN [Outright .. Part]*
  And: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
  And: *In loop FOR ii := 1 TO 3*
  And: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
  And: *Loan2Y <> Repaid*
  And: *MortType = IntLink*

policy := ('policy / pension / unit trust / ISA / PEP / investment ' + 'trust / other policy')

Compute if: *QAccomdat.Tenure IN [Outright .. Part]*
  And: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
  And: *In loop FOR ii := 1 TO 3*
  And: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
  And: *Loan2Y <> Repaid*
  And: *NOT (MortType = IntLink)*

policy := 'policy'

Compute if: *QAccomdat.Tenure IN [Outright .. Part]*
  And: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
  And: *In loop FOR ii := 1 TO 3*
  And: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
  And: *Loan2Y <> Repaid*
  And: *MortType IN [Endow, Pension .. Other]*
  And: *MortType IN [Endow, EndRep]*

Are := '(Can I just check), are'

Compute if: *QAccomdat.Tenure IN [Outright .. Part]*
  And: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
  And: *In loop FOR ii := 1 TO 3*
  And: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
  And: *Loan2Y <> Repaid*
  And: *MortType IN [Endow, Pension .. Other]*
  And: *NOT (MortType IN [Endow, EndRep])*

Are := 'Are'

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
```

FRS0605B.QOwner1.QMortgage.M[].MenPol

^I QOwner1
^I

^N^Are there any endowment policies covering the repayment of this mortgage or loan?^N

(1)	Yes	Yes
(2)	No	No

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: Edit = No
     AND: MortType IN [Endow, EndRep]
     ((MenPol = Yes) OR (MortType = Other)) AND INVOLVING(MortType,MenPol)
```

^I Normally there^B would^B be an endowment policy, with an endowment mortgage: please check.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: MenPol = Yes
     NOT(IN(MortType,[???,???])) AND INVOLVING(MortType,MenPol)
```

^I You described your mortgage as an interest only with NO linked investments or another type of mortgage, can I just check is this savings/investment scheme linked to your mortgage? Please amend the answer at MortType as appropriate.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: MenPol = Yes
     (MortType <> Other) AND INVOLVING(MortType,MenPol)
```

^I If there is an endowment, pension, ISA or other arrangement to cover the repayment of the original loan, please amend the answer at MortType as appropriate.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortType IN [Endow, Pension .. Other]
     AND: MenPol = No
     (MortType <> Other) AND INVOLVING(MortType,MenPol)
```

^I It is unusual for there to be no policies to cover the repayment of the loan. Please check. If original loan is included in monthly payments, please amend your answer at MortType to a Repayment (code 2) OR Endowment and Repayment (code 5) mortgage.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: RMort = No
```

```
How_Long := ('How long is the term of your mortgage. By this
we mean ' + 'the agreed term?')
```

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: NOT (RMort = No)
```

```
How_Long := ('How long is the term of your mortgage. By this
we ' + 'mean the agreed term since you have remortgaged or ' +
'extended the original loan?')
```

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
```

FRS0605B.QOwner1.QMortgage.M[].MorFlc

^I QOwner1
^I

^N All-in-one accounts are a new type of flexible mortgage which allow a person to link together accounts - for example, a current account, a savings account and a mortgage (as well as other types of loans). There are two types of all-in-one account: current account mortgages and offset mortgages.^N

^I^BLU^IC Examples include the One account (RBoS), a Woolwich Open Plan or some other all-in-one account..^BLU^I ^N

Is your mortgage an all-in-one account?^N

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MorFlc = Yes
```

FRS0605B.QOwner1.QMortgage.M[].MorAll

^I QOwner1
^I

^IS^I K AND L^I

^N Is your all-in-one account mortgage, a current account mortgage or an offset mortgage?^N

(1)	Current	Current account mortgage
(2)	OffSet	Offset mortgage

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
```

FRS0605B.QOwner1.QMortgage.M[].MortEnd

^I QOwner1
^I

^N^How_Long^N

^I^BLU^IC If remortgaged agreed term is from the point of remortgaging.^I

1..60

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: RMort = No
```

```
remortgage := 'mortgage was taken out'
```

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT (RMort = No)
```

```
remortgage := 'last re-mortgage'
```

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     MortEnd <= 40
```

^I Are you sure? The end-date would not normally be more than 40 years after the ^remortgage. Please check your figures.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MorAll = Current
```

What_amount := ('What is the amount of the negative balance or ' + 'overdraft on your current account mortgage')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT (MorAll = Current)
```

What_amount := ('What is the amount still outstanding on your ' + 'mortgage/loan from this source - that is, how ' + 'much do you still have to pay off')

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
```

FRS0605B.QOwner1.QMortgage.M[].MortL1Rs

^I QOwner1
^I

^N For ^this_kind_of mortgage, the amount outstanding should equal the ^amount ^borrowed. Please check and amend, else explain in a Note.^N

(1)	Passed	Passed
(2)	Hard	Hard
(3)	Soft	Soft
(4)	Suppress	Suppressed

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
```

FRS0605B.QOwner1.QMortgage.M[].MortL1Ex

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*

FRS0605B.QOwner1.QMortgage.M[].MortL2Rs

^I QOwner1
^I

^N For ^this_kind_of mortgage, the amount outstanding should be less than the ^amount ^borrowed. Please check and amend, else explain in a Note.^N

(1)	Passed	Passed
(2)	Hard	Hard
(3)	Soft	Soft
(4)	Suppress	Suppressed

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*

FRS0605B.QOwner1.QMortgage.M[].MortL2Ex

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*

FRS0605B.QOwner1.QMortgage.M[].MortLeft

^I QOwner1
^I

^N^What_amount?^N

-99999999.99..999999999.99

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortLeft = RESPONSE
     MortLeft >= 0
```

^I^BLU^IC Please enter a positive amount. Negative amounts (eg. -10) are not allowed.^I

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: MortLeft = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP, Other]
     AND: RMort = Yes
```

borrowed := 'of the re-mortgage'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP, Other]
     AND: RMort = Yes
```

amount := 'total amount'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP, Other]
     AND: RMort = Yes
     (ABS(MortLeft - RMAmt) <= 50) AND INVOLVING(MortLeft)
```

(ABS (MortLeft - RMAmt) <= 50) AND INVOLVING (MortLeft)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP, Other]
     AND: RMort = Yes
     AND: (MortL1Rs = Suppressed) OR MortL1Ex <> EMPTY
```

FRS0605B.QOwner1.QMortgage.M[].MortL1Ex

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP, Other]
     AND: NOT (RMort = Yes)
```

borrowed := 'originally borrowed'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP, Other]
     AND: NOT (RMort = Yes)
```

amount := 'amount'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Endow, Pension, PEP, Other]
     AND: NOT (RMort = Yes)
     (ABS(MortLeft - BorrAmt) <= 50) AND INVOLVING(MortLeft)
```

(ABS (MortLeft - BorrAmt) <= 50) AND INVOLVING (MortLeft)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
    AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
    AND: In loop FOR ii := 1 TO 3
    AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
    Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
    (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
    AND: Loan2Y <> Repaid
    AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
    AND: MortType IN [Endow, Pension, PEP, Other]
    AND: NOT (RMort = Yes)
    AND: (MortL1Rs = Suppressed) OR MortL1Ex <> EMPTY
```

FRS0605B.QOwner1.QMortgage.M[].MortL1Ex

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
    AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
    AND: In loop FOR ii := 1 TO 3
    AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
    Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
    (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
    AND: Loan2Y <> Repaid
    AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
    AND: MortType IN [Repay, EndRep]
    AND: RMort = Yes
```

borrowed := 'of re-mortgage'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
    AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
    AND: In loop FOR ii := 1 TO 3
    AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
    Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
    (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
    AND: Loan2Y <> Repaid
    AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
    AND: MortType IN [Repay, EndRep]
    AND: RMort = Yes
```

amount := 'amount'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
    AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
    AND: In loop FOR ii := 1 TO 3
    AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
    Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
    (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
    AND: Loan2Y <> Repaid
    AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
    AND: MortType IN [Repay, EndRep]
    AND: RMort = Yes
    (MortLeft < RMAmt) AND INVOLVING(MortLeft)
```

(MortLeft < RMAmt) AND INVOLVING (MortLeft)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Repay, EndRep]
     AND: RMort = Yes
     AND: (MortL2Rs = Suppressed) OR MortL2Ex <> EMPTY
```

FRS0605B.QOwner1.QMortgage.M[].MortL2Ex

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Repay, EndRep]
     AND: NOT (RMort = Yes)
```

borrowed := 'originally borrowed'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Repay, EndRep]
     AND: NOT (RMort = Yes)
```

amount := 'amount'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Repay, EndRep]
     AND: NOT (RMort = Yes)
     (MortLeft < BorrAmt) AND INVOLVING(MortLeft)
```

(MortLeft < BorrAmt) AND INVOLVING (MortLeft)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [Repay, EndRep]
     AND: NOT (RMort = Yes)
     AND: (MortL2Rs = Suppressed) OR MortL2Ex <> EMPTY
```

FRS0605B.QOwner1.QMortgage.M[].MortL2Ex

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [IntLink, IntNoLnk]
     AND: RMort = Yes
```

borrowed := 'of the re-mortgage'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [IntLink, IntNoLnk]
     AND: RMort = Yes
```

amount := 'amount'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [IntLink, IntNoLnk]
     AND: RMort = Yes
     (MortLeft = RMAmt) AND INVOLVING(MortLeft)
```

(MortLeft = RMAmt) AND INVOLVING (MortLeft)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [IntLink, IntNoLnk]
     AND: RMort = Yes
     AND: (MortL1Rs = Suppressed) OR MortL1Ex <> EMPTY
```

FRS0605B.QOwner1.QMortgage.M[].MortL1Ex

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [IntLink, IntNoLnk]
     AND: NOT (RMort = Yes)
```

borrowed := 'originally borrowed'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [IntLink, IntNoLnk]
     AND: NOT (RMort = Yes)
```

amount := 'amount'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)
     AND: MortType IN [IntLink, IntNoLnk]
     AND: NOT (RMort = Yes)
     (MortLeft = BorrAmt) AND INVOLVING(MortLeft)
```

(MortLeft = BorrAmt) AND INVOLVING (MortLeft)

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(MortLeft = RESPONSE) AND (BorrAmt = RESPONSE)*
AND: *MortType IN [IntLink, IntNoLnk]*
AND: *NOT (RMort = Yes)*
AND: *(MortL1Rs = Suppressed) OR MortL1Ex <> EMPTY*

FRS0605B.QOwner1.QMortgage.M[].MortL1Ex

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
    AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
    AND: *In loop FOR ii := 1 TO 3*
    AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
    AND: *Loan2Y <> Repaid*
    RESERVECHECK

RESERVECHECK

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
    AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
    AND: *In loop FOR ii := 1 TO 3*
    AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
    AND: *Loan2Y <> Repaid*
    AND: *RMort = Yes*

taking_out_the_loan := 'you re-mortgaged'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
    AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
    AND: *In loop FOR ii := 1 TO 3*
    AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
    AND: *Loan2Y <> Repaid*
    AND: *NOT (RMort = Yes)*

taking_out_the_loan := 'taking out the original loan'

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
    AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
    AND: *In loop FOR ii := 1 TO 3*
    AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
    AND: *Loan2Y <> Repaid*
    AND: *NOT ((MorAll = Current) OR (MortType = Repay))*

FRS0605B.QOwner1.QMortgage.M[].MorInPay

^I QOwner1
^I

^N How much was your last payment on this mortgage or loan?^N

0.00..9999.97

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
    AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
    AND: *In loop FOR ii := 1 TO 3*
    AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
    AND: *Loan2Y <> Repaid*
    AND: *NOT ((MorAll = Current) OR (MortType = Repay))*
    AND: *MorInPay = RESPONSE*

LastPay := STR(MorInPay,6,2)

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *NOT ((MorAll = Current) OR (MortType = Repay))*
AND: *MorInPay = DONTKNOW*

LastPay := '??????'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *NOT ((MorAll = Current) OR (MortType = Repay))*
AND: *MorInPay = DONTKNOW*

HMissVar := (HMissVar + 1)

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *NOT ((MorAll = Current) OR (MortType = Repay))*
AND: *MorInPay = REFUSAL*

LastPay := '!!!!!!'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *NOT ((MorAll = Current) OR (MortType = Repay))*
AND: *MorInPay = REFUSAL*

HMissVar := (HMissVar + 1)

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *NOT ((MorAll = Current) OR (MortType = Repay))*

FRS0605B.QOwner1.QMortgage.M[].MorInPx

^I QOwner1
^I

^I^BLU^IC^Pd97Txt^I

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
```

FRS0605B.QOwner1.QMortgage.M[].MorInPd

^I QOwner1
^I

^N How long did this cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPd = Note
```

FRS0605B.QOwner1.QMortgage.M[].MorInPx

^I QOwner1
^I

^I^BLU^IC^Pd97Txt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: Edit = Yes
     MorInPd <> Note
```

^I Editor: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
```

FRS0605B.QOwner1.QMortgage.M[].MorInUs

^I QOwner1
^I

^N Is this the amount you usually pay each time?^N

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInUs = No
```

FRS0605B.QOwner1.QMortgage.M[].MorUs

^I QOwner1
^I

^I If the last interest payment includes arrears accept the actual amount given but if it was a long time ago make a note using <Ctrl+M> to give the date, or if the amount was greater than normal to cover past arrears. Payments by people outside the household should be included.^I

0.00..9999.97

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInUs = No
```

FRS0605B.QOwner1.QMortgage.M[].MorUPx

^I QOwner1
^I

^I^BLU^IC^Pd97Txt^I

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInUs = No
```

FRS0605B.QOwner1.QMortgage.M[].MorUPd

^I QOwner1
^I

^N How long does this cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInUs = No
     AND: MorUPd = Note
```

FRS0605B.QOwner1.QMortgage.M[].MorUPx

^I QOwner1
^I

^I^BLU^IC^Pd97Txt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInUs = No
     AND: Edit = Yes
     MorUPd <> Note
```

^I Editor: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^I

FRS0605B.QOwner1.QMortgage.M[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorUs > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0605B.QOwner1.QMortgage.M[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: NOT (MorUs > 0)
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0605B.QOwner1.QMortgage.M[] (continued)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

MorIWkly := LWeekly

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: MortLeft = RESPONSE
```

EPIntC := (((MorIWkly * 52) / MortLeft) * 100)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: MortLeft = RESPONSE
```

IntFill := ROUND(EPIntC)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: MortLeft = RESPONSE
     AND: EPIntC <= 2
```

higher := 'lower'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: MortLeft = RESPONSE
     AND: EPIntC >= 11
```

higher := 'higher'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: NOT ((MorAll = Current) OR (MortType = Repay))
     AND: MorInPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: MortLeft = RESPONSE
     ((EPIntC > 2) AND (EPIntC < 11)) AND INVOLVING(MorInPd,MorInPay)
```

^I The interest payments work out roughly at ^IntFill per cent which is ^higher than most current interest rates available for a mortgage of this size.
If no particular reason for this, please check your answers.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
```

MenPolAm0 := Yes

FRS0605B.QOwner1.QMortgage.M[].QEndow[]

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
```

MortSeq := PPSeq

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
```

EndowSeq := PCount

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: PMenpol = Yes
     AND: PCount > 1
```

next := 'next'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: PMenpol = Yes
     AND: NOT (PCount > 1)
```

next := 'first'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: PMenpol = Yes
```

premium_payment := 'premium'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: PMenpol = Yes
```

policies_plans := 'endowment policies'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: NOT (PMenpol = Yes)
```

premium_payment := 'payment'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: NOT (PMenpol = Yes)
```

policies_plans := 'savings or investment plans'

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
```

FRS0605B.QOwner1.QMortgage.M[].QEndow[].MenPolAm

^I QOwner1
^I

^N How much was your last ^payment1 ^next ^payment2?^N
^I^BLU^IC: For interest only mortgages include combined interest and endowment payment.^I

0.00..9999.97

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm = NONRESPONSE
```

HMissVar := (HMissVar + 1)

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*
AND: *MorAll <> Current*
AND: *In loop FOR Count := 1 TO 4*
AND: *(Count = 1) OR (QEndow[Count - 1].MpMore = Yes)*
AND: *MenPolAm > 0*

FRS0605B.QOwner1.QMortgage.M[].QEndow[].MenPolPx

^I QOwner1
^I

^I^BLU^IC^Pd97Txt^I

OPEN

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*
AND: *MorAll <> Current*
AND: *In loop FOR Count := 1 TO 4*
AND: *(Count = 1) OR (QEndow[Count - 1].MpMore = Yes)*
AND: *MenPolAm > 0*

FRS0605B.QOwner1.QMortgage.M[].QEndow[].MenPolPd

^I QOwner1
^I

^N How long did this cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*
AND: *MorAll <> Current*
AND: *In loop FOR Count := 1 TO 4*
AND: *(Count = 1) OR (QEndow[Count - 1].MpMore = Yes)*
AND: *MenPolAm > 0*
AND: *MenPolPd = Note*

FRS0605B.QOwner1.QMortgage.M[].QEndow[].MenPolPx

^I QOwner1
^I

^I^BLU^IC^Pd97Txt^I

OPEN

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin)) OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)*
AND: *MorAll <> Current*
AND: *In loop FOR Count := 1 TO 4*
AND: *(Count = 1) OR (QEndow[Count - 1].MpMore = Yes)*
AND: *MenPolAm > 0*
AND: *Edit = Yes*
MenPolPd <> Note

^I Editor: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^I

FRS0605B.QOwner1.QMortgage.M[].QEndow[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year]))
```

PWeekly := 0

FRS0605B.QOwner1.QMortgage.M[].QEndow[] (continued)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
     AND: MenPolPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

MenPWkly := LWeekly

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: MenPolAm > 0
     AND: MenPolPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: Edit = No
     (MenPWkly < 100) AND INVOLVING(MenPolPd,MenPolAm)
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: (SUBSTRING (PLastPay, 1, 1) <> 0) AND (MenPolAm > 0)
```

FRS0605B.QOwner1.QMortgage.M[].QEndow[].IncInInt

^I QOwner1
^I

^N Is this ^premium_payment included in the amount you mentioned earlier (£^PLastPay)?^N

(1)	Yes	Yes
(2)	No	No

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: (SUBSTRING (PLastPay, 1, 1) <> 0) AND (MenPolAm > 0)
     AND: PMenpol = Yes
```

FRS0605B.QOwner1.QMortgage.M[].QEndow[].MenstRs

^I QOwner1
^I

^N Are you sure? That means the endowment was purchased before HHldr was 18.^N
^I^BLU^IC Check the circumstances and explain in a note.^I

(1)	Passed	Passed
(2)	Hard	Hard
(3)	Soft	Soft
(4)	Suppress	Suppressed

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: (SUBSTRING (PLastPay, 1, 1) <> 0) AND (MenPolAm > 0)
     AND: PMenpol = Yes
```

FRS0605B.QOwner1.QMortgage.M[].QEndow[].MenstEx

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: (SUBSTRING (PLastPay, 1, 1) <> 0) AND (MenPolAm > 0)
     AND: PMenpol = Yes
```

FRS0605B.QOwner1.QMortgage.M[].QEndow[].MenstYr

^I QOwner1
^I

^N In what year was this endowment ^policy taken out?^N

1901..2007

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: (SUBSTRING (PLastPay, 1, 1) <> 0) AND (MenPolAm > 0)
     AND: PMenpol = Yes
     AND: QDataBag.SampMnth IN [4 .. 12]
     MenstYr <> 2007
```

^I Wrong Year!^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: (SUBSTRING (PLastPay, 1, 1) <> 0) AND (MenPolAm > 0)
     AND: PMenpol = Yes
     AND: (((Edit <> Yes) AND (QSignIn.StartDat = RESPONSE)) AND
     (HHG.P[QHholder.DVHRPNum].AgeOf > 0)) AND (MenstYr > 0)
     (MenstYr >= ((QSignIn.StartDat.YEAR -
     HHG.P[].AgeOf[QHholder.DVHRPNum]) + 17)) AND INVOLVING(MenstYr)
```

(MenstYr >= ((QSignIn.StartDat.YEAR - HHG.P[QHholder.DVHRPNum].AgeOf) + 17)) AND INVOLVING (MenstYr)

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: (SUBSTRING (PLastPay, 1, 1) <> 0) AND (MenPolAm > 0)
     AND: PMenpol = Yes
     AND: (((Edit <> Yes) AND (QSignIn.StartDat = RESPONSE)) AND
     (HHG.P[QHholder.DVHRPNum].AgeOf > 0)) AND (MenstYr > 0)
     AND: (MenstRs = Suppressed) OR MenstEx <> EMPTY
```

FRS0605B.QOwner1.QMortgage.M[].QEndow[].MenstEx

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: PCount < 4
```

FRS0605B.QOwner1.QMortgage.M[].QEndow[].MpMore

^I QOwner1
^I

^N Can I just check, are there any more savings or investment plans covering the repayment of the mortgage or loan?^N

(1)	Yes	Yes
(2)	No	No

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     RESERVECHECK
```

RESERVECHECK

FRS0605B.QOwner1.QMortgage.M[] (continued)

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     (QEndow[Count].MenPWkly <= MorIWkly) AND
     INVOLVING(QEndow[Count].MenPolAm)
```

^I The payment/endowment premium is more than the last mortgage payment at MorInPay. This is very unusual - please check your figures.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: (MorIWkly > 0) AND (QEndow[Count].IncInInt = Yes)
     (QEndow[Count].MenPWkly < MorIWkly) AND
     INVOLVING(QEndow[Count].MenPolAm,QEndow[Count].MenPolPd)
```

^I The payment/endowment premium is included in the interest payment of £^LastPay, so it can't exceed this amount. Please check your figures.^I

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     AND: (Count = 1) OR (QEndow[Count - 1].MpMore = Yes)
     AND: QEndow[Count].MenPolAm <> 0

MenPolAm0 := No
```

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: MorAll <> Current
     AND: In loop FOR Count := 1 TO 4
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: ((((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: (MortSeq = 1) AND (QEndow[1].MpMore = No)
     NOT(QMortgage.M[].MortType[MortSeq] = IntLink) AND
     INVOLVING(QEndow[1].MpMore)
```

^I Are you sure? Earlier the respondent said they had more than 1 savings / investment policy for this mortgage. Check if the respondent has another policy to tell you about. If not you MUST make a note of the circumstances.^I

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
```

FRS0605B.QOwner1.QMortgage.M[].MpMore

^I QOwner1
^I

^N Are there any more policies/plans covering the repayment of the mortgage or loan?^N

(1)	Yes	Yes
(2)	No	No

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (((MenPol = Yes) OR (Pension IN EndwPrin)) OR (PEP IN EndwPrin))
     OR (UnitT IN EndwPrin)) OR (OthSch IN EndwPrin)
     AND: QEndow[1].MpMore = Yes
```

MpMore := Yes

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
```

FRS0605B.QOwner1.QMortgage.M[].IntPrPay

^I QOwner1
^I

^N How much was your last payment on this mortgage or loan?^N

0.00..9999.97

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: (Edit = No) AND (IntPrPay = RESPONSE)
     NOT(IntPrPay = 0)
```

^I You have entered that the respondent's last instalment on the mortgage/loan was £0. Please do not enter zero even if they paid nothing last time. Please collect the amount they usually pay or if there is no usual, the contractual or notional amount they would need to pay in order for the mortgage/loan to be paid off in the agreed period.^I

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntPrPay = RESPONSE
```

LastPay := STR(IntPrPay,6,2)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntPrPay = DONTKNOW
```

LastPay := '??????'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntPrPay = DONTKNOW
```

HMissVar := (HMissVar + 1)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntPrPay = REFUSAL
```

LastPay := '!!!!!!'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(MorAll <> Current) AND (MortType = Repay)*
AND: *IntPrPay = REFUSAL*

HMissVar := (HMissVar + 1)

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(MorAll <> Current) AND (MortType = Repay)*

FRS0605B.QOwner1.QMortgage.M[].IntPrPx

^I QOwner1
^I

^I^BLU^IC^Pd97Txt^I

OPEN

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(MorAll <> Current) AND (MortType = Repay)*

FRS0605B.QOwner1.QMortgage.M[].IntPrPd

^I QOwner1
^I

^N How long did this cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntPrPd = Note
```

FRS0605B.QOwner1.QMortgage.M[].IntPrPx

^I QOwner1
^I

^I^BLU^IC^Pd97Txt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: Edit = Yes
     IntPrPd <> Note
```

