

# MCBS MAIN STUDY

## R67 General Specifications for Blaise/WVS

Version 19.01.0001

### TABLE OF CONTENTS

1. Specification Terminology
2. Database Notation and Database “Deleted” flags.
3. Specification Items
4. Specification Standards
5. Cost Payment Summary: Setting COST.CPREASON
6. Roster Requirements

Other General Specification Documents include:

- Default Specifications
  - A quick reference document, summarizing the more common default specifications outlined in the general specifications.
- General Specifications Questionnaire Flow
  - Summaries components of the questionnaire
  - Specifies the order sections are administered in the questionnaire
  - Includes an overview of collecting COST data in ST, PS, NS, CPS.
- General Specifications Screen Types
  - Summary of WVS page layout
  - List of each screen type with required specifications
  - Sample WVS SpecWriter Specifications per screen type
  - \* *Specification in progress..*
- General Specifications Global Displays
  - Respondent Display fills, such as “[you/(SP)]”.
  - Reference Period Date Displays
  - General Date Displays
  - Plan Status Displays
  - Event Displays

- COST Series Displays
- Event and COST Context Header Displays
- Claim Control Number Displays
- Charges and Amounts remaining Displays
- Person and Plan name Displays
- IA custom Display fills.
- Address and Phone number Displays.
- General Specifications Dates
  - Summary of Reference periods and previous round
  - Date Entry and Display formats
  - Date Edit logic, summarizes when dates should not invoke edits due to DK and RF.
  - Specifications for setting ORP flag
  - Logic for checking overlapping and matching dates
  - Date calculation logic
- General Specifications Database
  - Overview of Cheshire Segments, record key structure, and corresponding section in Questionnaire.
  - Summary of critical fields used to test complex routing in Questionnaire  
\* *Specification in progress.*
  -
- General Specifications Attachments
  - Includes Attachments referenced in Section Specs.
- Section Overview
  - Includes an overview of each section, including Roster patterns  
\* *Specification in progress.*
- HI1 Attachment
- HI2 Attachment
- Specification Document Overview
  - Summarizes Specification document filenames and content.

# **1. SPECIFICATION TERMINOLOGY**

---

## **SAMPLE PERSON / RESPONDENT**

**SAMPLE PERSON** Refers to the person who has been selected to participate in study.  
We are collecting data about the Sample Person.

**SP** SP is used as an abbreviation for “Sample Person”.

**RESPONDENT** The Respondent is the person who is being interviewed.  
The Respondent may be the SP or a person answering questions for the SP, known as the Proxy.

**SAMPLE** We often refer to SP as being in one of the following “Samples”:

### **SUPPLEMENTAL SAMPLE**

Refers to an SP in their first interview in the study. Also known as “New Sample”.

**EXIT SAMPLE** Refers to an SP in their last interview in the study.

**CONTINUING SAMPLE** Refers to an SP who has received at least one community interview.

### **ORD or DUAL ELIGIBLE SAMPLE**

Refers to special types of Supplemental Sample SPs. These Sample Types are no longer used in the study, however are still reflected in the specifications.

## **ROUNDS**

The study conducts data collection in consecutive “Rounds”.  
Each Round is numbered sequentially.  
Three Rounds of data is collected per year.  
Each round is referenced as:

**FALL ROUND** September through December

**WINTER ROUND** January through April

**SUMMER ROUND** May through August

New Sample Persons are added to the study every Fall Round. It is often referred to as the **Supplemental Round**.

An SP will remain in the study for 12 rounds.

The SP will then exit the study in the 12<sup>th</sup> round, which will be a Summer Round. This round is often referred to as the **Exit Round**.

## **COMPONENTS**

MCBS data is collected in two different components, the Community and Facility Components.

**Community Component**                      SP resides in the community.

**Facility Component**                      SP resides in an eligible facility/nursing home.

Each component has its own questionnaire, the Community Questionnaire and the Facility Questionnaire. MCBS conducts the appropriate questionnaire based on the SP's current residence. When conducting a community interview, if we discover that the SP has already moved into a facility since the last interview, we will complete the community interview and will conduct the next interview in the facility, administering the Facility Questionnaire. The same is true if an SP has moved from a facility into the community.