^I Editor: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^I

FRS0605B.QOwner1.QMortgage.M[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0605B.QOwner1.QMortgage.M[] (continued)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

IntPWkly := LWeekly

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     (IntPWkly < 650) AND INVOLVING(IntPrPd,IntPrPay)
```

^I Are you sure? That is higher than the amount usually entered here. Confirm that the last payment was this amount and if Yes suppress check.^I

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: RMAmt > 0
```

PrIntC := ((((IntPWkly - (RMAmt / (25 * 52))) * 52) / (RMAmt * 0.62)) * 100)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: BorrAmt > 0
```

PrIntC := ((((IntPWkly - (BorrAmt / (25 * 52))) * 52) / (BorrAmt * 0.62)) * 100)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: PrIntC < 3
```

higher := 'lower'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: PrIntC > 10
```

higher := 'higher'

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: NOT (Edit = Yes)
     AND: IntPrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     ((PrIntC >= 3) AND (PrIntC <= 10)) AND INVOLVING(IntPrPay)
```

^I You have entered an amount that is ^higher than that usually paid for a mortgage of this size. Please check that you have entered the correct payment.^I

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(MorAll <> Current) AND (MortType = Repay)*

FRS0605B.QOwner1.QMortgage.M[].IntrUs

^I QOwner1
^I

^N Is this the amount you usually pay each time?^N

(1)	Yes	Yes
(2)	No	No

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(MorAll <> Current) AND (MortType = Repay)*
AND: *IntrUs = No*

FRS0605B.QOwner1.QMortgage.M[].IntrU

^I QOwner1
^I

^N How much are your usual payments on this mortgage or loan?^N

0.00..9999.97

RECORD IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(MorAll <> Current) AND (MortType = Repay)*
AND: *IntrUs = No*

FRS0605B.QOwner1.QMortgage.M[].IntrPx

^I QOwner1
^I

^I^BLU^IC^Pd97Txt^I

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
```

FRS0605B.QOwner1.QMortgage.M[].IntrPd

^I QOwner1
^I

^N How long did this cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: IntrPd = Note
```

FRS0605B.QOwner1.QMortgage.M[].IntrPx

^I QOwner1
^I

^I^BLU^IC^Pd97Txt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: Edit = Yes
     IntrPd <> Note
```

^I Editor: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^I

FRS0605B.QOwner1.QMortgage.M[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]

PdConW[1] := 1
```

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]

PdConW[2] := 2
```

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]

PdConW[3] := 3
```

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]

PdConW[4] := 4
```

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
```

PdConW[9] := 5.78

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(MorAll <> Current) AND (MortType = Repay)*
AND: *IntrUs = No*
AND: *NOT (Edit = Yes)*
AND: *IntrPd IN [OneWeek .. Year]*

PdConW[10] := 5.2

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(MorAll <> Current) AND (MortType = Repay)*
AND: *IntrUs = No*
AND: *NOT (Edit = Yes)*
AND: *IntrPd IN [OneWeek .. Year]*

PdConW[13] := 13

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(MorAll <> Current) AND (MortType = Repay)*
AND: *IntrUs = No*
AND: *NOT (Edit = Yes)*
AND: *IntrPd IN [OneWeek .. Year]*

PdConW[26] := 26

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(MorAll <> Current) AND (MortType = Repay)*
AND: *IntrUs = No*
AND: *NOT (Edit = Yes)*
AND: *IntrPd IN [OneWeek .. Year]*

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0605B.QOwner1.QMortgage.M[] (continued)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

IntPWkly := LWeekly

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     (IntPWkly < 650) AND INVOLVING(IntrPd,IntrU)
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: RMAmt > 0
```

PrIntC := ((((IntPWkly - (RMAmt / (25 * 52))) * 52) / (RMAmt * 0.62)) * 100)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: BorrAmt > 0
```

PrIntC := ((((IntPWkly - (BorrAmt / (25 * 52))) * 52) / (BorrAmt * 0.62)) * 100)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: PrIntC < 3
```

higher := 'lower'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: (MorAll <> Current) AND (MortType = Repay)
     AND: IntrUs = No
     AND: NOT (Edit = Yes)
     AND: IntrPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)
     AND: PrIntC > 10
```

higher := 'higher'

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(MorAll <> Current) AND (MortType = Repay)*
AND: *IntrUs = No*
AND: *NOT (Edit = Yes)*
AND: *IntrPd IN [OneWeek .. Year]*
AND: *LWeekly > 0*
AND: *(RMAmt = RESPONSE) OR (BorrAmt = RESPONSE)*
((PrIntC >= 3) AND (PrIntC <= 10)) AND INVOLVING(IntrU)

^I You have entered an amount that is ^higher than that usually paid for a mortgage of this size. Please check that you have entered the correct payment.^I

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *PPTenure IN [Mortgage, Part]*
AND: *MenPol = Yes*

Apart_do := ('Apart from any endowment policies already ' + 'mentioned, do')

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *PPTenure IN [Mortgage, Part]*
AND: *MenPol = Yes*

redundancy := ' or redundancy'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *PPTenure IN [Mortgage, Part]*
AND: *MenPol = Yes*

death := '(NOT USED)'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *PPTenure IN [Mortgage, Part]*
AND: *NOT (MenPol = Yes)*

Apart_do := 'Do'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: NOT (MenPol = Yes)
```

redundancy := ', redundancy or death'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: NOT (MenPol = Yes)
```

death := 'Death'

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
```

FRS0605B.QOwner1.QMortgage.M[].MortProt

^I QOwner1
^I

^I Only include policies which specifically pay the^B mortgage^B. Do not confuse these with polices that simply pay out money in the event of redundancy or sickness (and could be used to pay for anything).^I

(1)	Yes	Yes
(2)	No	No

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: (MortProt = RESPONSE) AND (MortType <> Endow)
     MortProt = Yes
```

^I Interviewer: for this type of mortgage there is normally a protection policy. Please check - is it included in the last mortgage payment? (If no policy, suppress warning and continue.)^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
```

FRS0605B.QOwner1.QMortgage.M[].MPCover

^I QOwner1
^I

^N What is covered by the mortgage protection policy?^N

^I^BLU^IC Probe to classify.
Code all that apply.^I

SET [3] OF

(1)	Sick	Sickness/accident
(2)	Redund	Redundancy/loss of employment
(3)	Dead	^death

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: MenPol = Yes
     NOT(IN(Dead,MPCover))
```

^I This code is not valid for this question.^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: MPCover.CARDINAL > 1
```

FRS0605B.QOwner1.QMortgage.M[].MPolNo

^I QOwner1
^I

^N Can I check, is there^B one^B mortgage protection policy, or^B more than^B one?^N

^I^BLU^IC Count as separate policy if separate^B payments (premiums)^B are made.

Enter number of policies.^I

1..3

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: ((IntPrPay > 0) OR (MorInPay > 0)) OR (MenPolAm0 = No)
     AND: IntPrPay <> EMPTY OR (MorInPay > 0)
```

PCP := ('your last payment on the mortgage/loan (£' + LastPay + ')')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: ((IntPrPay > 0) OR (MorInPay > 0)) OR (MenPolAm0 = No)
     AND: IntPrPay <> EMPTY OR (MorInPay > 0)
     AND: MorInPay > 0
     AND: MenPolAm0 = No
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

PCP := (PCP + ' or in the (pension/PEP/ISA/Unit' + ' Trust) contribution')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: ((IntPrPay > 0) OR (MorInPay > 0)) OR (MenPolAm0 = No)
     AND: IntPrPay <> EMPTY OR (MorInPay > 0)
     AND: MorInPay > 0
     AND: MenPolAm0 = No
     AND: NOT ((((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

PCP := (PCP + ' or in the endowment premium')

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: ((IntPrPay > 0) OR (MorInPay > 0)) OR (MenPolAm0 = No)
     AND: MorInPay <> EMPTY AND (MenPolAm0 = No)
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

PCP := 'the (pension/PEP/ISA/Unit Trust) contribution'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: ((IntPrPay > 0) OR (MorInPay > 0)) OR (MenPolAm0 = No)
     AND: MorInPay <> EMPTY AND (MenPolAm0 = No)
     AND: NOT (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

PCP := 'the endowment premium'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

PC := 'pension/PEP/ISA/Unit Trust contribution'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: NOT (((Pension IN EndwPrin) OR (PEP IN EndwPrin)) OR (UnitT IN
     EndwPrin)) OR (OthSch IN EndwPrin)
```

PC := 'endowment premium'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
```

Order[1] := 'FIRST'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
```

Order[2] := 'SECOND'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
```

Order[3] := 'THIRD'

FRS0605B.QOwner1.QMortgage.M[].QMortProt[]

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
```

LPayment_etc := ppayment

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: LPayment_etc =
```

LPayment_etc := 'the mortgage payment you mentioned earlier'

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
```

FRS0605B.QOwner1.QMortgage.M[].QMortProt[].IncMPAmt

^I QOwner1
^I

^B *** ^Order[Count] MORTGAGE PROTECTION POLICY ***^B

@?^I If the precise amount for the mortgage protection policy cannot be given, please ask the respondent to given an estimate rather than accept DK.^I

0.00..9997.99

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

FRS0605B.QOwner1.QMortgage.M[].QMortProt[].IncMPPx

^I QOwner1
^I

^B *** ^Order[Count] MORTGAGE PROTECTION POLICY ***^B

^I^BLU^IC^Pd97Txt^I

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

FRS0605B.QOwner1.QMortgage.M[].QMortProt[].IncMPPd

^I QOwner1
^I

^B *** ^Order[Count] MORTGAGE PROTECTION POLICY ***^B

^N How long did this cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
     AND: IncMPPd = Note
```

FRS0605B.QOwner1.QMortgage.M[].QMortProt[].IncMPPx

^I QOwner1
^I

^B *** ^Order[Count] MORTGAGE PROTECTION POLICY ***^B

^I^BLU^IC^Pd97Txt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
     AND: Edit = Yes
     IncMPPd <> Note
```

^I Editor: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^I

FRS0605B.QOwner1.QMortgage.M[].QMortProt[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0605B.QOwner1.QMortgage.M[].QMortProt[] (continued)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
     AND: IncMPPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

IncMWkly := LWeekly

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IncMPAmt > 0
     AND: IncMPPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     (IncMWkly < 30) AND INVOLVING(IncMPPd,IncMPAmt)
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
```

FRS0605B.QOwner1.QMortgage.M[].QMortProt[].IncMStYr

^I QOwner1
^I

^B *** ^Order[Count] MORTGAGE PROTECTION POLICY ***^B

^N In what year was the mortgage protection policy taken out?^N

1901..2007

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: QDataBag.SampMnth IN [4 .. 12]
     IncMStYr <> 2007
```

^I Wrong Year!^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: (IncMPAmt > 0) OR IncMPAmt = NONRESPONSE
```

FRS0605B.QOwner1.QMortgage.M[].QMortProt[].IncMP

^I QOwner1
^I

^B *** ^Order[Count] MORTGAGE PROTECTION POLICY ***^B

^N Was this mortgage protection payment included in ^LPayment_etc?^N

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: (IncMPAmt > 0) OR IncMPAmt = NONRESPONSE
     AND: ((IncMP = Yes) AND (SUBSTRING (PLastPay, 1, 1) <> 0)) AND
     (PMenPolAm0 = No)
```

FRS0605B.QOwner1.QMortgage.M[].QMortProt[].IncMIncl

^I QOwner1
^I

^B *** ^Order[Count] MORTGAGE PROTECTION POLICY ***^B

^I Ask or record.^I
Was it included in the mortgage payment or the ^ppremium?

(1)	Mort	mortgage payment
(2)	Endprm	^ppremium

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *PPTenure IN [Mortgage, Part]*
AND: *MortProt = Yes*
AND: *In loop FOR Count := 1 TO 3*
AND: *(Count = 1) OR (Count <= MPolNo)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *PPTenure IN [Mortgage, Part]*
AND: *MortProt = Yes*
AND: *In loop FOR Count := 1 TO 3*
AND: *(Count = 1) OR (Count <= MPolNo)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *PPTenure IN [Mortgage, Part]*
AND: *MortProt = Yes*
AND: *In loop FOR Count := 1 TO 3*
AND: *(Count = 1) OR (Count <= MPolNo)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *PPTenure IN [Mortgage, Part]*
AND: *MortProt = Yes*
AND: *In loop FOR Count := 1 TO 3*
AND: *(Count = 1) OR (Count <= MPolNo)*
RESERVECHECK

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     RESERVECHECK
```

RESERVECHECK

FRS0605B.QOwner1.QMortgage.M[] (continued)

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: (PSeq IN [1 .. 2]) AND (QMortProt[Count].IncMStYr = RESPONSE)
     PBuyYear <= QMortProt[Count].IncMStYr
```

^I The mortgage protection policy was taken out BEFORE the mortgage started ('BuyYear'). This seems very unusual - please check your dates.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: IntPWkly = RESPONSE
     (QMortProt[Count].IncMWkly <= IntPWkly) AND
     INVOLVING(QMortProt[Count].IncMPAmt,QMortProt[Count].IncMPPd,IntPrPay)
```

^I The mortgage protection premium is more than the last mortgage payment at IntPrPay (£^LastPay). This is very unusual - please check your figures.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: MorAll = Current
     QMortProt[Count].IncMPAmt <> DONTKNOW
```

^I Interviewer: Please try not to collect Don't Know at this question. Ask your respondent to be more specific and provide you with their best estimate as to their monthly payment.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     AND: (Count = 1) OR (Count <= MPolNo)
     AND: Edit = Yes
     QMortProt[Count].IncMPAmt<>NONRESPONSE
```

^I Missing amount for Mortgage Protection Policy. Note the size of last mortgage payment (£^LastPay), then follow Edit Instructions to fill in IncMPAmt.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: PPTenure IN [Mortgage, Part]
     AND: MortProt = Yes
     AND: In loop FOR Count := 1 TO 3
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     RESERVECHECK
```

RESERVECHECK

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*

FRS0605B.QOwner1.QMortgage.M[].OutsMort

^I QOwner1
^I

^N Does anyone from outside the household pay anything towards THIS mortgage/loan on your behalf, on a regular basis?^N

(1)	Yes	Yes
(2)	No	No

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *OutsMort = Yes*

FRS0605B.QOwner1.QMortgage.M[].QOutsPay

^I QOwner1
^I

^N Who is that?^N

SET [6] OF

(1)	GOV	^GOV2
(2)	Emp	Employer
(3)	Org	Other organisation
(4)	Rel	Friend or relative
(5)	Pol	Mortgage protection/insurance policy
(6)	Other	Other

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *OutsMort = Yes*

Payer[1] := GOV1

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *OutsMort = Yes*

Payer[2] := 'employer'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *OutsMort = Yes*

Payer[3] := 'other organisation'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *OutsMort = Yes*

Payer[4] := 'relative or friend'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *OutsMort = Yes*

Payer[5] := 'policy'

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *OutsMort = Yes*

Payer[6] := '

FRS0605B.QOwner1.QMortgage.M[].QOutside[]

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
```

MortSeq := PPSeq

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
```

ContSeq := POutsPay

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
```

OutsPay := POutsPay

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
```

FRS0605B.QOwner1.QMortgage.M[].QOutside[].OutsAmt

^I QOwner1
^I

^N How much did the ^PPayer pay last time?^N

0.01..999997.00

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

FRS0605B.QOwner1.QMortgage.M[].QOutside[].OutsPx

^I QOwner1
^I

^I^BLU^IC^Pd97Txt^I

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

FRS0605B.QOwner1.QMortgage.M[].QOutside[].OutsPd

^I QOwner1
^I

^N How long did that cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
     AND: OutsPd = Note
```

FRS0605B.QOwner1.QMortgage.M[].QOutside[].OutsPx

^I QOwner1
^I

^I^BLU^IC^Pd97Txt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
     AND: Edit = Yes
     OutsPd <> Note
```

^I Editor: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^I

FRS0605B.QOwner1.QMortgage.M[].QOutside[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
```

PdConW[52] := 52

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *OutsMort = Yes*
AND: *In loop FOR Count := 1 TO 6*
AND: *Count IN QOutsPay*
AND: *OutsAmt > 0*
AND: *(PAmount > 0) AND (PPeriod IN [OneWeek .. Year])*

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

COMPUTE IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *OutsMort = Yes*
AND: *In loop FOR Count := 1 TO 6*
AND: *Count IN QOutsPay*
AND: *OutsAmt > 0*
AND: *NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])*

PWeekly := 0

FRS0605B.QOwner1.QMortgage.M[].QOutside[] (continued)

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
     AND: OutsPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
```

OutWkly := LWeekly

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: OutsAmt > 0
     AND: OutsPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
     AND: Edit = No
     (OutWkly < 159) AND INVOLVING(OutsPd,OutsAmt)
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Loan2Y <> Repaid
     AND: OutsMort = Yes
     AND: In loop FOR Count := 1 TO 6
     AND: Count IN QOutsPay
     AND: (OutsAmt <> 0) AND (SUBSTRING (PLastPay, 1, 1) <> 0)
```

FRS0605B.QOwner1.QMortgage.M[].QOutside[].OutsIncl

^I QOwner1
^I

^N Was this included in the mortgage payment that you mentioned earlier?^N

(1)	Yes	Yes
(2)	No	No

FRS0605B.QOwner1.QMortgage.M[] (continued)

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *OutsMort = Yes*
AND: *In loop FOR Count := 1 TO 6*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *OutsMort = Yes*
AND: *In loop FOR Count := 1 TO 6*
RESERVECHECK

RESERVECHECK

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *Loan2Y <> Repaid*
AND: *(PSeq = 1) AND (PBuyYear > 1980)*

FRS0605B.QOwner1.QMortgage.M[].ExRent

^I QOwner1
^I

^N Had you been renting this house/flat before deciding to buy it?^N

^I^BLU 'You' = HRP/Householder, or spouse/partner^I

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: Loan2Y <> Repaid
     AND: (PSeq = 1) AND (PBuyYear > 1980)
     AND: ExRent = Yes
```

FRS0605B.QOwner1.QMortgage.M[].RentFrom

^I QOwner1
^I

^N Who was it rented from?^N

^I^BLU Prompt as necessary.^I

(1)	LA	^Council2
(2)	HA	Housing Association, co-operative, charitable trust
(3)	Emp	Employer
(4)	OthOrg	Other organisation
(5)	OthInd	Other individual

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[]))))
     AND: (PSeq = 1) AND (PPurcLoan = One)
```

FRS0605B.QOwner1.QMortgage.M[].OthMort1

^I QOwner1
^I

^N 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 secured on^B this^B property?^N

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: (PSeq = 2) AND (PPurcLoan = Two)
```

FRS0605B.QOwner1.QMortgage.M[].OthMort2

^I QOwner1
^I

^N May I just check, are you currently using^B this house/flat^B as security for a mortgage or loan of any other kind?^N

(1)	Yes	Yes
(2)	No	No

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: (OthMort1 = Yes) OR (OthMort2 = Yes)
```

FRS0605B.QOwner1.QMortgage.M[].OthPurRs

^I QOwner1
^I

^I This should only apply to loans for purchase. Please resolve, or make a Note.^I

(1)	Passed	Passed
(2)	Hard	Hard
(3)	Soft	Soft
(4)	Suppress	Suppressed

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: (OthMort1 = Yes) OR (OthMort2 = Yes)
```

FRS0605B.QOwner1.QMortgage.M[].OthPurEx

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *(OthMort1 = Yes) OR (OthMort2 = Yes)*

FRS0605B.QOwner1.QMortgage.M[].OthPur

^I QOwner1
^I

^IS^I N^I

^N Which of these items best describe the reasons why you took out the other loan or loans?
^Any_others?^N
^I^BLU^IC Code all that apply.^I

SET [7] OF

(1)	Improve	To make improvements or extensions to this property
(2)	Purcase	To help purchase a major item like a car, boat, caravan or second home
(3)	IntrRate	To get a better, or fixed, interest rate
(4)	Business	In connection with a business
(5)	BuyOut	To buy out another person's share in the property
(6)	Repairs	For essential repairs to make the property fit for occupation
(7)	Other	Some other purpose (INTERVIEWER: SPECIFY IN A NOTE.)

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *(OthMort1 = Yes) OR (OthMort2 = Yes)*
NOT(IN(IntrRate,OthPur))

NOT (IntrRate IN OthPur)

ASK IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
AND: *(OthMort1 = Yes) OR (OthMort2 = Yes)*
AND: *(OthPurRs = Suppressed) OR OthPurEx <> EMPTY*

FRS0605B.QOwner1.QMortgage.M[].OthPurEx

^I QOwner1
^I

^I^BLU^IC^SuppTxt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Edit = Yes
     AND: NOT (MortType IN [Endow, EndRep])
     NOT(IN(None,EndwPrin))
```

^I Editor: Mortgage Capital repaid by 'unknown' method: There should be a note attached. Please re-code into 1-4, IF possible.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Edit = Yes
     IntPrPay<>NONRESPONSE AND IntPrPd<>NONRESPONSE
```

^I
Missing amount and/or period for Mortgage Instalment.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Edit = Yes
     MorInPay<>NONRESPONSE AND MorInPd<>NONRESPONSE
```

^I Missing amount and/or period for Mortgage Instalment.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: Edit = Yes
     MenPol <> No
```

^I

There are no endowment policies covering the repayment of this mortgage OR loan.^I^B

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     AND: (RMAmt = RESPONSE) AND (BorrAmt = RESPONSE)
     BorrAmt <= RMAmt
```

^I The re-mortgage amount would normally be at least as large as the original mortgage.
Please check your figures.^I

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     AND: (((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan =
     Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR
     (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))
     RESERVECHECK
```

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
AND: *PurcAmt <> EMPTY OR (Repairs IN OthPur3)*
AND: *In loop FOR ii := 1 TO 3*
AND: *(((PPPurcLoan IN [One .. Two]) AND (ii = 1)) OR ((PPPurcLoan = Two) AND (ii = 2))) OR ((ii = 3) AND (((Repairs IN M[1].OthPur) OR (Repairs IN M[2].OthPur)) OR (Repairs IN OthPur3[])))*
RESERVECHECK

RESERVECHECK

FRS0605B.QOwner1.QMortgage

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     AND: In loop FOR ii := 1 TO 3
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     AND: PurcAmt <> EMPTY OR (Repairs IN OthPur3)
     RESERVECHECK
```

RESERVECHECK

FRS0605B.QOwner1 (continued)

Questions about mortgages

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part]
     RESERVECHECK
```

RESERVECHECK

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.Tenure IN [Outright .. Part]*
RESERVECHECK

RESERVECHECK

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *(((QAccomdat.Tenure IN [Mortgage .. Part]) OR (QOwner1.OthMort3 = Yes)) AND (QOwner1.QMortgage.M[1].MorInPay = EMPTY OR (QOwner1.QMortgage.M[1].MorInPay > 0))) AND (QOwner1.QMortgage.M[2].MorInPay = EMPTY OR (QOwner1.QMortgage.M[2].MorInPay > 0))*

AskStruc := 1

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *(((QAccomdat.Tenure IN [Outright .. Part]) OR QAccomdat.Tenure = NONRESPONSE) OR (QRenting.Landlord IN [Assocn .. OthIndiv])) OR QRenting.Landlord = NONRESPONSE*
AND: *AskStruc = 1*

AskStruc := 3

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *(((QAccomdat.Tenure IN [Outright .. Part]) OR QAccomdat.Tenure = NONRESPONSE) OR (QRenting.Landlord IN [Assocn .. OthIndiv])) OR QRenting.Landlord = NONRESPONSE*
AND: *NOT (AskStruc = 1)*

AskStruc := 2

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *(QAccomdat.Tenure = Part) AND (QAccomdat.SOBuy = Paid)*

AskStruc := 2

FRS0605B.QInsur

Questions about structure insurance.

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
```

FRS0605B.QInsur.StrMort

^I QInsur
^I
^N Did your last payment on the mortgage/ loan include an amount for any insurance on the structure of this accommodation, its furniture or contents, or any personal possessions?^N

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
```

FRS0605B.QInsur.StrCov

^I QInsur
^I
^N Was that for...^N^I^BLU Running prompt...^I

(1)	Struct	^N...structure^B only^B^N
(2)	Furn	^N...furniture and contents or personal possessions, only^N
(3)	Combine	^N...structure^B and^B furniture and contents, or personal possessions?^N

FRS0605B.QInsur.QStructure[]

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: (PStrCov = Struct) OR (PCovOths = Struct)
```

combined := ' structure'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: PStrCov = Furn
```

combined := ' contents'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: (PStrCov = Combine) OR (PCovOths = Combine)
```

combined := ' combined'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: PSeq = 1
```

included := ' included in your last mortgage payment'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: NOT (PSeq = 1)
```

last := ' last'

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
```

FRS0605B.QInsur.QStructure[].StrAmt

^I QInsur
^I

^N How much was the^last premium^included for this^combined policy?^N

0.01..99997.00

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

RECORD IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

FRS0605B.QInsur.QStructure[].StrPx

^I QInsur
^I

^I^BLU^IC ^Pd97Txt^I

OPEN

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*

FRS0605B.QInsur.QStructure[].StrPd

^I QInsur
^I

^N How long did this cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *PAskStruc IN [1, 3]*
AND: *StrMort = Yes*
AND: *StrAmt > 0*
AND: *StrPd = Note*

FRS0605B.QInsur.QStructure[].StrPx

^I QInsur
^I

^I^BLU^IC ^Pd97Txt^I

OPEN

FRS0605B.QInsur.QStructure[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0605B.QInsur.QStructure[] (continued)

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
     AND: StrPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
```

StrWkly := LWeekly

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     AND: StrAmt > 0
     AND: StrPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
     AND: Edit = No
     (StrWkly < 50) AND INVOLVING(StrPd,StrAmt)
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     StrPd <> Note
```

^I Editor: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^N

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [1, 3]
     AND: StrMort = Yes
     RESERVECHECK
```

RESERVECHECK

FRS0605B.QInsur (continued)

Questions about structure insurance.