The SP may receive multiple interviews in a single round. While in the community, the SP may only receive one community interview in a single round; however, if the SP moves into a facility, the SP may receive an additional facility interview in the same round. If the SP moves to multiple facilities in a single round, they may receive an interview for each facility where they have resided during that round. An exception to this rule is the SP's 1<sup>st</sup> interview in the study. An SP will only receive one interview in their 1<sup>st</sup> round in the study, either community or facility.

An SP may skip a round, but only one consecutive round.

When an SP moves from one component to the next, it is referred to as a **Crossover Case**.

## **INTERVIEW TYPES**

The SP's Interview Type is determined prior to fielding the case.

**MRES.INTTYPE** stores SP's Interview Type:

INTTYPE=1/StandardHadPrev  
INTTYPE=2/NewFromFacility  
INTTYPE=3/NewFromSupplement  
INTTYPE=4/StandardSkippedPrev  
INTTYPE=5/LastRndFacSum  
INTTYPE=6/LastRndFacBase  
INTTYPE=7/SupSmp1stTimeUtil  
INTTYPE=8/ExitInterviewHadPrev  
INTTYPE=9/ExitInterviewSkipPre  
INTTYPE=10/SupSmp1stTimeUtilSkipped

### **VALID IN ROUNDS:**

ALL  
ALL  
FALL ONLY  
ALL  
ALL  
ALL  
WINTER ONLY  
SUMMER ONLY  
SUMMER ONLY  
SUMMER ONLY

**TYPE #**

- 1 Standard Continuing SP.  
SP's last interview was in the Community.
- 2 SP's last interview was in a facility.  
SP's has not had a community interview.
- 3 SP is in the Supplemental Sample, their 1<sup>st</sup> round in the study.
- 4 Standard Continuing SP.  
SP skipped the previous round.
- 5 SP's last interview was in a facility.  
SP's last community interview was 2 rounds ago.
- 6 SP's last interview was in a facility.  
SP's last community interview was 3 or more rounds ago.
- 7 SP's 2<sup>nd</sup> interview.  
Utilization and COST data is collected for the 1<sup>st</sup> time.
- 8 SP's last interview.  
SP is in the Exit Sample.  
SP's last community interview was in the previous round.
- 9 SP's last interview.  
SP is in the Exit Sample  
SP's last community interview was 2 rounds ago.
- 10 SP's 2<sup>nd</sup> interview.  
SP skipped previous round.  
Utilization and COST data is collected for the 1<sup>st</sup> time.

## **2. DATABASE NOTATION**

---

Data Collection involves three different database structures: Cheshire, Blaise, SQL.

### **CHESHIRE DATABASE**

MCBS data has been collected in a DOS based system known as Cheshire. The input data is stored in a Cheshire database. Although data will now be collected into a system known as WVS, the Cheshire database will be maintained.

The Cheshire database is organized into tables known as **Segments**. Input fields are stored in **Variables** on a designated Segment. The variable notation is:

**Segment.VariableName**

**EX:**

**ROST.ROSTFNAM.**

The data collected from the community and facility questionnaires is stored in separate Cheshire databases. The community questionnaire collects data about the SP only and the data is stored based on the SP's CASE ID. The facility questionnaire collects data about the facility as well as each SP that resides at the facility; therefore, the data is stored based on the Facility's CASE ID. The facility database can be linked back to the community database using the SP's CASE ID.

### **BLAISE / SQL DATABASE**

MCBS will now be collected in a Windows based system known as WVS. WVS utilizes a COTS system known as Blaise. In WVS, input values are first stored in Blaise Fields. The Blaise Field is then copied to a SQL Field during data collection.

The SQL database is also organized into Tables, similar to the Cheshire database. The SQL field notation is:

**Table.FieldName**

**EX:**

**tblROST.ROSTFNAM**

To summarize:

Three Field Names are associated with each input field specified in WVS:

1. Input field            The input field is collected into a Blaise FieldName.
2. SQL field             The Blaise field is then copied to a SQL Table.FieldName during data collection.
3. Cheshire variable    The SQL field is transformed back to the Cheshire Segment.VariableName after data collection.

## **ROUND SPECIFIC DATABASE RECORDS**

If an input field is specified that is stored on a Round specific record, the default is to collect and store data on the current round record unless other specified. The exception to this rule is data collected/updated in HIS, UTS, PMS and CPS. These summary sections will collect/update data on PREVIOUS ROUND records.