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *PAskStruc IN [2 .. 3]*
AND: *(StrMort = EMPTY OR (StrMort = No)) OR (StrCov = Furn)*

FRS0605B.QInsur.StrOths

^I QInsur
^I
^N Do you pay an insurance premium on the^B structure^B of this accommodation?^N

(1)	Yes	Yes
(2)	No	No

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *PAskStruc IN [2 .. 3]*
AND: *StrOths = Yes*

FRS0605B.QInsur.CovOths

^I QInsur
^I
^N Does the premium cover...^N^I^BLU Running prompt...^I

(1)	Struct	^N...structure^B only^B,^N
(2)	Combine	^N...or structure combined with furniture, contents or personal possessions?^N

FRS0605B.QInsur.QStructure[]

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: (PStrCov = Struct) OR (PCovOths = Struct)
```

combined := ' structure'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: PStrCov = Furn
```

combined := ' contents'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: (PStrCov = Combine) OR (PCovOths = Combine)
```

combined := ' combined'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: PSeq = 1
```

included := ' included in your last mortgage payment'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: NOT (PSeq = 1)
```

last := ' last'

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
```

FRS0605B.QInsur.QStructure[].StrAmt

^I QInsur
^I

^N How much was the^last premium^included for this^combined policy?^N

0.01..99997.00

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

RECORD IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *PAskStruc IN [2 .. 3]*
AND: *StrOths = Yes*
AND: *StrAmt > 0*

FRS0605B.QInsur.QStructure[].StrPx

^I QInsur
^I

^I^BLU^IC ^Pd97Txt^I

OPEN

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *PAskStruc IN [2 .. 3]*
AND: *StrOths = Yes*
AND: *StrAmt > 0*

FRS0605B.QInsur.QStructure[].StrPd

^I QInsur
^I

^N How long did this cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *PAskStruc IN [2 .. 3]*
AND: *StrOths = Yes*
AND: *StrAmt > 0*
AND: *StrPd = Note*

FRS0605B.QInsur.QStructure[].StrPx

^I QInsur
^I

^I^BLU^IC ^Pd97Txt^I

OPEN

FRS0605B.QInsur.QStructure[].Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0605B.QInsur.QStructure[] (continued)

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
     AND: StrPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
```

StrWkly := LWeekly

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     AND: StrAmt > 0
     AND: StrPd IN [OneWeek .. Year]
     AND: LWeekly >= 0.01
     AND: Edit = No
     (StrWkly < 50) AND INVOLVING(StrPd,StrAmt)
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     StrPd <> Note
```

^I Editor: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^N

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: PAskStruc IN [2 .. 3]
     AND: StrOths = Yes
     RESERVECHECK
```

RESERVECHECK

FRS0605B.QInsur (continued)

Questions about structure insurance.

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *QInsur.QStructure[1].StrWkly = RESPONSE*
AND: *QOwner1.QMortgage.M[1].MorIWkly = RESPONSE*
(QInsur.QStructure[1].StrWkly < QOwner1.QMortgage.M[1].MorIWkly) AND INVOLVING(QOwner1.QMortgage.M[1].MorInPay,QInsur.QStructure[1].StrAmt, QInsur.QStructure[1].StrPd)

^I The amount you recorded for the premium on the insurance on the structure is greater than the amount recorded for the last mortgage payment.
Please check whether this is correct.^I

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *QInsur.QStructure[1].StrWkly = RESPONSE*
AND: *QOwner1.QMortgage.M[1].IntPWkly = RESPONSE*
(QInsur.QStructure[1].StrWkly < QOwner1.QMortgage.M[1].IntPWkly) AND INVOLVING(QOwner1.QMortgage.M[1].IntPrPay,QInsur.QStructure[1].StrAmt, QInsur.QStructure[1].StrPd)

^I The amount you recorded for the premium on the insurance on the structure is greater than the amount recorded for the last mortgage payment.
Please check whether this is correct.^I

FRS0605B.QCounTax

***COMPUTE IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NatCen <> NI*

BandAMax := 1045

***COMPUTE IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NatCen <> NI*

BandBMax := 1220

***COMPUTE IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NatCen <> NI*

BandCMax := 1395

***COMPUTE IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NatCen <> NI*

BandDMax := 1568

***COMPUTE IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NatCen <> NI*

BandEMax := 1916

***COMPUTE IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NatCen <> NI*

BandFMax := 2265

***COMPUTE IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NatCen <> NI*

BandGMax := 2613

***COMPUTE IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NatCen <> NI*

BandHMax := 3136

***COMPUTE IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NatCen <> NI*

BandIMax := 3140

***COMPUTE IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NatCen <> NI*

BandAMin := 430

***COMPUTE IF:** QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
***AND:** NatCen <> NI*

CTIntro := (B + ' Now there are some questions about Council Tax' + B)

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

FRS0605B.QCounTax.CTConDoc

^I QCounTax^I
^N^CTIntro
For your Council Tax, do you have a bill, or a payment book that you could consult?^N

^I^BLU^IC Accept a statement/bill from the year 2005-2006 if no payment for 2006-2007 yet made.^I

(1)	Yes	Yes - consulted now
(2)	No	No - no document (or will not consult)

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*

FRS0605B.QCounTax.CTBand

^I QCounTax^I
^N Could you please tell me which Council Tax band this accommodation is in?
^I^BLU^IC This must be the band given by the council - do not accept respondent's own estimate of value of property.
Council Tax band I exists in Wales for properties over £400,000.
If this household's accommodation is not valued separately (eg. because it's a rented part of larger premises), then use code 10.

If respondents initial band allocation was later changed because they are disabled enter^B original^B band here.^I

(1)	BandA	Band A
(2)	BandB	Band B
(3)	BandC	Band C
(4)	BandD	Band D
(5)	BandE	Band E
(6)	BandF	Band F
(7)	BandG	Band G
(8)	BandH	Band H
(9)	BandI	Band I
(10)	NotApp	Household accommodation not valued separately

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *Wales <> Yes*
CTBand <> BandI

^I This is not a Welsh property - that code is invalid.^I

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    AND: *NatCen <> NI*
    RESERVECHECK

RESERVECHECK

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    AND: *NatCen <> NI*

CTXAmt := 'Missing'

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    AND: *NatCen <> NI*

CTSXAmt := 'Missing'

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    AND: *NatCen <> NI*

CTSWAmt := 'Missing'

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    AND: *NatCen <> NI*

CTSSAmt := 'Missing'

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    AND: *NatCen <> NI*
    AND: *Edit = Yes*
    AND: *CTConDoc <> EMPTY*
    CTBand = RESPONSE

^I Editor: The Council Tax Band is missing. The Local Authority will need to be telephoned. Consult the fact sheet & telephone them.^I

RECORD IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    AND: *NatCen <> NI*

FRS0605B.QCounTax.CTValid

^I QCounTax^I

STRING[2]

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    AND: *NatCen <> NI*
    AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*

Letters[1] := 'A'

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    AND: *NatCen <> NI*
    AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*

Letters[2] := 'B'

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
    AND: *NatCen <> NI*
    AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*

Letters[3] := 'C'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
```

Letters[4] := 'D'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
```

Letters[5] := 'E'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
```

Letters[6] := 'F'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
```

Letters[7] := 'G'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
```

Letters[8] := 'H'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
```

Letters[9] := 'I'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTBand IN [BandA .. BandI]
```

Letter := Letters[ORD(CTBand)]

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTBand = NotApp
```

Letter := 'Not valued separately'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTBand = DONTKNOW
```

Letter := 'Don't know'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTBand = REFUSAL
```

Letter := 'Missing'

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
```

FRS0605B.QCounTax.CTLVBand

^I QCounTax^I
^N Was your Council Tax bill reduced to a lower band because there is a disabled person in the household?^N

^I^BLU^IC Households must make a special application in order to obtain this reduction.

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: (CTLVBand = Yes) AND (CTBand IN [BandA .. BandI])
```

FRS0605B.QCounTax.CTLVChk

^I QCounTax^I
^N You said you were in Band ^Letter; is that the Band^B after^B this lower valuation, or^B before^B?^N

(1)	Aftr	After lower valuation
(2)	Befor	Before

```
CHECK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: (CTLVBand = Yes) AND (CTBand IN [BandA .. BandI])
     AND: CTBand = BandI
     CTLVChk <> Aftr
```

^I Band I is the HIGHEST band, so it cannot be the band AFTER the lower valuation. Please change one or the other.^I

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: Scotland = Yes
```

ScotFill := (' including^B Domestic water & sewerage' + ' charges^B')

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: NOT (Scotland = Yes)
```

ScotFill := ''

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*

FRS0605B.QCounTax.CTAmt

^I QCounTax^I

^I If the respondent has not paid any tax for any reason then enter 0 and the later questions will probe the reasons.^I

0.00..9999.97

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*
AND: *Scotland = Yes*
CTAmt <> 0

^I No Council Tax paid, but you should record here the last payment of domestic water charge and domestic sewerage charge, which are not paid for by C. Tax benefit. If they have not been paid, suppress warning and continue.^I

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*
AND: *CTAmt = NONRESPONSE*

HMissVar := (HMissVar + 1)

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*
AND: *(CTAmt > 0) OR CTAmt = NONRESPONSE*

FRS0605B.QCounTax.CTInstal

^I QCounTax^I
^N^O1 Can I just check,^O2 Was that the full payment for the year, or was it an instalment?^N

^I^BLU^IC 'Year' = April to March (12 months).^N

(1)	Full	Full annual payment
(2)	Instal	An instalment

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*
AND: *(CTAmt > 0) OR CTAmt = NONRESPONSE*
AND: *CTInstal = Instal*

FRS0605B.QCounTax.CTTime

^I QCounTax^I
^N How many instalments are there, over the whole year?^N

^I^BLU^IC 'whole year' = april to march (12 months).
if payment given is from^B last^B year, enter number of instalments made last year.^I

2..52

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: (CTAmt > 0) OR CTAmt = NONRESPONSE
     AND: CTInstal = Instal
     AND: (CTConDoc = Yes) AND (CTTime = RESPONSE)
```

FRS0605B.QCounTax.CTAnnual

^I QCounTax^I
^I^BLU^IC Refer to document being consulted:^BLU^I

^N On the statement/bill, what is the^B total^B amount payable for the year,^ScotFill after deducting any discounts or benefit?^N

^I^BLU^IC 'Year' = April to March (12 months)^I

0.00..9999.97

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: (CTAmt = RESPONSE) AND (CTInstal = RESPONSE)
     AND: CTInstal = Full
```

CTAmtYr := CTAmt

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: (CTAmt = RESPONSE) AND (CTInstal = RESPONSE)
     AND: (CTInstal = Instal) AND (CTTime = RESPONSE)
```

CTAmtYr := (CTAmt * CTTime)

```
RECORD IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
```

FRS0605B.QCounTax.CWat1Rs

^I QCounTax^I
^I In Scotland, Domestic Water Charge should be^B included^B in the total Council Tax bill for the year - if not, please explain in a note.^I

(1)	Passed	Passed
(2)	Hard	Hard
(3)	Soft	Soft
(4)	Suppress	Suppressed

```
RECORD IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
```

FRS0605B.QCounTax.CWat1Ex

^I QCounTax^I
^I^BLU^IC^SuppTxt^I

OPEN

RECORD IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*

FRS0605B.QCounTax.CSew1Rs

^I QCounTax^I
^I In Scotland, Domestic Sewerage Charge should be^B included^B in the total Council Tax bill for the year - if not, please explain in a note.^I

(1)	Passed	Passed
(2)	Hard	Hard
(3)	Soft	Soft
(4)	Suppress	Suppressed

RECORD IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*

FRS0605B.QCounTax.CSew1Ex

^I QCounTax^I
^I^BLU^IC^SuppTxt^I^BLU^IC

OPEN

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*
AND: *(((Edit = No) AND (Scotland = Yes)) AND (CTConDoc = Yes)) AND ((CTInstal = Full) OR (CTAnnual > 0))*

FRS0605B.QCounTax.CWatAmt1

^I QCounTax^I
^N How much is the annual Domestic Water Charge, as shown on the bill?^N

^I^BLU^IC Enter the full charge, before any status discount.^I

0.00..9999.97

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*
AND: *(((Edit = No) AND (Scotland = Yes)) AND (CTConDoc = Yes)) AND ((CTInstal = Full) OR (CTAnnual > 0))*
AND: *CWatAmt1 = RESPONSE*
CWatAmt1 > 0

CWatAmt1 > 0

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*
AND: *(((Edit = No) AND (Scotland = Yes)) AND (CTConDoc = Yes)) AND ((CTInstal = Full) OR (CTAnnual > 0))*
AND: *CWatAmt1 = RESPONSE*
AND: *(CWat1Rs = Suppressed) OR CWat1Ex <> EMPTY*

FRS0605B.QCounTax.CWat1Ex

^I QCounTax^I
^I^BLU^IC^SuppTxt^I

OPEN

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*
AND: *(((Edit = No) AND (Scotland = Yes)) AND (CTConDoc = Yes)) AND ((CTInstal = Full) OR (CTAnnual > 0))*
AND: *CWatAmt1 = RESPONSE*

FRS0605B.QCounTax.CSewAmt1

^I QCounTax^I
^N How much is the annual Domestic Sewerage Charge, as shown on the bill?^N

^I^BLU^IC Enter the full charge, before any status discount or transitional relief.^I

0.00..9999.97

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*
AND: *(((Edit = No) AND (Scotland = Yes)) AND (CTConDoc = Yes)) AND ((CTInstal = Full) OR (CTAnnual > 0))*
AND: *CWatAmt1 = RESPONSE*
AND: *CSewAmt1 = RESPONSE*
CSewAmt1 > 0

CSewAmt1 > 0

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*
AND: *(((Edit = No) AND (Scotland = Yes)) AND (CTConDoc = Yes)) AND ((CTInstal = Full) OR (CTAnnual > 0))*
AND: *CWatAmt1 = RESPONSE*
AND: *CSewAmt1 = RESPONSE*
AND: *(CSew1Rs = Suppressed) OR CSew1Ex <> EMPTY*

FRS0605B.QCounTax.CSew1Ex

^I QCounTax^I
^I^BLU^IC^SuppTxt^I^BLU^IC

OPEN

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: ((Edit = Yes) AND (Scotland = Yes)) AND (CTConDoc = Yes)
```

FRS0605B.QCounTax.CWatAmt1

^I QCounTax^I
^N How much is the annual Domestic Water Charge, as shown on the bill?^N

^I^BLU^IC Enter the full charge, before any status discount.^I

0.00..9999.97

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: ((Edit = Yes) AND (Scotland = Yes)) AND (CTConDoc = Yes)
```

FRS0605B.QCounTax.CSewAmt1

^I QCounTax^I
^N How much is the annual Domestic Sewerage Charge, as shown on the bill?^N

^I^BLU^IC Enter the full charge, before any status discount or transitional relief.^I

0.00..9999.97

```
RECORD IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
```

FRS0605B.QCounTax.CTRebPx

^I QCounTax^I
^I^BLU^IC^Pd97Txt^I

OPEN

```
RECORD IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
```

FRS0605B.QCounTax.CTRebRs

^I QCounTax^I
^I That seems rather high. Please check the amount and frequency of payment. If correct, suppress warning and explain circumstances in a note.^I

(1)	Passed	Passed
(2)	Hard	Hard
(3)	Soft	Soft
(4)	Suppress	Suppressed

RECORD IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*

FRS0605B.QCounTax.CTRebEx

^I QCounTax^I
^I^BLU^IC^SuppTxt^I

OPEN

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*

FRS0605B.QCounTax.CTReb

^I QCounTax^I
^N Are you allowed Council Tax Benefit or rebate, to help pay your Council Tax?^N

(1)	Yes	Yes
(2)	No	No

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*
AND: *CTReb = No*
AND: *(CTAmt = 0) OR CTAmt = NONRESPONSE*

FRS0605B.QCounTax.WhyNoCT

^I QCounTax^I
^I^BLU^IC No council tax is paid, but no benefit received.
Ask or code: what was the reason for paying no council tax?^I

(1)	NotRec	Bill not yet^B received^B and household not previously liable for C.Tax
(2)	NotPaid	Bill not yet^B paid^B and household not previously liable for C.Tax
(3)	Delib	Deliberate non-payment, in dispute, appeal, etc.
(4)	Moved	Household only recently moved into accommodation
(5)	Exempt	Household has a 'formal exemption' from the Tax (all students; MoD property; severely mentally impaired.)
(6)	Other	Other reason (DESCRIBE IN A NOTE)

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*
AND: *CTReb = No*

FRS0605B.QCounTax.CTBWait

^I QCounTax^I
^N Are you awaiting the outcome of a claim for Council Tax benefit or rebate?^N

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
```

FRS0605B.QCounTax.CTRebAmt

^I QCounTax^I
^N How much was allowed?^N

0.00..9999.97

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
```

FRS0605B.QCounTax.CTRebPd

^I QCounTax^I
^N How long did this cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
     AND: CTRebPd = Note
```

FRS0605B.QCounTax.CTRebPx

^I QCounTax^I
^I^BLU^IC^Pd97Txt^I

OPEN

FRS0605B.QCounTax.Weekly()

Procedure Call

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
```

PdConW[1] := 1

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
```

PdConW[2] := 2

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
```

PdConW[3] := 3

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
```

PdConW[4] := 4

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
```

PdConW[52] := 52

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0605B.QCounTax (continued)

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
     AND: CTRebPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

CTRWkly := LWeekly

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
     AND: CTRebPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

CTRebYr := (CTRWkly * 52)

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
     AND: CTRebPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (CTRebYr > 0) AND (CTBand = RESPONSE)
     ((((((((((CTBand = BandA) AND (CTRebYr <= BandAMax)) OR ((CTBand =
     BandB) AND (CTRebYr <= BandBMax))) OR ((CTBand = BandC) AND (CTRebYr
     <= BandCMax))) OR ((CTBand = BandD) AND (CTRebYr <= BandDMax))) OR
     ((CTBand = BandE) AND (CTRebYr <= BandEMax))) OR ((C
```

((((((((((CTBand = BandA) AND (CTRebYr <= BandAMax)) OR ((CTBand = BandB) AND (CTRebYr <= BandBMax))) OR ((CTBand = BandC) AND (CTRebYr <= BandCMax))) OR ((CTBand = BandD) AND (CTRebYr <= BandDMax))) OR ((CTBand = BandE) AND (CTRebYr <= BandEMax))) OR ((CTBand = BandF) AND (CTRebYr <= BandFMax))) OR ((CTBand = BandG) AND (CTRebYr <= BandGMax))) OR ((CTBand = BandH) AND (CTRebYr <= BandHMax))) OR ((CTBand = BandI) AND (CTRebYr <= BandIMax))) AND INVOLVING (CTBand, CTRebPd, CTRebAmt)

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
     AND: CTRebAmt > 0
     AND: CTRebPd IN [OneWeek .. Year]
     AND: LWeekly > 0
     AND: (CTRebYr > 0) AND (CTBand = RESPONSE)
     AND: (CTRebRs = Suppressed) OR CTRebEx <> EMPTY
```

FRS0605B.QCounTax.CTRebEx

^I QCounTax^I
^I^BLU^IC^SuppTxt^I

OPEN

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: (((NewBU >= 2) AND (CTConDoc = Yes)) AND (CTReb = Yes)) AND
     (NotHRPBU = 1)
```

FRS0605B.QCounTax.WhoseCTB

^I QCounTax^I
^N According to the statement, who is the Council Tax Benefit for?^N

^I^BLU^IC Code all that apply.^I

SET [7] OF

(1)	p1	^BUAdName[1]
(2)	p2	^BUAdName[2]
(3)	p3	^BUAdName[3]
(4)	p4	^BUAdName[4]
(5)	p5	^BUAdName[5]
(6)	p6	^BUAdName[6]
(7)	p7	^BUAdName[7]
(8)	Oth	Someone else (SPECIFY IN A NOTE)
(9)	NS	Not on statement

```
CHECK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: (((NewBU >= 2) AND (CTConDoc = Yes)) AND (CTReb = Yes)) AND
     (NotHRPBU = 1)
     AND: In loop FOR Index := 1 TO 7
     AND: Index IN WhoseCTB
     BUAdName[[Index] <> ''
```

Code ^Index is not valid for this question.

```
CHECK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: (((NewBU >= 2) AND (CTConDoc = Yes)) AND (CTReb = Yes)) AND
     (NotHRPBU = 1)
     AND: NS IN WhoseCTB
     WhoseCTB.CARDINAL = 1
```

'Not known/not on statement' is an exclusive code!

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTReb = Yes
```

are := 'In addition to your rebate/ benefit, are'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: NOT (CTReb = Yes)
```

are := 'Are'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: NatCen <> NI
```

SHOWCARD := (IS + ' O')

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: NOT (NatCen <> NI)
```

SHOWCARD := ''

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
```

FRS0605B.QCounTax.CTDisc

^I QCounTax^I

^I The council tax assumes two adults per household. If only one adult lives there, a 25% status discount will usually apply. If a student/student nurse/apprentice etc. lives with^B one^B other adult then a discount of 25% will apply. However no discount will apply if that person lives with two or more other adults.^I

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: (CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE
     AND: CTDisc = Yes
```

FRS0605B.QCounTax.CT25D50D

^I QCounTax^I
^I Some households get a discount on their Council Tax because of the^B type^B of person living there. Usually this is a 25% discount. The rules are very detailed, but in general:

25% for Single adult households

25% for one adult, plus: a student/student nurse/person under 18/apprentice/YT Trainee/care worker/severely mentally impaired

50% when a household is made up of a mixture of those who are exempt (eg. student/student nurse/person under 18/apprentice/YT Trainee/care worker/severely mentally impaired)

Note: Households wholly occupied by students or under 18 year olds are entirely^B exempt^B from Council Tax.

(1)	D25	25%
(2)	D50	50%

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*
AND: *CTDisc = Yes*
CT25D50D <> D50

^I Are you sure? Discount is usually 25%. The 50% discount applies only if ALL household members belong to the groups shown on ^SHOWCARD. Please check with respondent. If discount IS DEFINITELY 50%, suppress warning and continue.^I

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTBand IN [BandA .. BandI]) OR CTBand = NONRESPONSE*
AND: *AllAd = 1*
(CTDisc = Yes) AND (CT25D50D = D25)

^I Are you sure? Households with only one adult would normally have a status discount (25% reduction of the bill).^I

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *(CTAmtYr > 0) AND (CTBand = RESPONSE)*
(((((((((CTBand = BandA) AND (CTAmtYr <= BandAMax)) OR ((CTBand = BandB) AND (CTAmtYr <= BandBMax))) OR ((CTBand = BandC) AND (CTAmtYr <= BandCMax))) OR ((CTBand = BandD) AND (CTAmtYr <= BandDMax))) OR ((CTBand = BandE) AND (CTAmtYr <= BandEMax))) OR ((C

^I That's £^CTAmtYr a year which seems rather high for a property in this Band. Please check the amount and frequency of payment. If correct, suppress warning and explain circumstances in a Note.^I

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: ((CTAmt > 0) AND (CTDisc <> Yes)) AND (CTReb <> Yes)
     ((CTInstal = Instal) AND ((CTAmt * CTTime) >= BandAMin)) OR ((CTInstal
     = Full) AND (CTAmt > BandAMin))
```

^I The annual Council Tax comes to less than the cheapest Council Tax. No discount or rebate is received, so please check for a typing error. If correct, please give explanation in a Note.^I

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: ((CTAmt = RESPONSE) AND (CTTime = RESPONSE)) AND (CTAnnual =
     RESPONSE)
```

CTReal := (CTAmt * CTTime)

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: ((CTAmt = RESPONSE) AND (CTTime = RESPONSE)) AND (CTAnnual =
     RESPONSE)
     CTAnnual < (1.2 * CTReal)
```

^I £^CTAnnual is more than would be expected, given the instalments mentioned earlier. Please check, from the document consulted, that it's for the^B same year^B as the instalments.^I

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: ((CTAmt = RESPONSE) AND (CTTime = RESPONSE)) AND (CTAnnual =
     RESPONSE)
     CTAnnual > (0.8 * CTReal)
```

^I £^CTAnnual is less than would be expected, given the instalments mentioned earlier. Please check, from the document consulted, that it's for the^B same year^B as the instalments.^I

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     (CTBand <> NotApp) AND (CTBand <> DONTKNOW)
```

^I^BLU^IC If necessary check which is correct; this accom:

- is NOT valued separately for C.Tax (code 9),
- or it IS valued for C.Tax, but respondent DOESN'T KNOW the Tax Band
(enter Don't know).