## **DATA TRANSFORMATION**

In order to collect data in WVS, the Cheshire data is transformed from the Cheshire Segment.VariableName(s) to the SQL Table.FieldName(s). Once the data is collected in SQL, the SQL Table.FieldName(s) is then transformed back to the Cheshire Segment.VariableName(s). The default naming convention is:

Blaise FieldName = SQL FieldName  
SQL FieldName = Cheshire VariableName.  
SQL SQL Table Name = tbl + Cheshire SegmentName

### **EX:**

Blaise FieldName	= <b>ROSTFNAM</b>
SQL Table.FieldName	= <b>tblROST.ROSTFNAM</b>
Cheshire Segment.VariableName	= <b>ROST.ROSTFNAM</b>

However, there are some exceptions to the naming conventions due to differences between WVS and Cheshire. See Data Transformation for a list of exceptions.

Because WVS utilizes the Blaise system, WVS specifications often refer to Blaise system/programming terminology. The WVS Specifications most often references the Question Item Tag and Blaise Input Field Name when referencing an input value from a designated question.

### **EX:**

ST1 – MHMOSTMT

However, if the specification references a database field that cannot easily be linked back to a specific question, it will reference the Cheshire Segment.VariableName.

### **EX:**

HRND.MHMOSTMT

It is understood that a link will be made between the Cheshire Segment.VariableName to the corresponding SQL Table.FieldName. The link is included in programming technical documents. The goal is to include this link in the Specification Database.

## **INPUT FIELD VS. BACKGROUND VARIABLE**

The terms “Field” and “Variable” have the same meaning. “Variable” is the term used in Cheshire to reference a database field.

Most data collected will be entered directly into an **Input field**. There will be at least one Input field specified for each Question in the instrument. A Question can have multiple Input fields.

An Input field may be considered “Temporary”. A “Temporary” input field is an Input field that is collected and stored during data collection only and will not be transformed back to the Cheshire database.

In addition to Input fields, **Background database fields** may be set “in the background” based on the response to a Question. Background fields (or sometimes referred to as Background Variables) are often specified when creating database records, setting “flags” that affect routing or display fills, storing “calculations”, or as a method of storing data collected in “Temporary” Input fields into a database field that will be transformed back to the Cheshire database.

## **DATABASE RECORDS FLAGGED AS “DELETED”**

The following Database tables include flags that indicate that the database record is no longer active and should not be included in Roster, Report or Grid displays, should not be included as a valid record in any programming logic or checks. These records are considered as “invalid”.

EVNT                      EVNT.EVNTDFLG = 1/Yes

\* Exception: If HP/HF Event exists on EVNT, and EVNT.EVNTDFLG = 1/Yes, if Provider selected as a current round Home Health provider in the current round, we will update the existing EVNT.EVNTDFLG = empty and reactive the original EVNT record. See BOX HH1AA.

HERO                      HERO.HERODFLG = 1/Yes

PMRO                      PMRO.PMRODFLG = 1/Yes

PLAN                      PLAN.PLANHIDE = 1/Yes                      NO LONGER USED

PLAN.PLANDFLG = 1/Yes

PLAN.MHMODFLG = 1/Yes

PLAN.LOSEPLFG ^= EMPTY

\* Exception: If Medicaid or Tricare plan exists on PLAN, and PLAN.PLANDFLG = 1/Yes or PLAN.LOSEPLFG = 1/Yes, if Medicaid or Tricare is added in the current round, we will update the existing PLAN.PLANDFLG = empty and/or PLAN.LOSEPLFG = 1/Yes and reactive the original PLAN record. See HI5, HIT1.



DMEM	DMEM.LOSEDMFG ^= EMPTY	
OSOP	OSOP.OSOPHIDE = 1/Yes	NO LONGER USED
	OSOP.LOSEOSFG ^= EMPTY	
PAYM	PAYM.PAYMDFLG = 1/Yes	
XCEV	XCEV.DELLINK = 1/Yes	

### 3. SPECIFICATION ITEMS

---

WVS specifications include two basic elements, **Questions** and **Route Boxes**. A Question is equivalent to a **Page** displayed in the WVS CAPI instrument. It collects data into **Input Fields**. The Route Box is a separate specification item that specifies complex routing between Questions in the CAPI instrument. Both Questions and Route Boxes are labeled by a unique **Item Tag**.