If correct, suppress check & continue.^I

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: ((((CTAmt = RESPONSE) AND (CTTime = RESPONSE)) AND (CTInstal =
     Instal)) AND (CTAnnual = RESPONSE)) AND (CWatAmt1 = RESPONSE)
```

CTReal := (CTAmt * CTTime)

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: ((((CTAmt = RESPONSE) AND (CTTime = RESPONSE)) AND (CTInstal =
     Instal)) AND (CTAnnual = RESPONSE)) AND (CWatAmt1 = RESPONSE)
     CWatAmt1 <= CTReal
```

^I The Domestic^B Water^B charge is more than the total amount of Council Tax paid for the year. Make sure that the last payment of Domestic Water charge (and domestic sewerage charge) was included at CTAmt.^I

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *((CWatAmt1 = RESPONSE) AND (CTAmt = RESPONSE)) AND (CTInstal = Full)*
CWatAmt1 <= CTAmt

^I The Domestic^B Water^B Charge is more than the total amount of Council Tax paid for the year. Make sure that the last payment of Domestic Water charge (and domestic sewerage charge) was included at CTAmt.^I

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *CTReb <> EMPTY AND (CTRebAmt = RESPONSE)*
CTRebAmt <> 0

^I Zero amount of Council Tax benefit: this contradicts previous answer (at 'CTReb') that benefit WAS received. Please resolve if possible.^I

RECORD IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *Edit = Yes*

FRS0605B.QCounTax.OrgWatAmt

^I QCounTax^I

^N Domestic Water Charge, original entry before discount.^N

0.00..9999.97

RECORD IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *Edit = Yes*

FRS0605B.QCounTax.OrgSewAmt

^I QCounTax^I
^N Domestic Sewerage Charge, original entry before discount as entered at interview.^N

0.00..9999.97

DISPLAY IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *Edit = Yes*
AND: *Scotland = Yes*

FRS0605B.QCounTax.OrgWatAmt

^I QCounTax^I

^N Domestic Water Charge, original entry before discount.^N

0.00..9999.97

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: Scotland = Yes
```

CTDiscR := 1

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: Scotland = Yes
     AND: CT25D50D = D25
```

CTDiscR := 0.75

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: Scotland = Yes
     AND: CT25D50D = D50
```

CTDiscR := 0.5

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: Scotland = Yes
```

CWatAmt := (OrgWatAmt * CTDiscR)

```
DISPLAY IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: Scotland = Yes
```

FRS0605B.QCounTax.CWatAmt

^I QCounTax^I
^N Water charge: Final value (after discount):^N

0.00..9999.97

```
DISPLAY IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: Scotland = Yes
```

FRS0605B.QCounTax.OrgSewAmt

^I QCounTax^I
^N Domestic Sewerage Charge, original entry before discount as entered at interview.^N

0.00..9999.97

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: Scotland = Yes
```

CSewAmt := (OrgSewAmt * CTDiscR)

DISPLAY IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
 AND: *NatCen <> NI*
 AND: *Edit = Yes*
 AND: *Scotland = Yes*

FRS0605B.QCounTax.CSewAmt

^I QCounTax^I
^N Sewerage charge: Final value (after discount):^N

0.00..9999.97

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
 AND: *NatCen <> NI*
 AND: *Edit = Yes*

CTChkB := ''

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
 AND: *NatCen <> NI*
 AND: *Edit = Yes*

CTChkC := 'Not known'

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
 AND: *NatCen <> NI*
 AND: *Edit = Yes*

CTChkD := ''

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
 AND: *NatCen <> NI*
 AND: *Edit = Yes*

CTChkE := 'N/A'

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
 AND: *NatCen <> NI*
 AND: *Edit = Yes*

CTChkF := 'Not calculated'

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
 AND: *NatCen <> NI*
 AND: *Edit = Yes*
 AND: *CTAnnual = RESPONSE*

CTChkCR := CTAnnual

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
 AND: *NatCen <> NI*
 AND: *Edit = Yes*
 AND: *CTAnnual = RESPONSE*

CTChkC := STR(CTAnnual,7,2)

COMPUTE IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
 AND: *NatCen <> NI*
 AND: *Edit = Yes*
 AND: *((CTAmt = RESPONSE) AND CTInstal <> NONRESPONSE) AND CTTime <> NONRESPONSE*
 AND: *(CTInstal = Full) OR (CTAmt = 0)*

CTChkCR := CTAmt

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: ((CTAmt = RESPONSE) AND CTInstal <> NONRESPONSE) AND CTTime <>
     NONRESPONSE
     AND: (CTInstal = Full) OR (CTAmt = 0)
```

CTChkC := STR(CTAmt,7,2)

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: ((CTAmt = RESPONSE) AND CTInstal <> NONRESPONSE) AND CTTime <>
     NONRESPONSE
     AND: CTInstal = Instal
```

CTChkCR := (CTAmt * CTTime)

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: ((CTAmt = RESPONSE) AND CTInstal <> NONRESPONSE) AND CTTime <>
     NONRESPONSE
     AND: CTInstal = Instal
```

CTChkC := STR(CTAmt * CTTime,7,2)

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: CTDisc = No
```

CTChkD := 'None'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: CT25D50D = D25
```

CTChkD := '25%'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: CT25D50D = D50
```

CTChkD := '50%'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: CT25D50D = NONRESPONSE
```

CTChkD := 'Amount not known'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: (CTRebAmt = RESPONSE) AND (CTRebPd = RESPONSE)
```

CTChkE := STR(CTRWkly * 52,7,2)

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: CTRebAmt = NONRESPONSE OR CTRebPd = NONRESPONSE
```

CTChkE := 'Annual amount not known'

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: ((((CTAmt = RESPONSE) AND CTInstal <> NONRESPONSE) AND CTTime <>
     NONRESPONSE) AND CTRebAmt <> NONRESPONSE) AND CTRebPd <> NONRESPONSE
```

CTChkFR := (CTChkCR + (CTRWkly * 52))

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: ((((CTAmt = RESPONSE) AND CTInstal <> NONRESPONSE) AND CTTime <>
     NONRESPONSE) AND CTRebAmt <> NONRESPONSE) AND CTRebPd <> NONRESPONSE
     AND: CT25D50D = D25
```

CTChkFR := ((CTChkFR * 4) / 3)

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: ((((CTAmt = RESPONSE) AND CTInstal <> NONRESPONSE) AND CTTime <>
     NONRESPONSE) AND CTRebAmt <> NONRESPONSE) AND CTRebPd <> NONRESPONSE
     AND: CT25D50D = D50
```

CTChkFR := (CTChkFR * 2)

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: ((((CTAmt = RESPONSE) AND CTInstal <> NONRESPONSE) AND CTTime <>
     NONRESPONSE) AND CTRebAmt <> NONRESPONSE) AND CTRebPd <> NONRESPONSE
```

CTChkF := STR(CTChkFR,7,2)

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *Edit = Yes*

FRS0605B.QCounTax.CTChk

^I QCounTax^I
^I Editor: The following calculations are based on the council tax details.
Local Authority @|@|@|: ^QDataBag.SLA
CTBand @|@|@|@|: ^Letter
Annual set charge (from lookup)@|: £^CTXAmt

SCOTLAND ONLY (from lookup):
Taking off water/sewerage charges@|: £^CTSXAmt @|(Water charges: ^CTSWAmt,@|Sewer charges: ^CTSSAmt)

Respondent's annual payment@|: £^CTChkc

Discount@|@|@|@|: ^CTChkd
Annual benefit received@|@|: £^CTChke
EXPECTED annual charge@|@|: £^CTChkf

PRESS <Enter> To continue.^I

STRING[1]

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *Edit = Yes*
(CTConDoc <> No) AND INVOLVING(CTChk)

^I Editor: No Council Tax document consulted
Examine display at 'CTChk' for discrepancies.
Compare set charge with expected charge (Should be v. similar) and refer to supervisor if necessary.^I

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *Edit = Yes*
CTAmt<>NONRESPONSE

^I Missing amount of council tax. Refer to display at 'ctchk' and enter annual payment, after taking off discount/rebate (Also check for notes). If in Scotland, and discount applies, then see edit instructions.^I

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *Edit = Yes*
AND: *CTAmt = RESPONSE*
CTAmt <> 0

^I Editor: Zero Council Tax recorded. Please check the details as necessary^I

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *Edit = Yes*
CTTime<>NONRESPONSE

^I Missing Period for Council Tax.^I

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     CTRebAmt<>NONRESPONSE
```

^I Missing amount for Council Tax rebate.^I

Editor: For new claims made from April 1999, the maximum rebate for bands F, G & H is the band^B E^B total.^I

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     CTRebPd<>NONRESPONSE
```

^I Missing period for Council Tax Rebate.^I

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: (CTBand = RESPONSE) AND CTLVChk <> EMPTY
     CTLVChk <> Aftr
```

^I Editor: The CT band is the band^B after^B Disablement re-valuation. But it should be the band^B before^B. Please change '^B CTBand^B' to the^B next band up^B (Eg. from 'C' to 'D') and then change the answer at '^B CTLVChk^B' to code 2, 'before'.^I

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: Edit = Yes
     CTRebPd <> Note
```

^I EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^I

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: (CTRebYr > 0) AND (CTBand = RESPONSE)
     (((((CTBand = BandA) AND (CTRebYr <= BandAMax)) OR ((CTBand = BandB)
     AND (CTRebYr <= BandBMax))) OR ((CTBand = BandC) AND (CTRebYr <=
     BandCMax))) OR ((CTBand = BandD) AND (CTRebYr <= BandDMax))) OR
     ((IN(CTBand,[???])) AND (CTRebYr <= BandEMax))
```

^I Editor: CT rebate is £^CTRebYr a year. For new claims made from April 1998, the maximum rebate for bands F, G & H is the band^B E^B total. For claims before April 1998 this capping does not apply.^I

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: (CTRebYr > 0) AND (CTBand <> RESPONSE)
     CTRebYr <= BandHMax
```

^I Editor: Council Tax rebate is £^CTRebYr a year which is greater than even the highest council tax rebate allowed. Please check the amount and period of payment.^I

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     WhyNoCT <> Other
```

^I Editor: Other reason for CT non-payment. Please check for a note; and see
(a) If the reason can be re-coded, at '^B WHYNOCT^B' {or at '^B CTEXREB^B}';
Or
(b) If the hhold is in fact not liable for CT - Eg. note states 'paid to landlord' or 'included in rent' (If so, change '^B CTBand^B' to '9').^I

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: Scotland = Yes
     AND: (CTInstal = Full) OR (CTAnnual > 0)
     CWatAmt1 = RESPONSE
```

^I Editor: Missing amount for Domestic Water Charge (Scotland): Please impute using figures provided.^I

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NatCen <> NI
     AND: Edit = Yes
     AND: Scotland = Yes
     AND: (CTInstal = Full) OR (CTAnnual > 0)
     CSewAmt1 = RESPONSE
```

^I Editor: Missing amount for domestic sewerage charge (Scotland): Please impute using figures provided.^I

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
QCounTax.WhyNoCT <> Other

^I If Tax is 'paid by landlord' or 'included in rent', check whether the accommodation is valued separately and if not, recode CTBand;
if a deduction is made from pay, enter it as a payment at CTAmt.^I

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NatCen <> NI*
AND: *QCounTax.CTReb <> EMPTY AND ((QRenting.HBenefit = Yes) OR (GOV IN QOwner1.QMortgage.M[1].QOutsPay))*
QCounTax.CTReb = Yes

^I Earlier, the respondent said they get Housing Benefit or help from ^GOV1 with mortgage. They should usually also get Council Tax Benefit. Please check: is their C.Tax bill 'reduced' - does the Council take anything off it? (apart from Discount). If so, change answer to 'Yes'.^I

FRS0605B.QNIRates

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 1
```

NIRate := 3.4857

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 2
```

NIRate := 3.5278

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 3
```

NIRate := 3.7189

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 4
```

NIRate := 3.4017

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 5
```

NIRate := 3.3867

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 6
```

NIRate := 3.5807

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 7
```

NIRate := 3.736

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 8
```

NIRate := 3.5094

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
      AND: NOT (NatCen <> NI)
      AND: NIDCoun = 9
```

NIRate := 3.2167

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
And: *NOT (NatCen <> NI)*
And: *NIDCoun = 10*

NIRate := 3.3567

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
And: *NOT (NatCen <> NI)*
And: *NIDCoun = 11*

NIRate := 3.4432

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
And: *NOT (NatCen <> NI)*
And: *NIDCoun = 12*

NIRate := 3.6182

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
And: *NOT (NatCen <> NI)*
And: *NIDCoun = 13*

NIRate := 3.6516

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
And: *NOT (NatCen <> NI)*
And: *NIDCoun = 14*

NIRate := 3.3551

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
And: *NOT (NatCen <> NI)*
And: *NIDCoun = 15*

NIRate := 3.1797

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
And: *NOT (NatCen <> NI)*
And: *NIDCoun = 16*

NIRate := 3.6637

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
And: *NOT (NatCen <> NI)*
And: *NIDCoun = 17*

NIRate := 3.5046

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
And: *NOT (NatCen <> NI)*
And: *NIDCoun = 18*

NIRate := 3.3255

Compute if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
And: *NOT (NatCen <> NI)*
And: *NIDCoun = 19*

NIRate := 3.7491

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NIDCoun = 20
```

NIRate := 3.1884

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NIDCoun = 21
```

NIRate := 3.775

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NIDCoun = 22
```

NIRate := 3.6567

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NIDCoun = 23
```

NIRate := 3.4836

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NIDCoun = 24
```

NIRate := 3.4617

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NIDCoun = 25
```

NIRate := 3.6293

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NIDCoun = 26
```

NIRate := 3.4721

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
```

PDCode[1] := 52

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
```

PDCode[2] := 26

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
```

PDCode[3] := 17.33

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
```

PDCode[4] := 13

COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
AND: NOT (NatCen <> NI)

PDCode[5] := 12

COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
AND: NOT (NatCen <> NI)

PDCode[7] := 6

COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
AND: NOT (NatCen <> NI)

PDCode[8] := 8

COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
AND: NOT (NatCen <> NI)

PDCode[9] := 9

COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
AND: NOT (NatCen <> NI)

PDCode[10] := 10

COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
AND: NOT (NatCen <> NI)

PDCode[13] := 4

COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
AND: NOT (NatCen <> NI)

PDCode[26] := 2

COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
AND: NOT (NatCen <> NI)

PDCode[52] := 1

COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
AND: NOT (NatCen <> NI)

PDCode[90] := 1

COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
AND: NOT (NatCen <> NI)

PDCode[95] := 1

COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
AND: NOT (NatCen <> NI)

RTIntro := '^N^B Now there are some questions about Rates^B^N '

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*

FRS0605B.QNIRates.BillRate

^N^RTIntro
Do you get a bill for rates on this accommodation?^N

(1)	Yes	Yes
(2)	No	No

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *BillRate = No*

FRS0605B.QNIRates.NoRate

^N Why do you not get a rates bill?^N

(1)	RateInc	Rented accommodation with rates included in rent
(2)	RateFree	Rent/rates free
(3)	RateRbt	Receive rebate
(4)	Other	Other reason (specify)

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *BillRate = No*
AND: *NoRate = Other*

FRS0605B.QNIRates.OthReas

^N Please specify this other reason^N

STRING[100]

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *NOT (BillRate = No)*

FRS0605B.QNIRates.PayRate

^N Do you, or someone in this household, pay the rates bill?^N

(1)	Yes	Yes
(2)	No	No

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *NOT (BillRate = No)*
AND: *PayRate = No*

FRS0605B.QNIRates.NoPay

^N Why don't you pay your rates bill?^N

STRING[100]

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NOT (BillRate = No)
     AND: PayRate = Yes
```

FRS0605B.QNIRates.RTConDoc

^N For your Rates, do you have a bill, or a payment book that you could consult?^N

^I^BLU Accept a statement/bill from the year 2001-2002 if no payment for 2002-2003 yet made.^I

(1)	Yes	Yes - consulted now
(2)	No	No - no document (or will not consult)

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NOT (BillRate = No)
     AND: PayRate = Yes
     AND: RTConDoc = Yes
```

FRS0605B.QNIRates.RTAnnual

^I Refer to document being consulted:^I

^N On the statement/bill, what is the^B total^B amount payable, after deducting any discounts or benefit?^N

^I 'Year' = April to March (12 months)

Note : No rates are payable in February and March each year^I

0.00..9999.97

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: NOT (BillRate = No)
     AND: PayRate = Yes
     AND: RTConDoc = No
```

FRS0605B.QNIRates.EstRTAnn

^N Can you tell me, what is the^B total^B amount of rates payable, after deducting any discounts or benefit?^N

^I^BLU^IC Probe to ensure amount given is as accurate as possible.
If respondent unsure of amount open a note to state value is an estimate.

'Year' = April to March (12 months)

Note : No rates are payable in February and March each year^I

0.00..9999.97

Ask if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
And: *NOT (NatCen <> NI)*
And: *NOT (BillRate = No)*
And: *PayRate = Yes*

FRS0605B.QNIRates.RTInstal

^N (Can I just check,) Was that the full payment for the year, or was it an instalment?^N

^I^BLU^IC 'Year' = April to March (12 months).^I

(1)	Full	Full annual payment
(2)	Instal	An instalment

Record if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
And: *NOT (NatCen <> NI)*
And: *NOT (BillRate = No)*
And: *PayRate = Yes*
And: *RTInstal = Instal*

FRS0605B.QNIRates.RTTimePx

^I^BLU^IC^Pd97Txt^I

OPEN

Ask if: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
And: *NOT (NatCen <> NI)*
And: *NOT (BillRate = No)*
And: *PayRate = Yes*
And: *RTInstal = Instal*

FRS0605B.QNIRates.RTTimePd

^N How often do you pay instalments?^N

^I^BLU^IC 'Whole year' = April to March (12 months).

Note : There are usually 10 monthly instalments per year as February and March are 'free' months.
If payment given is from^B last^B year, enter number of instalments made last year.^I

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *NOT (BillRate = No)*
AND: *PayRate = Yes*
AND: *RTInstal = Instal*
AND: *RTTimePd = Note*

FRS0605B.QNIRates.RTTimePx

^I^BLU^IC^Pd97Txt^I

OPEN

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *NOT (BillRate = No)*
AND: *PayRate = Yes*

FRS0605B.QNIRates.RTReb

^N Are you allowed a Rates Rebate?^N

(1)	Yes	Yes
(2)	No	No

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *(NoRate = RateRbt) OR (RTReb = Yes)*

FRS0605B.QNIRates.RTDeduc

^N Was this deducted from your LAST rates payment?^N

(1)	Yes	Yes
(2)	No	No

ASK IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *(NoRate = RateRbt) OR (RTReb = Yes)*
AND: *RTDeduc = Yes*

FRS0605B.QNIRates.RTRebAmt

^N How much was allowed?^N

0.01..9999.97

RECORD IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *(NoRate = RateRbt) OR (RTReb = Yes)*
AND: *RTDeduc = Yes*

FRS0605B.QNIRates.RTRebPx

^I^BLU^IC^Pd97Txt^I

OPEN

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: (NoRate = RateRbt) OR (RTReb = Yes)
     AND: RTDeduc = Yes
```

FRS0605B.QNIRates.RTRebPd

^N How long did this cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

```
ASK IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: (NoRate = RateRbt) OR (RTReb = Yes)
     AND: RTDeduc = Yes
     AND: RTRebPd = Note
```

FRS0605B.QNIRates.RTRebPx

^I^BLU^IC^Pd97Txt^I

OPEN

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: ((RTAnnual = RESPONSE) AND (RTInstal <> Full)) AND (RTTimePd IN
     [OneWeek .. LessWeek])
```

RTCheck := (RTAnnual * PDCode[ORD(RTTimePd)])

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: ((EstRTAnn = RESPONSE) AND (RTInstal <> Full)) AND (RTTimePd IN
     [OneWeek .. LessWeek])
```

RTCheck := (EstRTAnn * PDCode[ORD(RTTimePd)])

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: (RTAnnual = RESPONSE) AND (RTInstal = Full)
```

RTCheck := RTAnnual

```
COMPUTE IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: (EstRTAnn = RESPONSE) AND (RTInstal = Full)
```

RTCheck := EstRTAnn

```
WARN IF: QAccomdat.HHStat <> EMPTY OR (Edit = Yes)
     AND: NOT (NatCen <> NI)
     AND: QDataBag.NINRV > 0
     AND: (RTAnnual = RESPONSE) OR (EstRTAnn = RESPONSE)
     (RTCheck < (QDataBag.NINRV * NIRate)) AND
     INVOLVING(RTInstal,RTAnnual,EstRTAnn)
```

^I That's £^RTCheck per year for Rates which seems high for a property in this area. Are you sure the Amount of Rates paid and the Period are correct?^I

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *QNIRates.RTReb <> EMPTY AND ((QRenting.HBenefit = Yes) OR (GOV IN QOwner1.QMortgage.M[1].QOutsPay))*
QNIRates.RTReb = Yes

^I Earlier, the respondent said they get Housing Benefit or help from ^GOV1 with mortgage. They should usually also get a Rates Rebate. Please check: is their Rates bill 'reduced' - does the Rate Collection Agency take anything off it? If so, change answer to 'Yes'.^I

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
AND: *QNIRates.BillRate <> EMPTY AND (QNIRates.NoRate = RateInc)*
((QAccomdat.Tenure = Part) OR (QAccomdat.Tenure = Rents)) AND INVOLVING(QAccomdat.Tenure)

^I Earlier, the respondent said they did NOT rent/part rent this accommodation - the rates could NOT be included in the rent!!
Please check that Tenure is correct.^I

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
AND: *NOT (NatCen <> NI)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QAccomdat.HHStat <> EMPTY OR (Edit = Yes)*
RESERVECHECK

RESERVECHECK

COMPUTE IF: *QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
AND: *(QRenting.WSInc = EMPTY OR QRenting.WSInc = NONRESPONSE) OR (QRenting.WSInc IN [Sewer, Neith])*

AskWater := Yes

COMPUTE IF: *QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
AND: *(QRenting.WSInc = EMPTY OR QRenting.WSInc = NONRESPONSE) OR (QRenting.WSInc IN [Water, Neith])*