#### ITEMS SPECIFIED FOR QUESTIONS

<b>Item Tag</b>	Question Number. Matches Cheshire Screen Name. Format = Section ID + Numeric value EX: OM1
<b>Design Screen Type</b>	Defines layout/intent of screen. *See list of Screen Types
<b>Fields</b>	List of input field names in the order they are collected.
<b>Enable Functions:</b>	Yes/No WVS system functions enabled: HELP                      Disregard Spec. No longer implemented. COMMENTS              Disregard Spec. Should always be set to YES. JUMPBACK              Disregard Spec. Not currently supported.
<b>Roster Name:</b>	<b>Should read “Roster Pop-Up Name”.</b> Name of Roster Pop-Up Window specifications linked to Roster question.
<b>Roster Type:</b>	Single Item Select / Multiple Item Select  Single Item Select: Only one item can be selected from roster.  Multiple Item Select: More than one item can be selected from roster. An additional “limit” to the number of items that may be selected on Multiple Item Rosters to meet WVS database/system needs. This limit will be specified in General Specifications.  No Selection Required: Currently, Roster Type does not allow for “No Selection Required”.  “SELECTION NOT REQUIRED AT THIS ROSTER” should be added to the Roster Display instructions if no selection is required on the roster.

<b>Roster Functions:</b>	<p>Yes/No</p> <p>Roster Functions enabled:</p> <p>ADD ITEM</p> <p>EDIT ITEM</p> <p>DELETE ITEM</p> <p>SEARCH ITEM</p> <p>A separate field in SpecWriter stores the Function display text:</p> <p>EX: Display as ‘Add a Plan’</p>
<b>Grid Functions:</b>	NO LONGER SUPPORTED BY WVS.
<b><u>DISPLAY INSTRUCTIONS:</u></b>	Display instructions specify Display Fills, text layout and various text styles (color, underline, etc).
<b>Context Header display:</b>	Display instructions for Context header.
<b>Question display:</b>	<p>Display instructions for Question text.</p> <p>Display instructions for Interviewer Instructions I and II.</p>
<b>Multi Field display:</b>	<p>Vertical / Horizontal</p> <p>Enter additional complex screen layout specifications in Multi-Field Display Instructions. This specification item appears directly under Vertical/Horizontal alignment in the Full Specification report.</p> <p>The default is to display Multiple Input fields as “overlays”. However, an additional specification may be included to “display all input fields at the same time, not as overlays” that overrides the default. This will most often be used for Quantity Unit screen types.</p>
<b>Roster/Grid instructions:</b>	<p>Roster/Grid load instructions include:</p> <ul style="list-style-type: none"> <li>Items to be displayed in roster/grid</li> <li>Each item is displayed as a separate row in roster/grid.</li> <li>Order rows are to be displayed</li> <li>What rows are Protected</li> <li>Roster header</li> </ul>
<b>Roster/Grid display:</b>	<p>Roster/Grid column instructions include:</p> <ul style="list-style-type: none"> <li># of columns</li> <li>Column headers – WVS only supports UPPER CASE headers</li> <li>Data displayed in roster columns</li> <li>Data displayed/collected in grid columns</li> <li>Display fill instructions</li> <li>Include “SELECTION NOT REQUIRED AT THIS ROSTER.” if</li> </ul>

selection is not required.

**Report Display:** Report display instructions include:  
Items displayed in report  
Each item is displayed as a separate row in report.  
Data displayed per column  
Order rows are to be displayed  
Report header  
Column headers  
Location of report, above or below question text

**TEXT:**

**Context Header:** Context Header text.

**Int. Instr I:** Interviewer Instructions I text.

**Question Text:** Question text.

**Int. Instr II:** Interviewer Instructions II text.

NOTE: Additional text may be specified to be displayed with each input field. See Item Text and Item Label.

**INPUT FIELDS / ROUTING**

**Field:** Input Blaise FieldName.

**Cheshire Name:** **Segment.VariableName**

Specifies the Cheshire Segment.VariableName associated with input field.

**TEMP:** Is specified when the input field is considered “temporary” in Cheshire and there is no Cheshire Variable Name associated with this question. It is understood that these fields will not be transformed back to the Cheshire database.

**NONE:** Is specified when a new Field Name is generated that does not exist in the Cheshire Database. Typically, a list of Cheshire Variable Names associated with the input field is listed. There are special circumstances where we are collecting data into new fields during data collection and will be transforming the data back to the original Cheshire Variable Name(s).  
See Data Transformation and Field Name.

<b>Item Text display:</b>	<p>Display instructions for Item text.</p> <p>EX: Display (SPECIFY) in the same blue font used for interviewer instructions.</p>										
<b>Item Text:</b>	<p>Additional text that is displayed with the Input field only.</p> <p>Is used to provide additional instructions to the interviewer if the Input field is displayed.</p> <p>EX: OTHER (SPECIFY)</p> <p>Is displayed above the Input field, unless specified for LIST/GRID.</p> <p>LIST screens: Item Text should be specified for each unique Question in the LIST. It is displayed to the left of the LIST Response Categories.</p> <p>GRID screens: Item Text specified for an input field on a GRID is displayed only when that Input Column is “active”. It is displayed above the GRID.</p>										
<b>Label:</b>	<p>Text that will be displayed to the Left or Under the Input field.</p> <p>Is typically used to further describe input expected.</p> <p><b>EX:</b></p> <p>AMOUNT:</p> <p>NOTE: \$ and % symbols are specified using a MASK. See MASK instructions.</p>										
<b>Label Position:</b>	<p>Left/Under</p> <p>Location of Label in relationship to input field.</p>										
<b>Field Type:</b>	<p>Type of input field:</p> <table> <tr> <td>Enumerated</td><td>= list of response alternatives</td></tr> <tr> <td>String</td><td>= text field</td></tr> <tr> <td>Integer</td><td>= numeric field, no decimals</td></tr> <tr> <td>Real</td><td>= numeric field, decimals</td></tr> <tr> <td>Open</td><td>= Open text field</td></tr> </table>	Enumerated	= list of response alternatives	String	= text field	Integer	= numeric field, no decimals	Real	= numeric field, decimals	Open	= Open text field
Enumerated	= list of response alternatives										
String	= text field										
Integer	= numeric field, no decimals										
Real	= numeric field, decimals										
Open	= Open text field										
<b>Type Name:</b>	<p>WVS Blaise Field Type Name.</p> <p>Blaise Type Names are defined to simplify specification and programming.</p> <p>Blaise Type Names define common field attributes:</p> <ul style="list-style-type: none"> <li>The Field Type (Enumerated, Integer...)</li> <li>Field Length</li> <li>Minimum Value</li> </ul>										

Maximum Value  
Response alternatives  
Attributes (DK, RF, EMPTY)

The Type Name is defined once and is then linked to multiple input fields.

EX: TYesNoDKRF

Is used for all YES/NO input fields where the response alternatives equal:

1/Yes

2/No

Attributes = DK and RF

The default is to name the Blaise Type Name to match the corresponding  
Cheshire range name (if it exists):

T + Cheshire range name

**Answers Allowed:**

Numeric Value.

Specifies the maximum number of response alternatives that can be  
selected at an enumerated field.

Applies to CODE 1/CODE ALL questions only.

Answers Allowed > 1 indicates a CODE ALL question.

Answers Allowed = 1 indicates a CODE 1 question.

Default setting = maximum number of response alternatives, excluding  
DK and RF. Cannot exceed maximum number of response alternatives.

Answers Allowed = 1 will be specified at all other field types, including  
Roster input fields.

**Drop Down List:**

Yes/No

Default = No.

If “Yes”, Enumerated response is displayed as a drop down list.

This option is often used in a GRID to minimize space requirements.

**Lookup File:**

Yes/No

Default = No.

If “Yes”, response will be selected from an external Look-up File.

**Lookup File Name:**

Name of Lookup File.

**Field Size:**

Maximum Length of Input for String or Open field types.

Length is not required for Integer and Real field types. The length of  
integer and real fields is determined by Max Value.

Input values that exceed Field Size trigger a system error message.

**Min Value:** Minimum value for system range.  
 Default = Cheshire range minimum value.  
 Input values that are less than Min Value trigger a system error message.

**Max Value:** Maximum value for system range.  
 Default = Cheshire range maximum value.  
 Input values that are greater than Max Value trigger a system error message.