AskSewer := Yes

FRS0605B.QWaterSew

Questions about sewerage and water rates

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: PAskWater = Yes
```

FRS0605B.QWaterSew.WaterMet

^I QWaterSew
^I
^N Are your water charges metered?^N

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: PAskWater = Yes
```

FRS0605B.QWaterSew.WaterPay

^I QWaterSew
^I
^N Do you pay water rates or charges?^N

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: PAskSewer = Yes
```

FRS0605B.QWaterSew.SewerPay

^I QWaterSew
^I
^N Do you pay sewerage rates or charges?^N

(1)	Yes	Yes
(2)	No	No

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: (WaterPay = Yes) AND (SewerPay = Yes)
```

FRS0605B.QWaterSew.SewSep

^I QWaterSew
^I
^N Do you pay separate or combined water and sewerage rates or charges?^N

(1)	Separate	Separate
(2)	Combined	Combined

ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
AND: (AskWater = Yes) OR (AskSewer = Yes)
AND: (SewSep = Separate) OR ((WaterPay = Yes) AND (SewerPay <> Yes))

FRS0605B.QWaterSew.WatTime

^I QWaterSew
^I
^N How many times a year do you pay water rates or charges?^N

^I^BLU^IC Enter times a year.^I

1..52

ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
AND: (AskWater = Yes) OR (AskSewer = Yes)
AND: (SewSep = Separate) OR ((WaterPay = Yes) AND (SewerPay <> Yes))

FRS0605B.QWaterSew.WatAmt

^I QWaterSew
^I
^N How much did you actually pay last time?^N

0.01..9997.00

ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
AND: (AskWater = Yes) OR (AskSewer = Yes)
AND: (SewSep = Separate) OR ((WaterPay = Yes) AND (SewerPay <> Yes))

FRS0605B.QWaterSew.WatAnul

^I QWaterSew
^I
^N How much is your^B annual^B bill?^N
^I^BLU^IC Code as don't know if respondent has not yet received their annual bill because they are on a water meter.^I

0.01..9997.00

COMPUTE IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
AND: (AskWater = Yes) OR (AskSewer = Yes)
AND: (SewSep = Separate) OR ((WaterPay = Yes) AND (SewerPay <> Yes))
AND: (WatAmt = RESPONSE) AND (WatTime = RESPONSE)

WatWkly := ((WatAmt * WatTime) / 52)

WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
AND: (AskWater = Yes) OR (AskSewer = Yes)
AND: (SewSep = Separate) OR ((WaterPay = Yes) AND (SewerPay <> Yes))
AND: Edit = No
(WatWkly <= 12) AND INVOLVING(WatTime,WatAmt)

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

***WARN IF:** QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
***AND:** (AskWater = Yes) OR (AskSewer = Yes)*
***AND:** (SewSep = Separate) OR ((WaterPay = Yes) AND (SewerPay <> Yes))*
***AND:** Edit = No*
***AND:** ((WatTime = RESPONSE) AND (WatAnul = RESPONSE)) AND (WatAmt = RESPONSE)*
(ABS((WatTime * WatAmt) - WatAnul) <= 25) AND INVOLVING(WatTime,WatAnul,WatAmt)

^I Interviewer: The Annual payment for water rates/charges (WatAnul) is very different from the total for individual payments (WatTime x WatAmt).
Please check these figures.^I

***ASK IF:** QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
***AND:** (AskWater = Yes) OR (AskSewer = Yes)*
***AND:** (SewSep = Separate) OR ((SewerPay = Yes) AND (WaterPay <> Yes))*

FRS0605B.QWaterSew.SewTime

^I QWaterSew
^I
^N How many times a year do you pay sewerage rates or charges?^N

^I^BLU^IC Enter times a year.^I

1..52

***ASK IF:** QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
***AND:** (AskWater = Yes) OR (AskSewer = Yes)*
***AND:** (SewSep = Separate) OR ((SewerPay = Yes) AND (WaterPay <> Yes))*

FRS0605B.QWaterSew.SewAmt

^I QWaterSew
^I
^N How much did you actually pay last time?^N

0.01..9997.00

***ASK IF:** QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
***AND:** (AskWater = Yes) OR (AskSewer = Yes)*
***AND:** (SewSep = Separate) OR ((SewerPay = Yes) AND (WaterPay <> Yes))*

FRS0605B.QWaterSew.SewAnul

^I QWaterSew
^I
^N How much is your^B annual^B bill?^N
^I^BLU^IC Code as don't know if respondent has not yet received their annual bill because they have a septic tank.^I

0.01..9997.00

***COMPUTE IF:** QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
***AND:** (AskWater = Yes) OR (AskSewer = Yes)*
***AND:** (SewSep = Separate) OR ((SewerPay = Yes) AND (WaterPay <> Yes))*
***AND:** (SewAmt = RESPONSE) AND (SewTime = RESPONSE)*

SewWkly := ((SewAmt * SewTime) / 52)

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: (SewSep = Separate) OR ((SewerPay = Yes) AND (WaterPay <> Yes))
     AND: Edit = No
     (SewWkly <= 8) AND INVOLVING(SewTime,SewAmt)
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: (SewSep = Separate) OR ((SewerPay = Yes) AND (WaterPay <> Yes))
     AND: Edit = No
     AND: ((SewTime = RESPONSE) AND (SewAnul = RESPONSE)) AND (SewAmt =
     RESPONSE)
     (ABS((SewTime * SewAmt) - SewAnul) <= 25) AND
     INVOLVING(SewTime,SewAnul,SewAmt)
```

^I Interviewer: The Annual payment for sewerage rates/charges (SewAnul) is very different from the total for individual payments (SewTime x SewAmt).
Please check these figures.^I

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: SewSep = Combined
```

FRS0605B.QWaterSew.WSewTime

^I QWaterSew
^I
^N How many times a year do you pay?^N

^I^BLU^IC Enter times a year.^I

1..52

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: SewSep = Combined
```

FRS0605B.QWaterSew.WSewAmt

^I QWaterSew
^I

^I^B Metered Water^B - Charges made via a water meter should be treated as water rate payments and the last amount actually paid entered.^I

0.01..9997.00

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: SewSep = Combined
```

FRS0605B.QWaterSew.WSewAnul

^I QWaterSew
^I
^N How much is your^B annual^B bill?^N
^I^BLU^IC Code as don't know if respondent has not yet received their annual bill.^I

0.01..9997.00

```
COMPUTE IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: SewSep = Combined
     AND: (WSewAmt = RESPONSE) AND (WSewTime = RESPONSE)
```

WSewWkly := ((WSewAmt * WSewTime) / 52)

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: SewSep = Combined
     AND: Edit = No
     (WSewWkly < 20) AND INVOLVING(WSewTime,WSewAmt)
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: SewSep = Combined
     AND: Edit = No
     AND: ((WSewTime = RESPONSE) AND (WSewAnul = RESPONSE)) AND (WSewAmt =
     RESPONSE)
     (ABS((WSewTime * WSewAmt) - WSewAnul) <= 25) AND
     INVOLVING(WSewTime,WSewAnul,WSewAmt)
```

^I Interviewer: The Annual payment for water/sewerage rates/charges (WSewAnul) is very different from the total for individual payments (WSewTime x WSewAmt).
Please check these figures.^I

```
ASK IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     AND: WaterMet = Yes
```

FRS0605B.QWaterSew.WatRb

^I QWaterSew
^I
^I The vast majority of people have to pay the full water and sewage charges but there are also a few metered who are eligible for assistance under the Vulnerable Groups Scheme (Water Industry Act 1999). Under this scheme people who qualify receive a bill capped at the average charge for their region and do not have to pay the measured charge reflecting their genuine water consumption.^I

(1)	Yes	Yes
(2)	No	No

WARN IF: *QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
AND: *(AskWater = Yes) OR (AskSewer = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
AND: *(AskWater = Yes) OR (AskSewer = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
AND: *(AskWater = Yes) OR (AskSewer = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
AND: *(AskWater = Yes) OR (AskSewer = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)*
AND: *(AskWater = Yes) OR (AskSewer = Yes)*
RESERVECHECK

RESERVECHECK

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: QCounTax.CTBand <> EMPTY AND (Scotland <> Yes)
     AND: (AskWater = Yes) OR (AskSewer = Yes)
     RESERVECHECK
```

RESERVECHECK

FRS0605B.QAccomCharge

Questions on charges with accommodation.

ASK IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*

FRS0605B.QAccomCharge.Charge

^I QAccomCharge
^I
Use the combined charge category when it is not possible for respondents to split out separate amounts for ground rent, service charge and maintenance charge. If you record a combined amount, do not record the separate amounts as well.^I

SET [9] OF

(1)	Ground	Ground Rent
(2)	FeuDuty	Feu duty
(3)	Chief	Chief Rent
(4)	Service	Service charge
(5)	Maint	Compulsory or regular maintenance charges
(6)	SiteRent	Site rent (caravans)
(7)	Factor	Factoring (Payments to a land steward)
(8)	Other	Any other regular payments
(9)	Combined	Combined charges (eg. ground rent, service charge, maintenance charge, factoring etc.)
(10)	None	None of these

WARN IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
AND: *(((Ground IN Charge) AND (Service IN Charge)) AND (Maint IN Charge)) AND (Combined IN Charge)*
ERROR

^I Only code combined charges instead of the separate ground rent, service charge, maintenance charge, factoring etc.^I

ASK IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
AND: *Other IN Charge*

FRS0605B.QAccomCharge.ChargeO

^I QAccomCharge
^I
^N Please specify the other type of payments.^N

STRING[50]

ASK IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
AND: *(Service IN Charge) OR (Combined IN Charge)*

FRS0605B.QAccomCharge.ChIns

^I QAccomCharge
^I
^N Does this service charge include insurance?^N

(1)	Yes	Yes
(2)	No	No

FRS0605B.QAccomCharge.QChargeAmtPd[]

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[1] := 'Ground Rent'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[2] := 'Feu duty'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[3] := 'Chief Rent'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[4] := 'Service charge'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[5] := 'Maintenance charges'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[6] := 'Site rent'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[7] := 'Factoring'

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[8] := PChargeO

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

LCharges[9] := ('Combined charges (eg. ground rent, service charge, ' + 'maintenance charge, factoring etc.)')

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
```

FRS0605B.QAccomCharge.QChargeAmtPd[].ChrgAmt

^I QAccomCharge
^I

^N I would now like to ask about the charges you pay for ^LCharges[PSeq].
How much did you pay last time?^N

0.01..9997.00

```
COMPUTE IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
     AND: ChrgAmt = NONRESPONSE
```

HMissVar := (HMissVar + 1)

```
RECORD IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
     AND: ChrgAmt > 0
```

FRS0605B.QAccomCharge.QChargeAmtPd[].ChrgPx

^I QAccomCharge
^I

^I^BLU^IC ^Pd97Txt^I

OPEN

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
     AND: ChrgAmt > 0
```

FRS0605B.QAccomCharge.QChargeAmtPd[].ChrgPd

^I QAccomCharge
^I

^N How long did this cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

```
ASK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
     AND: ChrgAmt > 0
     AND: ChrgPd = Note
```

FRS0605B.QAccomCharge.QChargeAmtPd[].ChrgPx

^I QAccomCharge
^I

^I^BLU^IC ^Pd97Txt^I

OPEN

```
WARN IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: In loop FOR Idx := 1 TO 9
     AND: Idx IN Charge
     AND: Edit = Yes
     ChrgPd <> Note
```

^I Editor: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^I

FRS0605B.QAccomCharge (continued)

Questions on charges with accommodation.

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: FeuDuty IN Charge
     Scotland = Yes
```

^I Feu duty is only valid for Scottish households.^I

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     AND: None IN Charge
     Charge.CARDINAL = 1
```

^I 'None of these' is an exclusive code for this question.^I

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]
     RESERVECHECK
```

RESERVECHECK

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

CHECK IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
    RESERVECHECK

RESERVECHECK

CHECK IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
    RESERVECHECK

RESERVECHECK

CHECK IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
    RESERVECHECK

RESERVECHECK

CHECK IF: *QAccomdat.Tenure IN [Outright .. Part, RentFree, Squatting]*
    RESERVECHECK

RESERVECHECK

FRS0605B.QLodger

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: In loop FOR count := 1 TO HHSize
     AND: ((PRec[count].Depend IN [Adult .. DepAd]) AND (PRel.PR[count].R IN
     [Child .. NonRel])) AND (ECount < 5)
```

ECount := (ECount + 1)

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: In loop FOR count := 1 TO HHSize
     AND: ((PRec[count].Depend IN [Adult .. DepAd]) AND (PRel.PR[count].R IN
     [Child .. NonRel])) AND (ECount < 5)
```

ELodger[ECount] := count

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
```

BordLodg[count].BenUnit := DMBU[[ELodger[count]]

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
```

BordLodg[count].PersId := ELodger[count]

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRel.PR[ELodger[count]].R IN [FChild, FParent, FSib, GChild ..
     NonRel]
```

Relation := Distant

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRec[ELodger[count]].Depend = DepAd
```

Relation := Skip

COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)
AND: QAccomdat.HHStat = Conv
AND: ECount > 0
AND: In loop FOR count := 1 TO 5
AND: ELodger[count] > 0
AND: PRel.PR[ELodger[count]].R IN [Child .. StChild, ILChild .. StParent, ILParent .. StSib, ILSib]

Relation := Close

COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)
AND: QAccomdat.HHStat = Conv
AND: ECount > 0
AND: In loop FOR count := 1 TO 5
AND: ELodger[count] > 0
AND: PRec[ELodger[count]].Sex = Male

HeShe := 'he'

COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)
AND: QAccomdat.HHStat = Conv
AND: ECount > 0
AND: In loop FOR count := 1 TO 5
AND: ELodger[count] > 0
AND: NOT (PRec[ELodger[count]].Sex = Male)

HeShe := 'she'

COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)
AND: QAccomdat.HHStat = Conv
AND: ECount > 0
AND: In loop FOR count := 1 TO 5
AND: ELodger[count] > 0

LName := DMName[[ELodger[count]]

FRS0605B.QLodger.BordLodg[]

```
RECORD IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
```

FRS0605B.QLodger.BordLodg[].BenUnit

^I QLodger
^I

^N Benefit Unit of respondent.^N

0..7

```
RECORD IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
```

FRS0605B.QLodger.BordLodg[].PersId

^I QLodger
^I

Person number of respondent.

0..14

```
ASK IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
```

FRS0605B.QLodger.BordLodg[].ConvBL

^I QLodger
^I

^N (Can I just check), is ^LName ^N ...^I Running Prompt...^I

(1) Board ^N...a^B boarder:^B that is, someone who pays you a^B rent^B for board AND lodging^N
(2) Lodg ^N...a^B lodger:^B that is, someone who pays you a^B rent for lodging, but not food^N
(3) Neith ^N...or neither of these?^N

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL = Board
```

pay := ' pay for board and lodging'

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL = Lodg
```

pay := ' pay'

```
ASK IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
```

FRS0605B.QLodger.BordLodg[].CvPay

^I QLodger
^I

^N How much rent did ^LName^pay last time it was due, after deducting any Housing Benefit?^N

0.00..997.00

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay = NONRESPONSE
```

HMissVar := (HMissVar + 1)

RECORD IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRelation = Distant*
AND: *ConvBL IN [Board .. Lodg]*
AND: *CvPay > 0*

FRS0605B.QLodger.BordLodg[].CvPx

^I QLodger
^I

^I^BLU^IC^Pd97Txt^I

OPEN

ASK IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *QAccomdat.HHStat = Conv*
AND: *ECount > 0*
AND: *In loop FOR count := 1 TO 5*
AND: *ELodger[count] > 0*
AND: *PRelation = Distant*
AND: *ConvBL IN [Board .. Lodg]*
AND: *CvPay > 0*

FRS0605B.QLodger.BordLodg[].CvPd

^I QLodger
^I

^N How long does that cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

```
ASK IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
     AND: CvPd = Note
```

FRS0605B.QLodger.BordLodg[].CvPx

^I QLodger
^I

^I^BLU^IC^Pd97Txt^I

OPEN

FRS0605B.QLodger.BordLodg[].Weekly()

Procedure Call

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[1] := 1

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[2] := 2

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[3] := 3

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[4] := 4

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[13] := 13

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[26] := 26

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
```

PdConW[52] := 52

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
     AND: (PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
     AND: NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])
```

PWeekly := 0

FRS0605B.QLodger.BordLodg[] (continued)

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: CvPay > 0
     AND: CvPd IN [OneWeek .. Year]
     AND: LWeekly > 0
```

CvWkly := LWeekly

```
ASK IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: PRelation = Distant
     AND: ConvBL IN [Board .. Lodg]
     AND: (ConvBL = Lodg) AND (CvPay > 0)
```

FRS0605B.QLodger.BordLodg[].CvHt

^I QLodger
^I

^N Is HEATING included in that, or is it paid for separately?
^N

(1)	Included	Included
(2)	Separat	Paid for separately

```
WARN IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     CvPd <> Note
```

^N EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^N

FRS0605B.QLodger (continued)

```
WARN IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: QAccomdat.HHStat = Conv
     AND: ECount > 0
     AND: In loop FOR count := 1 TO 5
     AND: ELodger[count] > 0
     AND: Edit = No
     AND: BordLodg[count].CvWkly = RESPONSE
     (BordLodg[count].CvWkly < 119) AND
     INVOLVING(BordLodg[count].CvPd,BordLodg[count].CvPay)
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: QAccomdat.HHStat = Conv
     AND: In loop FOR Loop1 := 1 TO 5
     AND: QLodger.BordLodg[Loop1].CvPay > 0
```

BUHBElig[QLodger.BordLodg[Loop1].BenUnit] := Yes

```
WARN IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: QAccomdat.HHStat = Conv
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: QAccomdat.HHStat = Conv
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: QAccomdat.HHStat = Conv
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: QAccomdat.HHStat = Conv
     RESERVECHECK
```

RESERVECHECK

FRS0605B.QSharer

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO HHSize
     AND: (PRec[count].Depend = Adult) AND (ECount < 8)
```

ECount := (ECount + 1)

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO HHSize
     AND: (PRec[count].Depend = Adult) AND (ECount < 8)
```

ESharer[ECount] := count

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
```

Sharer[count].BenUnit := DMBU[[ESharer[count]]

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
```

Sharer[count].PersId := ESharer[count]

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
```

LName := DMName[[ESharer[count]]

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: PRec[ESharer[count]].Sex = Male
```

HeShe := 'he'

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: NOT (PRec[ESharer[count]].Sex = Male)
```

HeShe := 'she'

FRS0605B.QSharer.Sharer[]

RECORD IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*

FRS0605B.QSharer.Sharer[].BenUnit

^I QSharer
^I

BU number of person

0..7

RECORD IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*

FRS0605B.QSharer.Sharer[].PersId

^I QSharer
^I

Person identifier.

0..14

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*
AND: *PersId = 1*

Preamb := ('Now I'd like to ask how much each of you' + ' pays towards certain things.')

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*
AND: *NOT (PersId = 1)*

Preamb := ''

ASK IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*
AND: *BenUnit > 1*

FRS0605B.QSharer.Sharer[].SRentAmt

^I QSharer
^I

^N How much rent did ^LName pay last time it was due, after deducting any Housing Benefit?^N

0.00..997.00

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*
AND: *BenUnit > 1*
AND: *SRentAmt = NONRESPONSE*

HMissVar := (HMissVar + 1)

RECORD IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*
AND: *BenUnit > 1*
AND: *SRentAmt > 0*

FRS0605B.QSharer.Sharer[].SRentPx

^I QSharer
^I

^I^BLU^IC^Pd97Txt^I

OPEN

ASK IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*
AND: *BenUnit > 1*
AND: *SRentAmt > 0*

FRS0605B.QSharer.Sharer[].SRentPd

^I QSharer
^I

^N How long does that cover?^N

(1)	OneWeek	One week
(2)	TwoWeek	Two weeks
(3)	ThrWeek	Three weeks
(4)	Fourweek	Four weeks
(5)	Month	Calendar month
(7)	TwoMonth	Two Calendar months
(8)	EighYear	Eight times a year
(9)	NineYear	Nine times a year
(10)	TenYear	Ten times a year
(13)	ThrMonth	Three months/13 weeks
(26)	SixMonth	Six months/26 weeks
(52)	Year	One Year/12 months/52 weeks
(90)	LessWeek	Less than one week
(95)	LumpSum	One off/lump sum
(97)	Note	None of these ^I(Explain in a note)^I

ASK IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*
AND: *BenUnit > 1*
AND: *SRentAmt > 0*
AND: *SRentPd = Note*

FRS0605B.QSharer.Sharer[].SRentPx

^I QSharer
^I

^I^BLU^IC^Pd97Txt^I

OPEN

FRS0605B.QSharer.Sharer[].Weekly()

Procedure Call

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[1] := 1

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[2] := 2

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[3] := 3

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[4] := 4

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[5] := 4.333

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[7] := 8.67

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[8] := 6.5

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[9] := 5.78

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[10] := 5.2

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[13] := 13

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[26] := 26

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
```

PdConW[52] := 52

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*
AND: *BenUnit > 1*
AND: *SRentAmt > 0*
AND: *(PAmount > 0) AND (PPeriod IN [OneWeek .. Year])*

PWeekly := (PAmount / PdConW[ORD(PPeriod)])

COMPUTE IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
AND: *In loop FOR count := 1 TO 8*
AND: *ESharer[count] > 0*
AND: *BenUnit > 1*
AND: *SRentAmt > 0*
AND: *NOT ((PAmount > 0) AND (PPeriod IN [OneWeek .. Year])*

PWeekly := 0

FRS0605B.QSharer.Sharer[] (continued)

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit > 1
     AND: SRentAmt > 0
     AND: SRentPd IN [OneWeek .. Year]
     AND: LWeekly >= 0
```

SRntWkly := LWeekly

```
WARN IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     SRentPd <> Note
```

^I EDITOR: Code 97 must be re-coded into existing list.
If you temporarily suppress this check you must come back to resolve it.^I

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: BenUnit = 1
```

How := 'Apart from any rent, how'

```
COMPUTE IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit
     = Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: NOT (BenUnit = 1)
```

How := 'How'

FRS0605B.QSharer (continued)

```
WARN IF: (QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit =
     Yes)
     AND: (QAccomdat.HHStat = Shared) AND (NewBU > 1)
     AND: In loop FOR count := 1 TO 8
     AND: ESharer[count] > 0
     AND: Edit = No
     AND: Sharer[count].SRntWkly = RESPONSE
     (Sharer[count].SRntWkly < 103) AND
     INVOLVING(Sharer[count].SRentPd,Sharer[count].SRentAmt)
```

^I Are you sure? Enter here only the RESPONDENT'S SHARE of the household rent.^I

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

WARN IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
RESERVECHECK

RESERVECHECK

WARN IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
RESERVECHECK

RESERVECHECK

WARN IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
RESERVECHECK

RESERVECHECK

WARN IF: *(QCounTax.CTBand <> EMPTY OR QNIRates.BillRate <> EMPTY) OR (Edit = Yes)*
AND: *(QAccomdat.HHStat = Shared) AND (NewBU > 1)*
RESERVECHECK

RESERVECHECK

FRS0605B.QProperty

Questions about other property

ASK IF: QAccomdat.SubLet = Yes

FRS0605B.QProperty.SubRent

^I QProperty
^I
^N You mentioned earlier that you let, or sub-let, part of this accommodation to someone outside your household.

How much rent have you received from this in the last 12 months, ie. since ^DLYear : that's BEFORE deducting any income tax that might be due on it?^N

0.00..99999.97

```
COMPUTE IF: QAccomdat.SubLet = Yes
      AND: SubRent = RESPONSE
```

SubWkly := (SubRent / 52)

```
WARN IF: QAccomdat.SubLet = Yes
      AND: Edit = No
      (SubWkly < 180) AND INVOLVING(SubRent)
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
COMPUTE IF: QAccomdat.SubLet = Yes
      AND: SubRent = NONRESPONSE
```

HMissVar := (HMissVar + 1)

ASK IF: QAccomdat.SubLet = Yes

FRS0605B.QProperty.SubAllow

^I QProperty
^I
^N And is that BEFORE or AFTER deducting allowable expenses?

(1)	Befor	Before
(2)	Aftr	After

COMPUTE IF: QAccomdat.SubLet = Yes

Im := 'Apart from that, in'

COMPUTE IF: NOT (QAccomdat.SubLet = Yes)

Im := 'In'

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

FRS0605B.QTeleV

Televisions

Ask always:

FRS0605B.QTeleV.TellyPre

^I QTeleV
^I
^N The next questions ask whether you have a television set within your household.^N

(1)	Continue	Continue

Compute always:

```
Does_your := (N + 'Does your')
```

Compute always:

```
a_colour := (N + 'a colour')
```

Ask always:

FRS0605B.QTeleV.ConTV

^I QTeleV
^I
^Does_your household have any of the following items?

...^a_colour TV set?^N
 ^I^BLU
Include items stored but in working order, and items under repair.^I

(1)	One	One only
(2)	More	more than one
(3)	None	none

Compute always:

```
Does_your := 'Does your'
```

Compute always:

```
a_colour := (N + 'a black and white')
```

ASK ALWAYS:

FRS0605B.QTeleV.ConTV

^I QTeleV
^I
^Does_your household have any of the following items?

...^a_colour TV set?^N
 ^I^BLU
Include items stored but in working order, and items under repair.^I

(1)	One	One only
(2)	More	more than one
(3)	None	none

ASK IF: *(Over75 > 0) AND ((ConTV[1] IN [One .. More]) OR (ConTV[2] IN [One .. More]))*

FRS0605B.QTeleV.TVLic

^I QTeleV
^I
^N Do you claim a concessionary television licence?^N

^I^BLU^IC These are free tv licences for those aged 75 or over.

(1)	Yes	Yes
(2)	No	No

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

COMPUTE ALWAYS:

```
PersList[1] := ''
```

COMPUTE ALWAYS:

```
PersList[2] := ''
```

COMPUTE ALWAYS:

```
PersList[3] := ''
```

COMPUTE ALWAYS:

```
Elig[1] := 0
```

COMPUTE ALWAYS:

```
Elig[2] := 0
```

COMPUTE ALWAYS:

```
Elig[3] := 0
```

COMPUTE IF: *In loop FOR Loop1 := 1 TO HHSize*
AND: *((DMAge[Loop1] IN [16 .. 50]) AND (PRec[Loop1].Sex = Female)) OR (DMAge[Loop1] IN [0 .. 15])*

```
PersList[1] := (PersList[1] + STR(Loop1,2) + ' : ' +
DMName[Loop1] + '
')
```

COMPUTE IF: *In loop FOR Loop1 := 1 TO HHSize*
AND: *((DMAge[Loop1] IN [16 .. 50]) AND (PRec[Loop1].Sex = Female)) OR (DMAge[Loop1] IN [0 .. 15])*

Elig[1] := (Elig[1] + 1)

COMPUTE IF: *In loop FOR Loop1 := 1 TO HHSize*
AND: *PRec[Loop1].TypeEd IN [Nursery .. Nonadv]*

**PersList[2] := (PersList[2] + STR(Loop1,2,0) + ' : ' +
DMName[Loop1] + '
')**

COMPUTE IF: *In loop FOR Loop1 := 1 TO HHSize*
AND: *PRec[Loop1].TypeEd IN [Nursery .. Nonadv]*

Elig[2] := (Elig[2] + 1)

COMPUTE IF: *In loop FOR Loop1 := 1 TO HHSize*
AND: *(DMAge[Loop1] IN [2 .. 18]) AND (PRec[Loop1].TypeEd IN [Nursery .. Nonadv])*

**PersList[3] := (PersList[3] + STR(Loop1,2,0) + ' : ' +
DMName[Loop1] + '
')**

COMPUTE IF: *In loop FOR Loop1 := 1 TO HHSize*
AND: *(DMAge[Loop1] IN [2 .. 18]) AND (PRec[Loop1].TypeEd IN [Nursery .. Nonadv])*

Elig[3] := (Elig[3] + 1)

FRS0605B.QWelfare

Questions about free meals etc

COMPUTE IF: PAllCh > 0

incl_child := ' (including any of your children under 16)'

COMPUTE IF: PAllCh = 1

incl_child := ' (including your child under 16)'

COMPUTE IF: ((Elig[1] + Elig[2]) + Elig[3]) > 1

READ_OUT := (I + 'Individual prompt...' + I)

```
COMPUTE IF: ((Elig[1] + Elig[2]) + Elig[3]) > 0
     AND: Elig[1] >= 1
```

WelfMilk := (N + '...any free welfare milk?' + N)

```
COMPUTE IF: ((Elig[1] + Elig[2]) + Elig[3]) > 0
     AND: NOT (Elig[1] >= 1)
```

WelfMilk := ''

```
COMPUTE IF: ((Elig[1] + Elig[2]) + Elig[3]) > 0
     AND: Elig[2] >= 1
```

SchMilk := (N + '...any free school milk?' + N)

```
COMPUTE IF: ((Elig[1] + Elig[2]) + Elig[3]) > 0
     AND: NOT (Elig[2] >= 1)
```

SchMilk := ''

```
COMPUTE IF: ((Elig[1] + Elig[2]) + Elig[3]) > 0
     AND: Elig[3] >= 1
```

SchMeal := (N + '...any free school meals?' + N)

```
COMPUTE IF: ((Elig[1] + Elig[2]) + Elig[3]) > 0
     AND: NOT (Elig[3] >= 1)
```

SchMeal := ''

ASK IF: ((Elig[1] + Elig[2]) + Elig[3]) > 0

FRS0605B.QWelfare.FreeItem

^I QWelfare
^I
^I^BLU^IC Questions about free school meals and welfare milk.^BLU^I

^N In the last 7 days, have ^you^incl_child had...^N

^BLU^READ_OUT

SET [3] OF

(1)	WMilk	^WelfMilk
(2)	SMilk	^SchMilk
(3)	SMeal	^SchMeal
(4)	None	None of these

CHECK IF: ((Elig[1] + Elig[2]) + Elig[3]) > 0
AND: None IN FreeItem
FreeItem.CARDINAL = 1

^I 'None of these' is an exclusive code for this question.^I

CHECK IF: ((Elig[1] + Elig[2]) + Elig[3]) > 0
AND: WMilk IN FreeItem
Elig[1 > 0

^I Code 1 is not valid for this question.^I

CHECK IF: ((Elig[1] + Elig[2]) + Elig[3]) > 0
AND: SMilk IN FreeItem
Elig[2 > 0

^I Code 2 is not valid for this question.^I

CHECK IF: ((Elig[1] + Elig[2]) + Elig[3]) > 0
AND: SMeal IN FreeItem
Elig[3 > 0

^I Code 3 is not valid for this question.^I

FRS0605B.QWelfare.WMkQ[]

```
RECORD IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
```

FRS0605B.QWelfare.WMkQ[].Person

^I QWelfare
^I

^N Person identifier.^N

0..14

```
RECORD IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
```

FRS0605B.QWelfare.WMkQ[].BenUnit

^I QWelfare
^I

^N BU number of recipient.^N

0..7

```
COMPUTE IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: PHHSize = 1
```

WMkPer := 1

```
COMPUTE IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: PHHSize = 1
```

NameOf := 'you'

```
ASK IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: NOT (PHHSize = 1)
```

FRS0605B.QWelfare.WMkQ[].WMkPer

^I QWelfare
^I

^N Who received the free^B welfare milk^B?^N

^I^BLU^IC Type in person number.

^PersList[1]^I

0..14

```
CHECK IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: NOT (PHHSize = 1)
     AND: WMkPer = RESPONSE
     (WMkPer > 0) AND (WMkPer <= PHHSize)
```

^I This code is not valid for this question.^I

```
COMPUTE IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: NOT (PHHSize = 1)
```

NameOf := DMName[[WMkPer]

```
COMPUTE IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
```

Person := WMkPer

```
ASK IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: WMkPer = RESPONSE
```

FRS0605B.QWelfare.WMkQ[].WMkIt

^I QWelfare
^I

^N Thinking just of the past seven days ending yesterday - how many pints did ^NameOf receive?
^N

0..97

```
WARN IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: WMkPer = RESPONSE
     AND: Edit = No
     WMkIt <= 7
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
ASK IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: Elig[1] > 1
```

FRS0605B.QWelfare.WMkQ[].WMIntro

^I QWelfare
^I

^I^BLU^IC Prompt: Has anyone else had any free welfare milk during the past seven days ending yesterday?
^I

(1)	Yes	Yes
(2)	No	No

```
COMPUTE IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
     AND: NOT (Elig[1] > 1)
```

WMIntro := No

FRS0605B.QWelfare (continued)

Questions about free meals etc

```
COMPUTE IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (WMkQ[Index1 - 1].WMIntro = Yes)
```

WMkQ[Index1].BenUnit := DMBU[[WMkQ[Index1].WMkPer]

```
CHECK IF: WMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: In loop FOR Index2 := 1 TO 5
     AND: Index2 < Index1
     WMkQ[Index2].WMkPer <> WMkQ[Index1].WMkPer
```

^I You have already entered this person number.^I

FRS0605B.QWelfare.SMkQ[]

RECORD IF: *SMilk IN FreeItem*
    AND: *In loop FOR Index1 := 1 TO 5*
    AND: *(Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)*

FRS0605B.QWelfare.SMkQ[].Person

^I QWelfare
^I

^N Person identifier.^N

0..14

RECORD IF: *SMilk IN FreeItem*
    AND: *In loop FOR Index1 := 1 TO 5*
    AND: *(Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)*

FRS0605B.QWelfare.SMkQ[].BenUnit

^I QWelfare
^I

^N BU number of recipient.^N

0..7

ASK IF: *SMilk IN FreeItem*
    AND: *In loop FOR Index1 := 1 TO 5*
    AND: *(Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)*

FRS0605B.QWelfare.SMkQ[].SMkPer

^I QWelfare
^I

^I Free School Milk may be available if the child is in full-time education at a state school. The rules for offering Free School Milk differ by Local Education Authority. The age of children for whom it can be claimed also varies by LEA.^I

0..14

COMPUTE IF: *SMilk IN FreeItem*
    AND: *In loop FOR Index1 := 1 TO 5*
    AND: *(Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)*

Person := SMkPer

```
ASK IF: SMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)
```

FRS0605B.QWelfare.SMkQ[].SMkIt

^I QWelfare
^I

^N Thinking just of the past seven days ending yesterday - how many cartons or bottles did ^DMName[SmkPer] receive?
^N

0..97

```
WARN IF: SMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)
     AND: Edit = No
     SMkIt <= 6
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
ASK IF: SMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)
     AND: Elig[2] > 1
```

FRS0605B.QWelfare.SMkQ[].SMIntro

^I QWelfare
^I

^I^BLU^IC Prompt:^I^N Has any other child had any free school milk during the past seven days ending yesterday?

Only applicable to children at state schools.^N

(1)	Yes	Yes
(2)	No	No

```
COMPUTE IF: SMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)
     AND: NOT (Elig[2] > 1)
```

SMIntro := No

FRS0605B.QWelfare (continued)

Questions about free meals etc

```
COMPUTE IF: SMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMkQ[Index1 - 1].SMIntro = Yes)
```

SMkQ[Index1].BenUnit := DMBU[[SMkQ[Index1].SMkPer]

```
CHECK IF: SMilk IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: In loop FOR Index2 := 1 TO 5
     AND: Index2 < Index1
     SMkQ[Index2].SMkPer <> SMkQ[Index1].SMkPer
```

^I You have already entered this person number.^I

FRS0605B.QWelfare.SMlQ[]

```
RECORD IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
```

FRS0605B.QWelfare.SMlQ[].BenUnit

^I QWelfare
^I

^N BU number of recipient.

0..7

```
RECORD IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
```

FRS0605B.QWelfare.SMlQ[].Person

^I QWelfare
^I

^N Person identifier.

0..14

```
COMPUTE IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
     AND: PHHSize = 1
```

MLPer := 1

```
COMPUTE IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
     AND: PHHSize = 1
```

have_you := 'have you'

```
ASK IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
     AND: NOT (PHHSize = 1)
```

FRS0605B.QWelfare.SMlQ[].MLPer

^I QWelfare
^I

^N Who received the^B free school meals^B?

^BLU Only applicable to children at state schools. Can include 16-18 year olds.^N
^I^IC Type in person number.

^PersList[3]^I

0..14

```
COMPUTE IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
     AND: NOT (PHHSize = 1)
```

have_you := ('has ' + DMName[[MLPer])

```
COMPUTE IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
```

Person := MLPer

```
ASK IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
     AND: MLPer = RESPONSE
```

FRS0605B.QWelfare.SMlQ[].SMlIt

^I QWelfare
^I

^N Thinking just of the PAST seven days ending yesterday, how many free school meals ^have_you had?
^N

0..97

```
WARN IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
     AND: MLPer = RESPONSE
     AND: Edit = No
     SMlIt <= 21
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
ASK IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
     AND: Elig[3] > 1
```

FRS0605B.QWelfare.SMlQ[].MLIntro

^I QWelfare
^I

^I^BLU^IC Prompt:^I^N Has any other child had any free school meals during the past seven days ending yesterday?

Only applicable to children at state schools. Can include 16-18 year olds.^N

(1)	Yes	Yes
(2)	No	No

```
COMPUTE IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
     AND: NOT (Elig[3] > 1)
```

MLIntro := No

FRS0605B.QWelfare (continued)

Questions about free meals etc

```
COMPUTE IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: (Index1 = 1) OR (SMlQ[Index1 - 1].MLIntro = Yes)
```

SMlQ[Index1].BenUnit := DMBU[[SMlQ[Index1].MLPer]

```
CHECK IF: SMeal IN FreeItem
     AND: In loop FOR Index1 := 1 TO 5
     AND: In loop FOR Index2 := 1 TO 5
     AND: Index2 < Index1
     SMlQ[Index2].MLPer <> SMlQ[Index1].MLPer
```

^I You have already entered this person number.^I

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

```
COMPUTE IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.WMkQ[Loop1].WMkPer = RESPONSE
```

Loop5 := QWelfare.WMkQ[Loop1].WMkPer

```
CHECK IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.WMkQ[Loop1].WMkPer = RESPONSE
     ((Loop5 > 0) AND (Loop5 <= HHSize)) AND (((IN(DMAge[Loop5],[16..50]))
     AND (PRec[Loop5].Sex = Female)) OR (IN(DMAge[Loop5],[0..15])))
```

^I This code is not valid for this question.^I

```
COMPUTE IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.WMkQ[Loop1].WMkPer = RESPONSE
```

QWelfare.WMkQ[Loop1].BenUnit := DMBU[Loop5]

```
COMPUTE IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.SMkQ[Loop1].SMkPer = RESPONSE
```

Loop5 := QWelfare.SMkQ[Loop1].SMkPer

```
CHECK IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.SMkQ[Loop1].SMkPer = RESPONSE
     ((Loop5 > 0) AND (Loop5 <= HHSize)) AND (IN(PRec[Loop5].TypeEd,[???]))
```

^I This code is not valid for this question.^I

```
COMPUTE IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.SMkQ[Loop1].SMkPer = RESPONSE
```

QWelfare.SMkQ[Loop1].BenUnit := DMBU[Loop5]

```
COMPUTE IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.SMlQ[Loop1].MLPer = RESPONSE
```

Loop5 := QWelfare.SMlQ[Loop1].MLPer

```
CHECK IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.SMlQ[Loop1].MLPer = RESPONSE
     (((Loop5 > 0) AND (Loop5 <= HHSize)) AND (IN(DMAge[Loop5],[2..18])))
     AND (IN(PRec[Loop5].TypeEd,[???]))
```

^I This code is not valid for this question.^I

```
COMPUTE IF: In loop FOR Loop1 := 1 TO 5
     AND: QWelfare.SMlQ[Loop1].MLPer = RESPONSE
```

QWelfare.SMlQ[Loop1].BenUnit := DMBU[Loop5]

WARN IF: *In loop FOR Loop1 := 1 TO 5*
AND: *PRec[QWelfare.SMlQ[Loop1].MLPer].TypeEd IN [Nursery, Primry, MidPri .. Nonadv]*
(IN(QWelfare.SMlQ[Loop1].SMlIt,[0..5])) AND INVOLVING(QWelfare.SMlQ[Loop1].SMlIt)

^I That's ^QWelfare.SMlQ[Loop1].SMlIt meals - for this type of school the weekly maximum would normally be 5 (ie. one meal per day) - please check.^I

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

COMPUTE IF: *SMeal IN QWelfare.FreeItem*

QAccomdat.SchMeal := Yes

COMPUTE IF: *NOT (SMeal IN QWelfare.FreeItem)*

QAccomdat.SchMeal := No

COMPUTE IF: *SMilk IN QWelfare.FreeItem*

QAccomdat.SchMilk := Yes

COMPUTE IF: *NOT (SMilk IN QWelfare.FreeItem)*

QAccomdat.SchMilk := No

COMPUTE IF: *WMilk IN QWelfare.FreeItem*

QAccomdat.WelfMilk := Yes

COMPUTE IF: *NOT (WMilk IN QWelfare.FreeItem)*

QAccomdat.WelfMilk := No

FRS0605B.QChCare

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

Child[Index2].BenUnit := DMBU[[Index2]

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

Child[Index2].Person := Index2

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: DMNumParn[Index2] = 1
```

you := 'you'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: DMNumParn[Index2] = 1
```

PNames := DMName[[DMParent1[[Index2]]

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: DMNumParn[Index2] > 1
```

you := 'either of you'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: DMNumParn[Index2] > 1
```

PNames := (DMName[[DMParent1[[Index2]] + ' and ' + DMName[[DMParent2[[Index2]])

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

ChName := DMName[[Index2]

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

ChAge := DMAge[[Index2]

FRS0605B.QChCare.Child[]

```
RECORD IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

FRS0605B.QChCare.Child[].BenUnit

^I QChCare
^I

^N BU number of person

0..7

```
RECORD IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

FRS0605B.QChCare.Child[].Person

^I QChCare
^I

^N Person identifier.

0..14

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: NatCen = NI
```

HBenCTRT := 'Housing Benefit/rent/rates rebate'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: NOT (NatCen = NI)
```

HBenCTRT := 'Housing Benefit/Council Tax Benefit'

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

FRS0605B.QChCare.Child[].Disp

^I QChCare
^I

^N The next questions are about childcare for your child. This includes all types of childcare such as playschool or nursery school or a childminder^B as well as^B relatives or friends who look after your child. ^N

^I^BLU^IC (The questions should be directed at ^B^PNames^B wherever possible)^I

(1) Cont Press <Enter> to continue.

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ParentNo = 1
```

LoneParent := Yes

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ParentNo = 1
```

NCDVLP := Yes

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: NOT (ParentNo = 1)
```

LoneParent := No

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

AttTxt[[1] := ''

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

AttTxt[[2] := ''

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

AttTxt[[3] := ''

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

AttTxt[[4] := ''

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

AttTxt[[5] := ''

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

AttTxt[[6] := ''

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

AttTxt[[9] := ''

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

AttTxt[[10] := 'other provider'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

AttTxt[[7] := 'Holiday scheme / club'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

AttTxt[[8] := 'Children's centres / integrated centres / combined centres'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ChAge < 6
```

AttTxt[[1] := 'Playgroup or pre school'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ChAge < 6
```

AttTxt[[2] := 'Day nursery or workplace creche'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ChAge < 6
```

AttTxt[[3] := 'Nursery school'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ChAge IN [3 .. 5]
```

AttTxt[[4] := 'Infant's school'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ChAge IN [2 .. 5]
```

AttTxt[[5] := 'Primary school'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ChAge >= 2
```

AttTxt[[6] := 'Breakfast / After school club'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ChAge >= 2
```

`AttTxt[[9] := 'SPONTANEOUS ONLY - Boarding school'`

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

FRS0605B.QChCare.Child[].ChAtt

^I QChCare
^I

^I Count all after school activities as 'After school clubs'.
This includes all study support or recreational activities after school hours.^I

SET [10] OF

(1)	PlayGp	^N^AttTxt[1]
(2)	DayNurse	^N^AttTxt[2]
(3)	Nursery	^N^AttTxt[3]
(4)	InfantS	^N^AttTxt[4]
(5)	PrimaryS	^N^AttTxt[5]
(6)	Breakfst	^N^AttTxt[6]
(7)	Holiday	^N^AttTxt[7]
(8)	ChCentre	^N^AttTxt[8]
(9)	Boarding	^N^AttTxt[9]
(10)	Other	SPONTANEOUS ONLY - ^N Other
(11)	None	None of the above

```
CHECK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: None IN ChAtt
     ChAtt.CARDINAL = 1
```

'None of the above' is an exclusive code.

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 9
     AND: ii IN ChAtt
     AttTxt[[ii] <> ''
```

Code ^ii is not valid for this child

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 9
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 9
     RESERVECHECK
```

RESERVECHECK

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: InfantS IN ChAtt
```

FRS0605B.QChCare.Child[].ChInf

^I QChCare
^I

(Can I just check), ^N at the infant's school, was ^ChName in a^N

(1)	Recept	^N Reception class?
(2)	Nursery	^N Nursery class?
(3)	None	^N None of the above

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: PrimaryS IN ChAtt
```

FRS0605B.QChCare.Child[].ChPri

^I QChCare
^I

(Can I just check), ^N at the Primary school, was ^ChName in a^N

(1)	Recept	^N Reception class?
(2)	Nursery	^N Nursery class?
(3)	None	^N None of the above

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ((((((((PlayGp IN ChAtt) OR (DayNurse IN ChAtt)) OR (Nursery IN
     ChAtt)) OR (Breakfst IN ChAtt)) OR (Holiday IN ChAtt)) OR (ChCentre IN
     ChAtt)) OR (Boarding IN ChAtt)) OR (Other IN ChAtt)) OR (ChInf IN
     [Recept, Nursery])) OR (ChPri IN [Recept, Nursery])
     AND: ChAge >= 3
```

FRS0605B.QChCare.Child[].CTrm

^I QChCare
^I

(Can I just check)^N for ^ChName was that week in term time or was it a school holiday...^N

(1)	Termtime	^N Term time^N
(2)	Halfterm	^N Half term^N
(3)	Holiday	^N Or other school holiday?^N
(4)	NotApp	^N Not applicable^N

```
ASK IF: AllCh > 0
    AND: In loop FOR Index2 := 1 TO HHSize
    AND: DMAge[Index2] IN [0 .. 15]
```

FRS0605B.QChCare.Child[].ChPeo

^I QChCare
^I

^IS^I R^I^N
And during those seven days (ending Sunday the ^DatLSun) were there any other people who looked after ^ChName?^N

^I^BLU^IC Other than resident parent/guardian, and staff contact whilst at places previously mentioned. Code babysitters as 8 'other non-relative'

SET [9] OF

(1)	Granps	Child's grand parents
(2)	NonRes	Child's non-resident parent/an ex-spouse/an ex-partner
(3)	BroSis	Child's brother or sister
(4)	Rels	Other relatives
(5)	ChMind	Childminder
(6)	Nanny	Nanny/Au pair (includes both live-in and day nannies)
(7)	Friends	Friends or neighbours
(8)	NonRels	Other non-relatives (includes babysitters)
(9)	NotMind	SPONTANEOUS ONLY - Does not require minding
(10)	None	None of the above

```
CHECK IF: AllCh > 0
    AND: In loop FOR Index2 := 1 TO HHSize
    AND: DMAge[Index2] IN [0 .. 15]
    AND: None IN ChPeo
    ChPeo.CARDINAL = 1
```

'None' exclusive for this question.

```
COMPUTE IF: AllCh > 0
    AND: In loop FOR Index2 := 1 TO HHSize
    AND: DMAge[Index2] IN [0 .. 15]
    AND: ChMind IN ChPeo
```

provider := 'Childminder'

ASK IF: *AllCh > 0*
   AND: *In loop FOR Index2 := 1 TO HHSize*
   AND: *DMAge[Index2] IN [0 .. 15]*
   AND: *ChMind IN ChPeo*

FRS0605B.QChCare.Child[].Registrd

^I QChCare
^I

^I Registered or approved childcare can include:
Registered childminders, nurseries and play schemes.
Out of hours clubs on school premises run by a school of local authority.
Childcare schemes run by approved providers , for example, an out of school hours scheme or a provider approved under a Ministry of Defence accreditation scheme.