**Mask:** A mask further specifies the display of input field:

<b>Dollars</b>	Displays REAL fields with decimal. Displays “\$” left of input field.
<b>Dollars No Cents</b>	Displays REAL fields without decimal. Displays “\$” left of input field.
<b>Percent</b>	Displays % right of input field.

## INPUT FIELDS:

### Example of an Enumerated Input field:

Number	Label	Route
1	Yes <b>English text:</b> YES	BOX ER6 (ERQ1365)
2	No <b>English text:</b> NO	ER3B – HMOREFER (ERQ1080)

**Number** Numeric value assigned to response alternative.  
 Default = Matches Cheshire WORD specifications.  
 The Cheshire database will continue to store the numeric value after data transformation.

**Label** WVS Blaise Label assigned to response alternative.  
 Another method for referencing numeric value.  
 Default = abbreviate English Text, no spaces allowed. Upper and Lower case is used for easy recognition.

Unlike Cheshire, WVS SQL field stores the label, not the numeric value. Therefore, both the label and numeric value are referenced in specifications.

**English text:** Response text that is displayed on screen  
 Default = ALL CAPS.  
 Default = Matches Cheshire WORD specifications.

**Route:** The route is specified for each response alternative.  
 The Route specifies the destination Item tag and field name.

Value in parenthesis is an internal ID in SpecWriter.  
 Additional routing instructions may be included to specify complex routing on multi-field screens.  
 “DO NOT DISPLAY” is specified when a response alternative should not be displayed on screen.

**Example of other field types:      STRING, REAL, INTEGER, OPEN**

<b>Number</b>	<b>Label</b>	<b>Route</b>
1	[Continuous answer.]	BOX ER6 (ERQ1365)

All Non-Enumerated fields are displayed in a similar format as Enumerated fields in the specifications.

Number and Label are always set to:  
                   1      [Continuous answer]

Only one route is specified.

## **ATTRIBUTES**

Don't Know	Allow Don't Know response
Refusal	Allow Refusal response
Empty	Allow non-response

Attributes are used to indicate whether or not Don't Know, Refused or Empty (non-response) is a valid response for input field.

If “Attribute” is a valid response, it will be specified with its appropriate route instruction.

In Cheshire, DK and RF are specified as part of the variable's response alternatives. In WVS, DK and RF are considered attributes of the field and are invoked using a system function.

## **BACKGROUND VARIABLE ASSIGNMENTS**

Background Variable Assignments are stored in multiple fields in SpecWriter:

**Assignment Summary**  
**Field Name + Assignment Instructions**

### **Assignment Summary:**

The Assignment Summary is displayed first in the Full Specification Report.

It is used to:

Specify fields populated in Roster Pop-Up windows.  
 Specify when database records are generated.



Specify additional conditions for setting background fields.  
Summarize why background fields are set.

**Field Name + Assignment Instructions:**

Blaise/SQL Field name + instructions.

**SOFT EDIT CHECKS**

It is also known as a “SIGNAL” in Blaise.

A Soft Edit is specified to check for inconsistency in data.

If input data causes the edit check to fail, an edit window will display an error message and a list of fields involved in the edit check.

The interviewer can select SUPPRESS to ignore the edit warning and continue with the interview or can navigate back to an input field and correct the data.

SOFT EDIT CHECKS are not supported on Rosters.

**HARD EDIT CHECKS**

A Hard Edit is specified to check for inconsistency in data.

If input data causes the edit check to fail, an edit window will display an error message and a list of fields involved in the edit check.

The interviewer must navigate back to an input field and correct the data before continuing.

A Hard edit is not necessary if check is covered by the input field Min and Max values.

In some cases, the programmer may be able to implement the logic specified at the hard edit as a WVS system range error. When this occurs, the program error message may not match the specifications. However, the logic that invokes the edit should match.

**TECHNICAL NOTES**

When display instructions, routing, background field assignments involve complex database logic, the wording of the “condition” in the specifications is simplified and is then further defined by a Technical Note. The Technical Note defines in programming logic the database conditions that must be true to satisfy the condition specified.

Technical notes are stored in 2 fields in SpecWriter:

**Technical Note Name + Definition**

Technical notes are global and can be linked to multiple questions and route boxes.

**DESIGN NOTES**

Design notes are used to further define the intent of Question/Box.

Design notes often specify if Question/Box is the beginning or ending point in a loop.

Design notes often specify at what point another section jumps into the current section, or exits the current section.

## **SPECIFICATION ITEMS FOR ROUTE BOXES**

<b>Item Tag</b>	Route Box Number Matches Cheshire Route Box Name
<b>Route Box Instructions</b>	Instructions that direct complex routing. Instructions are specified with IF, ELSE structure. Each condition specified is separated by a line space and ends with Routing instructions.  <b>EX:</b> IF CONDITION IS TRUE, GO TO ST5 – MSNCLNUM.  ELSE GO TO ST6 – INSCLNUM.  May include Background Variable assignment instructions/calculations.

## **BACKGROUND VARIABLE ASSIGNMENTS**

Background Variable Assignments are stored in multiple fields in SpecWriter:

### **Assignment Summary Field Name + Assignment Instructions**

#### **Assignment Summary:**

The Assignment Summary is displayed first in the Full Specification Report.

It is used to:

Specify fields populated in Roster Pop-Up windows.

Specify when database records are generated.

Specify additional conditions for setting background fields.

Summarize why background fields are set.

#### **Field Name + Assignment Instructions:**

Blaise/SQL Field name + instructions.

## **TECHNICAL NOTES**

When routing or background variable assignments involve complex database logic, the wording of the “condition” in the Route Box is simplified and is then further defined by a Technical Note. The Technical Note defines in programming logic the database conditions that must be true to satisfy the condition specified.

Technical notes are stored in 2 fields in SpecWriter:

### **Technical Note Name + Definition**

Technical notes are global and can be linked to multiple questions and route boxes.

### **DESIGN NOTES**

Design notes are used to further define the intent of Route Box.

Design notes often specify if Question/Box is the beginning or ending point in a loop.

Design notes often specify at what point another section jumps into the current section, or exits the current section.

## DISPLAY INSTRUCTIONS

A Display Fill is specified in Text fields in parenthesis and lists display alternatives separated by “/”. Brackets are also used when including nested display fills.

[(Have you/Has (SP)] seen a Medical Doctor since (REFERENCE DATE)?

Display Fill instructions are specified with “IF, ELSE” structure.

Typically, the display fill instruction may reference a previous input field (or fields). The input field will be referenced in the following format:

If OM6 - ORTHTYPE=1/Braces, 2/Supports, 3/Shoes, or 7/Stockings, display "them". Else display "it".

If the display fill instruction involves a field that cannot easily be linked back to one question, the display fill instruction may reference the database table name. Typically, the Cheshire Segment.FieldName will be used.

If EVNT.ORTHTYPE= 91/Other, display other specify text, EVNT.EVOSTEXT.  
Else display EVNT.ORTHTYPE response.

Display Person's first name, ROST.ROSTFNAM.

If the display fill instruction involves multiple fields and the logic is too complex to describe in a simple “IF” condition, the specification is simplified by using a generic statement that is further defined by a Technical Note. See Technical Notes.

**EX:**

If SP is the respondent, display "have".

Else display "has".

**Technical Note:**

**Respondent**

A Proxy interview (respondent is a proxy) =

MRES.SPPROXY=2/Proxy.

MRES.RROSTNUM=ROST.ROSTNUM of respondent.

An SP interview (respondent is the SP) =

MRES.SPPROXY=1/SP.

MRES.RROSTNUM='01'.

**EX:**

If event entered as a Repeat Visit, display number of visits, EVNT.RVTIMES.

Else do not display.

**Technical Note:**

**RepeatVisit**

If event entered as a Repeat Visit =

EVNT.VISTTYPE=2/RepeatVisit.

**CODE ALL conditions:**

Because CODE ALL fields can include multiple responses, CODE ALL fields are referenced using “includes”:

**EX:**

If response to OP3 - OPDREAS includes 1/MedCondNamed or 6/Surgery, then display ...

**ALWAYS DISPLAY PARENTHESIS:**

When question text is in parentheses and is not a display fill, include a display instruction to always display text in parenthesis:

**EX:**

Always display "(may)" and "(health maintenance organizations)" in second sentence in parenthesis.

**EX:**

Always display “(s)” in 3<sup>rd</sup> sentence in parenthesis.

### **EMBEDDED INTERVIEWER INSTRUCTIONS IN QUESTION TEXT:**

Interviewer Instructions embedded in Question text appear in ALL CAPS and is enclosed in brackets.

Interviewer Instructions embedded in Question text should always be displayed in brackets as specified.