A registered childminder, nursery or childcare scheme is one that is registered by:

OFSTED or the Social Care Inspection Commission in England
The National Assembly for Wales (through the Care Standards Inspectorate for Wales) in Wales
The Scottish Commission for the Regulation for Care in Scotland
A Health and Social Services Trust in Northern Ireland.

Approved childcare in England is childcare approved by a body acting under the authority of the Secretary of State.

Childcare providers who are eligible to apply for approval in England include:
Childminders who are not required to register.
Nannies or au pairs^I

(1)	Register	Registered OR Approved
(2)	NonReg	Not registered OR Not approved

COMPUTE IF: *AllCh > 0*
   AND: *In loop FOR Index2 := 1 TO HHSize*
   AND: *DMAge[Index2] IN [0 .. 15]*
   AND: *ChMind IN ChPeo*
   AND: *Registrd[1] = Registered*

ChText := provider

COMPUTE IF: *AllCh > 0*
   AND: *In loop FOR Index2 := 1 TO HHSize*
   AND: *DMAge[Index2] IN [0 .. 15]*
   AND: *ChMind IN ChPeo*

with1 := 'with'

ASK IF: *AllCh > 0*
   AND: *In loop FOR Index2 := 1 TO HHSize*
   AND: *DMAge[Index2] IN [0 .. 15]*
   AND: *ChMind IN ChPeo*

FRS0605B.QChCare.Child[].EmplProv

^I QChCare
^I

^N Is the childcare ^with1 ^provider provided by your employer?^N

(1)	Yes	Yes
(2)	No	No
(3)	DNA	Does not apply - child's parents/guardians have no employer

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: PlayGp IN ChAtt
```

provider := 'playgroup or pre-school'

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: PlayGp IN ChAtt
```

FRS0605B.QChCare.Child[].Registrd

^I QChCare
^I

^I Registered or approved childcare can include:
Registered childminders, nurseries and play schemes.
Out of hours clubs on school premises run by a school of local authority.
Childcare schemes run by approved providers , for example, an out of school hours scheme or a provider approved under a Ministry of Defence accreditation scheme.

A registered childminder, nursery or childcare scheme is one that is registered by:

OFSTED or the Social Care Inspection Commission in England
The National Assembly for Wales (through the Care Standards Inspectorate for Wales) in Wales
The Scottish Commission for the Regulation for Care in Scotland
A Health and Social Services Trust in Northern Ireland.

Approved childcare in England is childcare approved by a body acting under the authority of the Secretary of State.

Childcare providers who are eligible to apply for approval in England include:
Childminders who are not required to register.
Nannies or au pairs^I

(1)	Register	Registered OR Approved
(2)	NonReg	Not registered OR Not approved

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: PlayGp IN ChAtt
     AND: Registrd[2] = Registered
     AND: ChText =
```

ChText := provider

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: PlayGp IN ChAtt
     AND: Registrd[2] = Registered
     AND: NOT (ChText =)
```

ChText := (ChText + ' and ' + provider)

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: DayNurse IN ChAtt
```

provider := 'day nursery or creche'

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: DayNurse IN ChAtt
```

FRS0605B.QChCare.Child[].Registrd

^I QChCare
^I

^I Registered or approved childcare can include:
Registered childminders, nurseries and play schemes.
Out of hours clubs on school premises run by a school of local authority.
Childcare schemes run by approved providers , for example, an out of school hours scheme or a provider approved under a Ministry of Defence accreditation scheme.

A registered childminder, nursery or childcare scheme is one that is registered by:

OFSTED or the Social Care Inspection Commission in England
The National Assembly for Wales (through the Care Standards Inspectorate for Wales) in Wales
The Scottish Commission for the Regulation for Care in Scotland
A Health and Social Services Trust in Northern Ireland.

Approved childcare in England is childcare approved by a body acting under the authority of the Secretary of State.

Childcare providers who are eligible to apply for approval in England include:
Childminders who are not required to register.
Nannies or au pairs^I

(1)	Register	Registered OR Approved
(2)	NonReg	Not registered OR Not approved

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: DayNurse IN ChAtt
     AND: Registrd[3] = Registered
     AND: ChText =
```

ChText := provider

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: DayNurse IN ChAtt
     AND: Registrd[3] = Registered
     AND: NOT (ChText =)
```

ChText := (ChText + ' and ' + provider)

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: DayNurse IN ChAtt
```

with1 := 'in'

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: DayNurse IN ChAtt
```

FRS0605B.QChCare.Child[].EmplProv

^I QChCare
^I

^N Is the childcare ^with1 ^provider provided by your employer?^N

(1)	Yes	Yes
(2)	No	No
(3)	DNA	Does not apply - child's parents/guardians have no employer

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Nursery IN ChAtt
```

provider := (ChName + ''s nursery school')

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Nursery IN ChAtt
```

FRS0605B.QChCare.Child[].Registrd

^I QChCare
^I

^I Registered or approved childcare can include:
Registered childminders, nurseries and play schemes.
Out of hours clubs on school premises run by a school of local authority.
Childcare schemes run by approved providers , for example, an out of school hours scheme or a provider approved under a Ministry of Defence accreditation scheme.

A registered childminder, nursery or childcare scheme is one that is registered by:

OFSTED or the Social Care Inspection Commission in England
The National Assembly for Wales (through the Care Standards Inspectorate for Wales) in Wales
The Scottish Commission for the Regulation for Care in Scotland
A Health and Social Services Trust in Northern Ireland.

Approved childcare in England is childcare approved by a body acting under the authority of the Secretary of State.

Childcare providers who are eligible to apply for approval in England include:
Childminders who are not required to register.
Nannies or au pairs^I

(1)	Register	Registered OR Approved
(2)	NonReg	Not registered OR Not approved

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Nursery IN ChAtt
     AND: Registrd[4] = Registered
     AND: ChText =
```

ChText := provider

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Nursery IN ChAtt
     AND: Registrd[4] = Registered
     AND: NOT (ChText =)
```

ChText := (ChText + ' and ' + provider)

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ChInf IN [Recept, Nursery]
```

provider := (ChName + ''s infant's school')

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ChInf IN [Recept, Nursery]
```

FRS0605B.QChCare.Child[].Registrd

^I QChCare
^I

^I Registered or approved childcare can include:
Registered childminders, nurseries and play schemes.
Out of hours clubs on school premises run by a school of local authority.
Childcare schemes run by approved providers , for example, an out of school hours scheme or a provider approved under a Ministry of Defence accreditation scheme.

A registered childminder, nursery or childcare scheme is one that is registered by:

OFSTED or the Social Care Inspection Commission in England
The National Assembly for Wales (through the Care Standards Inspectorate for Wales) in Wales
The Scottish Commission for the Regulation for Care in Scotland
A Health and Social Services Trust in Northern Ireland.

Approved childcare in England is childcare approved by a body acting under the authority of the Secretary of State.

Childcare providers who are eligible to apply for approval in England include:
Childminders who are not required to register.
Nannies or au pairs^I

(1)	Register	Registered OR Approved
(2)	NonReg	Not registered OR Not approved

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ChInf IN [Recept, Nursery]
     AND: Registrd[5] = Registered
     AND: ChText =
```

ChText := provider

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ChInf IN [Recept, Nursery]
     AND: Registrd[5] = Registered
     AND: NOT (ChText =)
```

ChText := (ChText + ' and ' + provider)

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ChPri IN [Recept, Nursery]
```

provider := 'primary school'

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ChPri IN [Recept, Nursery]
```

FRS0605B.QChCare.Child[].Registrd

^I QChCare
^I

^I Registered or approved childcare can include:
Registered childminders, nurseries and play schemes.
Out of hours clubs on school premises run by a school of local authority.
Childcare schemes run by approved providers , for example, an out of school hours scheme or a provider approved under a Ministry of Defence accreditation scheme.

A registered childminder, nursery or childcare scheme is one that is registered by:

OFSTED or the Social Care Inspection Commission in England
The National Assembly for Wales (through the Care Standards Inspectorate for Wales) in Wales
The Scottish Commission for the Regulation for Care in Scotland
A Health and Social Services Trust in Northern Ireland.

Approved childcare in England is childcare approved by a body acting under the authority of the Secretary of State.

Childcare providers who are eligible to apply for approval in England include:
Childminders who are not required to register.
Nannies or au pairs^I

(1)	Register	Registered OR Approved
(2)	NonReg	Not registered OR Not approved

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ChPri IN [Recept, Nursery]
     AND: Registrd[6] = Registered
     AND: ChText =
```

ChText := provider

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: ChPri IN [Recept, Nursery]
     AND: Registrd[6] = Registered
     AND: NOT (ChText =)
```

ChText := (ChText + ' and ' + provider)

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Nanny IN ChPeo
```

provider := 'nanny/au-pair'

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Nanny IN ChPeo
```

FRS0605B.QChCare.Child[].Registrd

^I QChCare
^I

^I Registered or approved childcare can include:
Registered childminders, nurseries and play schemes.
Out of hours clubs on school premises run by a school of local authority.
Childcare schemes run by approved providers , for example, an out of school hours scheme or a provider approved under a Ministry of Defence accreditation scheme.

A registered childminder, nursery or childcare scheme is one that is registered by:

OFSTED or the Social Care Inspection Commission in England
The National Assembly for Wales (through the Care Standards Inspectorate for Wales) in Wales
The Scottish Commission for the Regulation for Care in Scotland
A Health and Social Services Trust in Northern Ireland.

Approved childcare in England is childcare approved by a body acting under the authority of the Secretary of State.

Childcare providers who are eligible to apply for approval in England include:
Childminders who are not required to register.
Nannies or au pairs^I

(1)	Register	Registered OR Approved
(2)	NonReg	Not registered OR Not approved

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Nanny IN ChPeo
     AND: Registrd[7] = Registered
     AND: ChText =
```

ChText := provider

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Nanny IN ChPeo
     AND: Registrd[7] = Registered
     AND: NOT (ChText =)
```

ChText := (ChText + ' and ' + provider)

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Nanny IN ChPeo
```

with1 := 'with'

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Nanny IN ChPeo
```

FRS0605B.QChCare.Child[].EmplProv

^I QChCare
^I

^N Is the childcare ^with1 ^provider provided by your employer?^N

(1)	Yes	Yes
(2)	No	No
(3)	DNA	Does not apply - child's parents/guardians have no employer

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[1] := 'Playgroup or pre school'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[2] := 'Day nursery or workplace creche'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[3] := 'Nursery school'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[4] := ''

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[5] := ''

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[6] := 'Breakfast / After school club'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[7] := 'Holiday scheme / club'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[8] := 'Children's centres / integrated centres / combined centres'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[9] := 'Boarding school'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[10] := 'other provider'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[11] := 'grandparents'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[12] := 'non-resident parent/an ex-spouse/an ex-partner'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[13] := 'brother or sister'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[14] := 'other relatives'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[15] := 'childminder'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[16] := 'nanny/au pair (includes both live-in and day nannies)'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[17] := 'friends or neighbours'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
```

typecare[[18] := 'other non-relatives'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
```

with1 := 'in'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: ii = 4
     AND: ChInf = Recept
```

typecare[[4] := 'Infant's school (reception class)'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: ii = 4
     AND: ChInf = Nursery
```

typecare[[4] := 'Infant's school (nursery class)'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: ii = 4
     AND: NOT (ChInf = Nursery)
```

typecare[[4] := ''

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: ii = 5
     AND: ChPri = Recept
```

typecare[[5] := 'Primary school (reception class)'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: ii = 5
     AND: ChPri = Nursery
```

typecare[[5] := 'Primary school (nursery class)'

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: ii = 5
     AND: NOT (ChPri = Nursery)
```

typecare[[5] := ''

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: typecare[ii] <>
```

FRS0605B.QChCare.Child[].ChHr1

^I QChCare
^I

@>^I Help <F9>^I
@< ^N
About how many hours in the seven days ending Sunday the ^DatLSun did ^ChName spend ^with1 his/her ^typecare[ii]?^N

0..60

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: typecare[ii] <>
     AND: ChHr1[ii] > 0
     AND: ii = 1
```

provider := (ChName + ''s playgroup or pre school')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: typecare[ii] <>
     AND: ChHr1[ii] > 0
     AND: ii = 2
```

provider := (ChName + ''s day nursery or workplace creche')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: typecare[ii] <>
     AND: ChHr1[ii] > 0
     AND: ii = 3
```

provider := (ChName + ''s nursery school')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: typecare[ii] <>
     AND: ChHr1[ii] > 0
     AND: ii = 4
```

provider := (ChName + ''s infant school')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: typecare[ii] <>
     AND: ChHr1[ii] > 0
     AND: ii = 5
```

provider := (ChName + ''s primary school')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: typecare[ii] <>
     AND: ChHr1[ii] > 0
     AND: ii = 6
```

provider := (ChName + ''s Breakfast / After school club')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: typecare[ii] <>
     AND: ChHr1[ii] > 0
     AND: ii = 7
```

provider := (ChName + ''s Holiday scheme / club')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: typecare[ii] <>
     AND: ChHr1[ii] > 0
     AND: ii = 8
```

provider := (ChName + ''s children's centres / integrated centres / combined centres')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: typecare[ii] <>
     AND: ChHr1[ii] > 0
     AND: ii = 9
```

provider := (ChName + ''s boarding school')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: typecare[ii] <>
     AND: ChHr1[ii] > 0
     AND: ii = 10
```

provider := (ChName + ''s other provider')

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [1 .. 10]
     AND: ii IN ChAtt
     AND: typecare[ii] <>
     AND: ChHr1[ii] > 0
     AND: Cost[ii] = Yes
```

FRS0605B.QChCare.Child[].ChAmt1

^I QChCare
^I

^N How much was your most recent payment for the childcare ^with1 ^provider?^N

^I^BLU^IC If unable to attribute costs per child, then enter an estimate by dividing total childcare costs by number of children
^I^IC Only include payments made by the parents/guardian. Exclude payments made by others e.g grandparents or where payment is made with childcare vouchers.^I

0.00..99997.00

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [11 .. 18]
```

with1 := 'with'

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [11 .. 18]
     AND: (ii - 10) IN ChPeo
```

FRS0605B.QChCare.Child[].ChHr1

^I QChCare
^I

@>^I Help <F9>^I
@< ^N
About how many hours in the seven days ending Sunday the ^DatLSun did ^ChName spend ^with1 his/her ^typecare[ii]?^N

0..60

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [11 .. 18]
     AND: (ii - 10) IN ChPeo
     AND: ChHr1[ii] > 0
     AND: ii = 11
```

provider := (ChName + ''s grandparents')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [11 .. 18]
     AND: (ii - 10) IN ChPeo
     AND: ChHr1[ii] > 0
     AND: ii = 12
```

provider := (ChName + ''s non-resident parent/ex-spouse/ex-partner')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [11 .. 18]
     AND: (ii - 10) IN ChPeo
     AND: ChHr1[ii] > 0
     AND: ii = 13
```

provider := (ChName + ''s brother or sister')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [11 .. 18]
     AND: (ii - 10) IN ChPeo
     AND: ChHr1[ii] > 0
     AND: ii = 14
```

provider := (ChName + ''s other relatives')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [11 .. 18]
     AND: (ii - 10) IN ChPeo
     AND: ChHr1[ii] > 0
     AND: ii = 15
```

provider := (ChName + ''s childminder')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [11 .. 18]
     AND: (ii - 10) IN ChPeo
     AND: ChHr1[ii] > 0
     AND: ii = 16
```

provider := (ChName + ''s nanny/au pair')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [11 .. 18]
     AND: (ii - 10) IN ChPeo
     AND: ChHr1[ii] > 0
     AND: ii = 17
```

provider := (ChName + ''s friends or neighbours')

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [11 .. 18]
     AND: (ii - 10) IN ChPeo
     AND: ChHr1[ii] > 0
     AND: ii = 18
```

provider := (ChName + ''s other non-relatives')

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [11 .. 18]
     AND: (ii - 10) IN ChPeo
     AND: ChHr1[ii] > 0
     AND: (ChPeo = RESPONSE) AND NOT (None IN ChPeo)
```

FRS0605B.QChCare.Child[].PMChk

^I QChCare
^I

I^BLU^IC Ask or record:
Was the time spent with ^provider during the daytime; or at night; or both in the day^B and^B at night?

(1) Yes	Daytime	Daytime only
(1) Yes	Yes	Daytime only
(2) No	Night	At night only
(2) No	No	At night only
(3)	Both	Both day and night (include overnight only stays)

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [11 .. 18]
     AND: (ii - 10) IN ChPeo
     AND: ChHr1[ii] > 0
     AND: Cost[ii] = Yes
```

FRS0605B.QChCare.Child[].ChAmt1

^I QChCare
^I

^N How much was your most recent payment for the childcare ^with1 ^provider?^N

^I^BLU^IC If unable to attribute costs per child, then enter an estimate by dividing total childcare costs by number of children
^I^IC Only include payments made by the parents/guardian. Exclude payments made by others e.g grandparents or where payment is made with childcare vouchers.^I

0.00..99997.00

FRS0605B.QChCare.Child[].QChInKnd[]

```
ASK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [11 .. 18]
     AND: (ii - 10) IN ChPeo
```

FRS0605B.QChCare.Child[].QChInKnd[].ChInKnd

^I QChCare
^I

^IS^I S^I

^X [And/And apart from any money which you paid]^N do you do any of the things on this card to repay ^provider for looking after him/her?

^I^BLU^IC This is an opinion question. Code anything which is not found in categories 1, 2 or 3 as Other and make a note as necessary.

SET [4] OF

(1)	Lookaft	Looked after his/her child(ren) in return
(2)	Favour	Did him/her a favour
(3)	Gift	Gave him/her a gift or treat
(4)	Other	Other
(5)	Nothing	No, nothing

```
CHECK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [11 .. 18]
     AND: (ii - 10) IN ChPeo
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ii IN [11 .. 18]
     AND: (ii - 10) IN ChPeo
     RESERVECHECK
```

RESERVECHECK

FRS0605B.QChCare.Child[] (continued)

```
CHECK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: Edit = No
     AND: ChHr1[ii] = RESPONSE
     ChHr1[ii] < 55
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: Edit = No
     AND: ChAmt1[ii] = RESPONSE
     ChAmt1[ii] < 130
```

^I Warning: The answer is much higher than the figures usually given at this question. Please check that your figure is correct. If so, suppress warning and continue.^I

```
COMPUTE IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: In loop FOR ii := 1 TO 18
     AND: ChAmt1[ii] = NONRESPONSE

HMissVar := (HMissVar + 1)
```

ASK IF: *AllCh > 0*
AND: *In loop FOR Index2 := 1 TO HHSize*
AND: *DMAge[Index2] IN [0 .. 15]*
AND: *(((((((ChMind IN ChPeo) OR (PlayGp IN ChAtt)) OR (DayNurse IN ChAtt)) OR (Nursery IN ChAtt)) OR (ChInf IN [Recept, Nursery])) OR (ChPri IN [Recept, Nursery])) OR (Nanny IN ChPeo)) AND (ChAge < 13)) AND (((((QRenting.HBenAmt > 0) OR (QRenting.HBenefit = Yes)) OR (QCounTax.CTRebAmt > 0)) OR (QCounTax.CTReb = Yes)) OR (QNIRates.RTRebAmt > 0)) OR (QNIRates.RTReb = Yes))*

FRS0605B.QChCare.Child[].BenCCDis

^I QChCare
^I

^N You said earlier that you get (^HBenCTRT). Does the benefit take account of the cost of the ^ChText?^N

(1)	Yes	Yes
(2)	No	No

WARN IF: *AllCh > 0*
AND: *In loop FOR Index2 := 1 TO HHSize*
AND: *DMAge[Index2] IN [0 .. 15]*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
AND: *In loop FOR Index2 := 1 TO HHSize*
AND: *DMAge[Index2] IN [0 .. 15]*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
AND: *In loop FOR Index2 := 1 TO HHSize*
AND: *DMAge[Index2] IN [0 .. 15]*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
AND: *In loop FOR Index2 := 1 TO HHSize*
AND: *DMAge[Index2] IN [0 .. 15]*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
AND: *In loop FOR Index2 := 1 TO HHSize*
AND: *DMAge[Index2] IN [0 .. 15]*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
AND: *In loop FOR Index2 := 1 TO HHSize*
AND: *DMAge[Index2] IN [0 .. 15]*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
AND: *In loop FOR Index2 := 1 TO HHSize*
AND: *DMAge[Index2] IN [0 .. 15]*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
AND: *In loop FOR Index2 := 1 TO HHSize*
AND: *DMAge[Index2] IN [0 .. 15]*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
AND: *In loop FOR Index2 := 1 TO HHSize*
AND: *DMAge[Index2] IN [0 .. 15]*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
AND: *In loop FOR Index2 := 1 TO HHSize*
AND: *DMAge[Index2] IN [0 .. 15]*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
AND: *In loop FOR Index2 := 1 TO HHSize*
AND: *DMAge[Index2] IN [0 .. 15]*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
AND: *In loop FOR Index2 := 1 TO HHSize*
AND: *DMAge[Index2] IN [0 .. 15]*
RESERVECHECK

RESERVECHECK

FRS0605B.QChCare (continued)

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     AND: DMAge[Index2] IN [0 .. 15]
     AND: Edit = No
     AND: ((DMBU[Index2] = 1) AND (Child[Index2].BenCCDis = Yes)) AND
     ((ChMind IN Child[Index2].ChPeo) OR (Nursery IN Child[Index2].ChAtt))
     (((Child[Index2].Registrd[1] = Registered) OR
     (Child[Index2].Registrd[2] = Registered)) OR
     (Child[Index2].Registrd[3] = Registered)) OR
     (Child[Index2].Registrd[4] = Registered)
```

^I Please check this answer with respondent: earlier they said they got the 'Child Care Disregard' for the benefit(s) shown below, and this is usually only possible if the childminder/nursery is registered.^I

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: AllCh > 0
     AND: In loop FOR Index2 := 1 TO HHSize
     RESERVECHECK
```

RESERVECHECK

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

WARN IF: *AllCh > 0*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
RESERVECHECK

RESERVECHECK

WARN IF: *AllCh > 0*
RESERVECHECK

RESERVECHECK

```
COMPUTE IF: In loop FOR Loop1 := 1 TO 14
     AND: ((((QChCare.Child[Loop1].Registrd[1] = Registered) OR
     (QChCare.Child[Loop1].Registrd[2] = Registered)) OR
     (QChCare.Child[Loop1].Registrd[3] = Registered)) OR
     (QChCare.Child[Loop1].Registrd[4] = Registered)) OR
     (QChCare.Child[Loop1].Registrd[5] = Registered)
```

ChRegis := (ChRegis + 1)

FRS0605B.QCare

Questions about carers/cared for

ASK ALWAYS:

FRS0605B.QCare.NeedHelp

^I QCare
^I
^B Keeping an eye out, 'being there':^B
Being available if needed Making your whereabouts known so you can be contacted if needed

^B Social support and assistance:^B
Sitting with Chatting with/ listening to/reading to Making/receiving telephone calls to talk to them Encouraging them to do things for themselves

^B Accompanying on trips out to go:^B Shopping To hospital/ GP/optician/dentist/chiropodist To the park/place of worship/restaurant

^B Paperwork/official/financial:^B
Helping with paperwork Dealing with 'officials' (including by phone)

^B Home and garden:^B
Making meals Going shopping for someone Washing/ironing/changing sheets Cleaning /housework Gardening Odd jobs/maintenance

^B Medical:^B
Collecting prescriptions/giving medication Changing dressings

^B Moving about the home: Giving help with ^B
Getting up and down stairs Moving from room to room Getting in and out of bed

^B Personal care: help with^B
Getting dressed Feeding Washing/bathing/using the toilet

(1)	Yes	Yes
(2)	No	No

Ask always:

FRS0605B.QCare.GiveHelp

^I QCare
^I
^B Keeping an eye out, 'being there':^B
Being available if needed Making your whereabouts known so you can be contacted if needed

^B Social support and assistance:^B
Sitting with Chatting with/ listening to/reading to Making/receiving telephone calls to talk to them Encouraging them to do things for themselves

^B Accompanying on trips out to go:^B Shopping To hospital/ GP/optician/dentist/chiropodist To the park/place of worship/restaurant

^B Paperwork/official/financial:^B
Helping with paperwork Dealing with 'officials' (including by phone)

^B Home and garden:^B
Making meals Going shopping for someone Washing/ironing/changing sheets Cleaning /housework Gardening Odd jobs/maintenance

^B Medical:^B
Collecting prescriptions/giving medication Changing dressings

^B Moving about the home: Giving help with ^B
Getting up and down stairs Moving from room to room Getting in and out of bed

^B Personal care: help with^B
Getting dressed Feeding Washing/bathing/using the toilet

(1)	Yes	Yes
(2)	No	No

FRS0605B.QCare.QRecHelp

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: NeedHelp = Yes
AND: In loop FOR Idy := 1 TO HHSize

LName[Idy] := DMName[[Idy]

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: GiveHelp = Yes

LName[15] := 'Parent outside household'

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: GiveHelp = Yes

LName[16] := 'Other Parent outside household'

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: GiveHelp = Yes

LName[17] := 'Child outside household'

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: GiveHelp = Yes

LName[18] := 'Spouse outside household'

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: GiveHelp = Yes

LName[19] := 'Other relative'

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: GiveHelp = Yes

LName[20] := 'Friend/neighbour'

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: GiveHelp = Yes

LName[21] := 'Client of voluntary organisation'

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: GiveHelp = Yes

LName[22] := 'Other non-household'

ASK IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*

FRS0605B.QCare.QRecHelp.QNeedPer

^I QCare
^I

^I If they provide help or give help for^B more than one^B individual in any one category of:

15: Parent outside household
16: Other parent outside household
18: Child outside household
19: Relative
20: Friend/neighbour
21: Client of voluntary organisation
22: Other non-household

Please make a Note here <Ctrl+M> to tell us how many^B more^B people are involved.^I

SET [5] OF

(1)	Per1	^LName[1]
(2)	Per2	^LName[2]
(3)	Per3	^LName[3]
(4)	Per4	^LName[4]
(5)	Per5	^LName[5]
(6)	Per6	^LName[6]
(7)	Per7	^LName[7]
(8)	Per8	^LName[8]
(9)	Per9	^LName[9]
(10)	Per10	^LName[10]
(11)	Per11	^LName[11]
(12)	Per12	^LName[12]
(13)	Per13	^LName[13]
(14)	Per14	^LName[14]
(15)	Per15	^LName[15]
(16)	Per16	^LName[16]
(17)	Per17	^LName[17]
(18)	Per18	^LName[18]
(19)	Per19	^LName[19]
(20)	Per20	^LName[20]
(21)	Per21	^LName[21]
(22)	Per22	^LName[22]

CHECK IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
AND: *In loop FOR ixy := 1 TO 22*
AND: *ixy IN QNeedPer*
(LName[ixy] <> '') AND INVOLVING(QNeedPer)

^N Record a valid code for person cared for.

ASK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: Per22 IN QNeedPer

FRS0605B.QCare.QRecHelp.NeedPerO

^I QCare
^I

^N Who is the other person outside the household receiving help or being looked after?^N

STRING[40]

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: In loop FOR Idx := 1 TO 5
AND: Idx <= QNeedPer.CARDINAL
AND: QNeedPer[Idx] IN [Per1 .. Per22]

NeedNum := ORD(QNeedPer[Idx])

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: In loop FOR Idx := 1 TO 5
AND: Idx <= QNeedPer.CARDINAL
AND: NeedNum IN [1 .. 14]

NeedName := DMName[[NeedNum]

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: In loop FOR Idx := 1 TO 5
AND: Idx <= QNeedPer.CARDINAL
AND: QNeedPer[Idx] = Per15

NeedName := 'the PARENT'

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: In loop FOR Idx := 1 TO 5
AND: Idx <= QNeedPer.CARDINAL
AND: QNeedPer[Idx] = Per16

NeedName := 'the OTHER PARENT'

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: In loop FOR Idx := 1 TO 5
AND: Idx <= QNeedPer.CARDINAL
AND: QNeedPer[Idx] = Per17

NeedName := 'the CHILD'

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: In loop FOR Idx := 1 TO 5
AND: Idx <= QNeedPer.CARDINAL
AND: QNeedPer[Idx] = Per18

NeedName := 'the SPOUSE'

COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
AND: In loop FOR Idx := 1 TO 5
AND: Idx <= QNeedPer.CARDINAL
AND: QNeedPer[Idx] = Per19

NeedName := 'the RELATIVE'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: QNeedPer[Idx] = Per20
```

NeedName := 'the FRIEND/NEIGHBOUR'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: QNeedPer[Idx] = Per21
```

NeedName := 'the CLIENT of a voluntary organisation'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: QNeedPer[Idx] = Per22
```

NeedName := UPCASE(NeedPerO)

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
```

Recip[Idx].NeedPer := NeedNum

FRS0605B.QCare.QRecHelp.Recip[]

***Record if:** (NeedHelp = Yes) OR (GiveHelp = Yes)*
***And:** In loop FOR Idx := 1 TO 5*
***And:** Idx <= QNeedPer.CARDINAL*

FRS0605B.QCare.QRecHelp.Recip[].NeedPer

^I QCare
^I

^N Who is receiving help/being looked after.^N

0..22

***Compute if:** (NeedHelp = Yes) OR (GiveHelp = Yes)*
***And:** In loop FOR Idx := 1 TO 5*
***And:** Idx <= QNeedPer.CARDINAL*

LNeedPer := NeedPer

***Compute if:** (NeedHelp = Yes) OR (GiveHelp = Yes)*
***And:** In loop FOR Idx := 1 TO 5*
***And:** Idx <= QNeedPer.CARDINAL*

LNeedName := PNeedName

***Ask if:** (NeedHelp = Yes) OR (GiveHelp = Yes)*
***And:** In loop FOR Idx := 1 TO 5*
***And:** Idx <= QNeedPer.CARDINAL*

FRS0605B.QCare.QRecHelp.Recip[].Freq

^I QCare
^I

^N How frequently does ^LNeedName receive such help?^N

(1)	Continuo	Continuously
(2)	SevDay	Several times a day
(3)	OTDay	Once or twice a day
(4)	SevWeek	Several times a week
(5)	OWeek	Once a week
(6)	LessFreq	Less frequently

```
ASK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: Freq IN [Continuously, SevDay]
```

FRS0605B.QCare.QRecHelp.Recip[].DayNight

^I QCare
^I

^N And does ^LNeedName receive help during the daytime; or at night; or both in the day^B and^B at night?^N

(1)	Day	Daytime only
(2)	Night	At night only
(3)	Both	Both day and night

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: LNeedPer > 14
```

Rel := ''

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: LNeedPer > 14
```

Fri := ''

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: LNeedPer > 14
```

LAH := ''

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: LNeedPer > 14
```

Dom := ''

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: LNeedPer > 14
```

Nur := ''

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: LNeedPer > 14
```

Hel := ''

Compute if: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
And: *In loop FOR Idx := 1 TO 5*
And: *Idx <= QNeedPer.CARDINAL*
And: *NOT (LNeedPer > 14)*

Rel := 'Relatives'

Compute if: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
And: *In loop FOR Idx := 1 TO 5*
And: *Idx <= QNeedPer.CARDINAL*
And: *NOT (LNeedPer > 14)*

Fri := 'Friends/Neighbours'

Compute if: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
And: *In loop FOR Idx := 1 TO 5*
And: *Idx <= QNeedPer.CARDINAL*
And: *NOT (LNeedPer > 14)*
And: *NatCen = NI*

LAH := 'Social Services home help or home care worker'

Compute if: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
And: *In loop FOR Idx := 1 TO 5*
And: *Idx <= QNeedPer.CARDINAL*
And: *NOT (LNeedPer > 14)*
And: *NOT (NatCen = NI)*

LAH := 'Local Authority home help or home care worker'

Compute if: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
And: *In loop FOR Idx := 1 TO 5*
And: *Idx <= QNeedPer.CARDINAL*
And: *NOT (LNeedPer > 14)*

Dom := 'Private domestic help'

Compute if: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
And: *In loop FOR Idx := 1 TO 5*
And: *Idx <= QNeedPer.CARDINAL*
And: *NOT (LNeedPer > 14)*

Nur := 'District nurse, health visitor or other kind of nurse'

Compute if: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
And: *In loop FOR Idx := 1 TO 5*
And: *Idx <= QNeedPer.CARDINAL*
And: *NOT (LNeedPer > 14)*

Hel := 'Other outside helpers'

```
ASK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
```

FRS0605B.QCare.QRecHelp.Recip[].WhoLook

^I QCare
^I

^N Who looks after, or provides help for ^LNeedName? ^Anyone_else?^N

^I^BLU^IC Code all that apply.^I

SET [5] OF

(1)	Per1	^DMName[1]
(2)	Per2	^DMName[2]
(3)	Per3	^DMName[3]
(4)	Per4	^DMName[4]
(5)	Per5	^DMName[5]
(6)	Per6	^DMName[6]
(7)	Per7	^DMName[7]
(8)	Per8	^DMName[8]
(9)	Per9	^DMName[9]
(10)	Per10	^DMName[10]
(11)	Per11	^DMName[11]
(12)	Per12	^DMName[12]
(13)	Per13	^DMName[13]
(14)	Per14	^DMName[14]
(15)	Relative	^Rel
(16)	Friends	^Fri
(17)	LAHelp	^LAH
(18)	Domestic	^Dom
(19)	Nurse	^Nur
(20)	Helpers	^Hel

```
CHECK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR ixx := 1 TO 14
     AND: ixx IN WhoLook
     (DMName[[ixx] <> '') AND INVOLVING(WhoLook)
```

^N Record valid code for carer.

```
CHECK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: Relative IN WhoLook
     (Rel <> '') AND INVOLVING(WhoLook)
```

^N Record valid code for carer.

```
CHECK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: Friends IN WhoLook
     (Fri <> '') AND INVOLVING(WhoLook)
```

^N Record valid code for carer.

```
CHECK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: LAHelp IN WhoLook
     (LAH <> '') AND INVOLVING(WhoLook)
```

^N Record valid code for carer.

```
CHECK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: Domestic IN WhoLook
     (Dom <> '') AND INVOLVING(WhoLook)
```

^N Record valid code for carer.

```
CHECK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: Nurse IN WhoLook
     (Nur <> '') AND INVOLVING(WhoLook)
```

^N Record valid code for carer.

```
CHECK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: Helpers IN WhoLook
     (Hel <> '') AND INVOLVING(WhoLook)
```

^N Record valid code for carer.

```
CHECK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: LNeedPer > 14
     NOT((((((IN(Relative,WhoLook)) OR (IN(Friends,WhoLook))) OR
     (IN(LAHelp,WhoLook))) OR (IN(Domestic,WhoLook))) OR
     (IN(Nurse,WhoLook))) OR (IN(Helpers,WhoLook)))
```

^I^BLU^IC That code is invalid^I

```
CHECK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: LNeedPer <= 14
     NOT(IN(LNeedPer,WhoLook))
```

^I^BLU^IC You've included ^LNeedName as looking after him-/herself. Please remove him/her from the answer at WhoLook.^I

```
CHECK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 14
     AND: Count IN WhoLook
     DMAge[[Count] >= 0
```

^I^BLU^IC Code ^Count is not valid for this question.^I

```
CHECK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 14
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 14
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 14
     RESERVECHECK
```

RESERVECHECK

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Relative
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Relative
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Relative
```

helper := 'the relative'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Relative
```

HCount := 15

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Friends
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Friends
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Friends
```

helper := 'the friend'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Friends
```

HCount := 16

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = LAHelp
     AND: NatCen = NI
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = LAHelp
     AND: NatCen = NI
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = LAHelp
     AND: NatCen = NI
```

helper := ('the Social Services home help or ' + 'home care worker')

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = LAHelp
     AND: NOT (NatCen = NI)
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = LAHelp
     AND: NOT (NatCen = NI)
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = LAHelp
     AND: NOT (NatCen = NI)
```

helper := 'the LA home help or home care worker'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = LAHelp
```

HCount := 17

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Domestic
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Domestic
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Domestic
```

helper := 'the private domestic help'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Domestic
```

HCount := 18

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Nurse
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Nurse
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Nurse
```

helper := 'the nurse'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Nurse
```

HCount := 19

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Helpers
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Helpers
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Helpers
```

helper := 'the outside helper'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Helpers
```

HCount := 20

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per1
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per1
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per1
```

helper := DMName[[1]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per1
```

HCount := 1

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per2
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per2
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per2
```

helper := DMName[[2]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per2
```

HCount := 2

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per3
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per3
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per3
```

helper := DMName[[3]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per3
```

HCount := 3

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per4
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per4
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per4
```

helper := DMName[[4]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per4
```

HCount := 4

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per5
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per5
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per5
```

helper := DMName[[5]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per5
```

HCount := 5

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per6
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per6
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per6
```

helper := DMName[[6]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per6
```

HCount := 6

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per7
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per7
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per7
```

helper := DMName[[7]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per7
```

HCount := 7

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per8
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per8
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per8
```

helper := DMName[[8]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per8
```

HCount := 8

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per9
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per9
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per9
```

helper := DMName[[9]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per9
```

HCount := 9

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per10
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per10
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per10
```

helper := DMName[[10]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per10
```

HCount := 10

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per11
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per11
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per11
```

helper := DMName[[11]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per11
```

HCount := 11

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per12
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per12
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per12
```

helper := DMName[[12]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per12
```

HCount := 12

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per13
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per13
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per13
```

helper := DMName[[13]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per13
```

HCount := 13

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per14
```

does := 'does'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per14
```

has := 'has'

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per14
```

helper := DMName[[14]

```
COMPUTE IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: WhoLook[Count] = Per14
```

HCount := 14

```
ASK IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
```

FRS0605B.QCare.QRecHelp.Recip[].Hour

^I QCare
^I

^IS^I U^I
^N About how many hours a week, on average, does ^helper spend actually providing help for or looking after ^LNeedName?^N

(1)	upto4	0-4 hours per week
(2)	upto9	5-9 hours per week
(3)	upto19	10-19 hours per week
(4)	upto34	20-34 hours per week
(5)	upto49	35-49 hours per week
(6)	upto99	50-99 hours per week
(7)	over100	100 or more hours per week
(8)	var20	Varies - under 20 hours per week
(9)	var2034	Varies - 20-34 hours per week
(10)	var35	Varies - 35 hours a week or more

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     AND: (Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE
     AND: In loop FOR Count := 1 TO 5
     AND: Count <= WhoLook.CARDINAL
     AND: Edit <> Yes
     AND: HCount <= 14
     AND: DMAge[HCount] IN [1 .. 15]
     Hour[Count] <> over100
```

^I Are you sure that the child(ren) are looking after ^LNeedName for 100 hours a week or more?
If so, suppress warning.^I

Ask if: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
 And: *In loop FOR Idx := 1 TO 5*
 And: *Idx <= QNeedPer.CARDINAL*
 And: *(Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE*
 And: *In loop FOR Count := 1 TO 5*
 And: *Count <= WhoLook.CARDINAL*

FRS0605B.QCare.QRecHelp.Recip[].HowLng

^I QCare
^I

^IS^I V^I
Thinking of when ^helper first began providing help for or looking after ^LNeedname, about how long ^has ^helper been providing help for or looking after ^LneedName?

^I^BLU^IC Prompt as necessary
^I^IC Record how long care has been given regardless of the number of hours the carer gives now or has given in the past.

(1)	Less6M	Less than 6 months
(2)	LessYr	6 months, less than 1 year
(3)	Less3Yr	1 year, less than 3 years
(4)	Less5Yr	3 years, less than 5 years
(5)	Less10Yr	5 years, less than 10 years
(6)	Less15Yr	10 years, less than 15 years
(7)	Less20Yr	15 years, less than 20 years
(8)	More20	20 years or more

Warn if: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
 And: *In loop FOR Idx := 1 TO 5*
 And: *Idx <= QNeedPer.CARDINAL*
 And: *(Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE*
 And: *In loop FOR Count := 1 TO 5*
 RESERVECHECK

RESERVECHECK

Warn if: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
 And: *In loop FOR Idx := 1 TO 5*
 And: *Idx <= QNeedPer.CARDINAL*
 And: *(Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE*
 And: *In loop FOR Count := 1 TO 5*
 RESERVECHECK

RESERVECHECK

Warn if: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
 And: *In loop FOR Idx := 1 TO 5*
 And: *Idx <= QNeedPer.CARDINAL*
 And: *(Freq IN [Continuously .. OWeek]) OR Freq = NONRESPONSE*
 And: *In loop FOR Count := 1 TO 5*
 RESERVECHECK

RESERVECHECK

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Idx := 1 TO 5
     AND: Idx <= QNeedPer.CARDINAL
     RESERVECHECK
```

RESERVECHECK

FRS0605B.QCare.QRecHelp (continued)

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
AND: *In loop FOR Idx := 1 TO 5*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
AND: *In loop FOR Idx := 1 TO 5*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
AND: *In loop FOR Idx := 1 TO 5*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
AND: *In loop FOR Idx := 1 TO 5*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
RESERVECHECK

RESERVECHECK

FRS0605B.QCare (continued)

Questions about carers/cared for

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: QRecHelp.QNeedPer <> EMPTY AND (NeedHelp = Yes)
     ((((((((((((IN(Per1,QRecHelp.QNeedPer)) OR
     (IN(Per2,QRecHelp.QNeedPer))) OR (IN(Per3,QRecHelp.QNeedPer))) OR
     (IN(Per4,QRecHelp.QNeedPer))) OR (IN(Per5,QRecHelp.QNeedPer))) OR
     (IN(Per6,QRecHelp.QNeedPer))) OR (IN(Per7,QRecHelp.QNeedPer))) OR
     (IN(Per8,QRec
```

^I Please include the household member who receives regular help, or change 'NeedHelp' to 'No'.^I

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Index1 := 1 TO HHSize
     AND: NeedHelp <> EMPTY AND (Index1 IN QRecHelp.QNeedPer)
     NeedHelp = Yes
```

^I You have coded a household member as receiving regular help, so please change 'NeedHelp' to 'Yes', or remove the household member from 'QNeedPer'.^I

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: QRecHelp.QNeedPer <> EMPTY AND (GiveHelp = Yes)
     (((((((IN(Per15,QRecHelp.QNeedPer)) OR (IN(Per16,QRecHelp.QNeedPer)))
     OR (IN(Per17,QRecHelp.QNeedPer))) OR (IN(Per18,QRecHelp.QNeedPer))) OR
     (IN(Per19,QRecHelp.QNeedPer))) OR (IN(Per20,QRecHelp.QNeedPer))) OR
     (IN(Per21,QRecHelp.QNeedPer))) OR (IN(Per22,QR
```

^I Please include the non-household member receiving help from someone in the household, or change 'GiveHelp' to 'No'.^I

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Index1 := 15 TO 22
     AND: GiveHelp <> EMPTY AND (Index1 IN QRecHelp.QNeedPer)
     GiveHelp = Yes
```

^I You have coded a non-household member as receiving help, so please change 'GiveHelp' to 'Yes', or remove the non-household member from 'QNeedPer'.^I

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Index1 := 15 TO 22
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Index1 := 15 TO 22
     RESERVECHECK
```

RESERVECHECK

```
WARN IF: (NeedHelp = Yes) OR (GiveHelp = Yes)
     AND: In loop FOR Index1 := 15 TO 22
     RESERVECHECK
```

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
AND: *In loop FOR Index1 := 15 TO 22*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
AND: *In loop FOR Index1 := 15 TO 22*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
AND: *In loop FOR Index1 := 15 TO 22*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
AND: *In loop FOR Index1 := 15 TO 22*
RESERVECHECK

RESERVECHECK

WARN IF: *(NeedHelp = Yes) OR (GiveHelp = Yes)*
AND: *In loop FOR Index1 := 15 TO 22*
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

WARN ALWAYS:
RESERVECHECK

RESERVECHECK

FRS0605B (continued)

FAMILY RESOURCES SURVEY 2006-2007

```
CHECK IF: In loop FOR Loop1 := 1 TO 14
     AND: Loop1 IN QCare.QRecHelp.QNeedPer
     DMAge[Loop1] >= 0
```

^I^BLU^IC Code ^Loop1 is not valid for this question.^I

```
COMPUTE IF: In loop FOR Loop1 := 1 TO 14
     AND: In loop FOR Loop2 := 1 TO 5
     AND: Loop1 IN QCare.QRecHelp.Recip[Loop2].WhoLook
```

DMCarer[Loop1] := Yes

```
CHECK IF: In loop FOR Loop1 := 1 TO 14
     AND: In loop FOR Loop2 := 1 TO 5
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: In loop FOR Loop1 := 1 TO 14
     AND: In loop FOR Loop2 := 1 TO 5
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: In loop FOR Loop1 := 1 TO 14
     RESERVECHECK
```

RESERVECHECK

```
CHECK IF: In loop FOR Loop1 := 1 TO 14
     RESERVECHECK
```

RESERVECHECK

```
CHECK ALWAYS:
     RESERVECHECK
```

RESERVECHECK

```
CHECK ALWAYS:
     RESERVECHECK
```

RESERVECHECK

```
CHECK ALWAYS:
     RESERVECHECK
```

RESERVECHECK

```
CHECK ALWAYS:
     RESERVECHECK
```

RESERVECHECK

ASK ALWAYS:

FRS0605B.EndDisp

^I^BLU^IC@|-@|End of 'Household' Schedule.
Now administer 'Benefit Unit' Schedule(s).
B.U. members
^NameInBU[1]^NameInBU[2]^NameInBU[3]^NameInBU[4]^NameInBU[5]
^NameInBU[6]^NameInBU[7]

Total number of Benefit Units = ^NewBU

Press <Ctrl + Enter> to select Benefit Unit or to fill in Admin details.
Alternatively, press <1> and <Enter> to start the first Benefit Unit.^I

1..1

RECORD ALWAYS:

FRS0605B.HHTime

^I Time taken from interview start to end of household grid.

Only visible for testing purposes, just press <Enter>.^I

TIME

RECORD ALWAYS:

FRS0605B.HHMins

^I Total minutes in household grid.

Only visible for testing purposes, just press <Enter>.^I

0.00..1440.00

COMPUTE IF: *HHTime = EMPTY AND EndDisp <> EMPTY*

```
HHTime := SYSTIME
```

COMPUTE IF: *HHTime = EMPTY AND EndDisp <> EMPTY*

```
HHMins := ((HHTime.ABSTIME - QSignIn.IntSTime.ABSTIME) /
60000)
```

ASK IF: *Test = Yes*

FRS0605B.HHTime

^I Time taken from interview start to end of household grid.

Only visible for testing purposes, just press <Enter>.^I

TIME

Ask if: *Test = Yes*

FRS0605B.HHMins

^I Total minutes in household grid.

Only visible for testing purposes, just press <Enter>.^I

0.00..1440.00

Check always:
RESERVECHECK

RESERVECHECK

Check always:
RESERVECHECK

RESERVECHECK

Check always:
RESERVECHECK

RESERVECHECK

Check always:
RESERVECHECK

RESERVECHECK

Compute always:

NBusRooms := 0

Compute if: *In loop FOR Loop1 := 1 TO NewBU*

AdInBU[1] := 1

Compute if: *In loop FOR Loop1 := 1 TO NewBU*

AdInBU[2] := 1

Compute if: *In loop FOR Loop1 := 1 TO NewBU*

Child1 := 0

Compute if: *In loop FOR Loop1 := 1 TO NewBU*

Child2 := 0

Compute if: *In loop FOR Loop1 := 1 TO NewBU*

Child3 := 0

Compute if: *In loop FOR Loop1 := 1 TO NewBU*

Child4 := 0

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU

Child5 := 0
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU

Child6 := 0
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU

Child7 := 0
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU

Child8 := 0
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend = Adult
     AND: AdInBU[1] = EMPTY

AdInBU[1] := Loop2
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend = Adult
     AND: AdInBU[2] = EMPTY

AdInBU[2] := Loop2
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child1 = 0

Child1 := Loop2
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child2 = 0

Child2 := Loop2
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child3 = 0

Child3 := Loop2
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child4 = 0

Child4 := Loop2
```

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child5 = 0
```

Child5 := Loop2

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child6 = 0
```

Child6 := Loop2

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child7 = 0
```

Child7 := Loop2

```
COMPUTE IF: In loop FOR Loop1 := 1 TO NewBU
     AND: In loop FOR Loop2 := 1 TO HHSize
     AND: Loop1 = ABen[Loop2]
     AND: PRec[Loop2].Depend IN [DepAd .. Child]
     AND: Child8 = 0
```

Child8 := Loop2