1) One type of Interviewer instruction embedded in the Question text is an instruction to read a list of items in a report.

**EX:**

[READ PLAN NAMES BELOW]

[IF RESPONDENT HAS BOTTLE, ASK:]

The default display instruction is to display this text in the same blue font used for interviewer instructions.

2) A second type of Interviewer Instruction embedded in the Question text is an instruction to read additional Question text to the respondent.

**EX:**

[ PROBE: Addition text that can be read.]

[ EXPLAIN IF NECESSARY: Additional text can be read.]

The default display instruction is:

Display “Interviewer Instruction” in the same blue font used by interviewer instructions.

Display a “single space” between Open Bracket and “Interviewer Instruction”.

Display a “colon” after “Interviewer Instruction”.

Display a double space” between “colon” and additional Question Text.

Always display brackets that enclose the Interviewer Instruction and additional text.

### **DISPLAY INSTRUCTIONS IN LOOP:**

When specifying text at a question in a loop, where the question will be asked for multiple items, include instruction to “display name of item” currently being asked about.

**EX:**

Display the event date for the orthopedic item being asked about.

### **HIGHLIGHTING TEXT IN QUESTION:**

When text on the screen should be highlighted, it will be specified as:

**EX:**

Display “not” in BOLD.

**BOLD definition:** Display text in larger font size.

### **“DO NOT DISPLAY” DISPLAY FILLS:**

If a display fill is specified with the option of “Else do not display.”, programming should correct spacing in text:

**EX:**

If PM has already been entered, display “any other”.

Else do not display.

- 1) Correct any “double spacing” in Question text if display fill is not displayed.
- 2) Correct extra space in front of punctuation when display fill appears at the end of a sentence and display fill is not displayed.

## **RESPONSE ALTERNATIVES**

Question response options include the following types of fields:

### **Enumerated field**

Interviewer selects from predetermined list of response alternatives.

EX: YES, NO

Response alternatives displayed with radio button = single select response.

Response alternatives displayed with check box = multi select response.

Response alternative text displayed in ALL CAPS is not read to respondent.

Response alternative text displayed in Upper/Lower case is read to the respondent.

### **String field**

Alphanumeric response.

Field defined by length of field.

Entry field displayed as a single line equal to the length specified.

### **Integer field**

Numeric response, no decimals.

Field defined by minimum and maximum value allowed.

Entry field displayed is large enough to hold maximum value.

Can specify MASK to automatically display "\$", "%", "." as part of input field display.

### **Real field**

Numeric response, collect decimals.

Field defined by minimum and maximum value allowed.

Entry field displayed is large enough to hold maximum value.

Can specify MASK to automatically display "\$", "%", "." as part of input field display.

### **Open field**

Alphanumeric response.

Field defined by length of field.

Entry field displayed as an OPEN text box that allows text wrap.

Implemented for Cheshire Verbatim Text questions.

### **Roster**

List of related items displayed in table format, with multiple columns and rows.

Each row is a unique item.

Each column displays common data for each row.

Response is entered by selecting a row.

Input field = string field.

String field holds "pointers" to items selected.

Roster can be specified to add, delete and edit rows.



## Grid

Used to collect details for multiple items in a list.

EX: Household grid collects details about all persons already identified in the household.

Unlike a Roster, the Grid is displayed with a fixed number of rows. WVS does not support adding or deleting rows in a grid. Therefore, what will be displayed in a grid is predetermined, often through a roster, and then loaded into the grid.

Typically, the first column in a grid is specified as “display only” and contains identifying information for each row. EX: Household person name.

The remaining columns contain input fields for specific questions. EX: Column 2 = Person’s Gender, Column 3 = Person’s Date of Birth.

Unlike a roster, the responses to these questions are entered directly into a grid cell.

Each cell is specified as a unique input field **that can be an enumerated, string, real, or integer input field.**

The interview can navigate between rows and columns in the grid.

## No Entry field

A NO ENTRY screen is used to present information to the respondent when a response is not required.

Input field is specified as an **enumerated field.**

Default Field Type Name = TContinueEmpty.

Although an input field is specified, it is used for programming purposes only and is not displayed on the screen. The interviewer can simply select the appropriate system function to move to the next page.

## **EDIT CHECKS**

Edit checks are specified to check inconsistencies in data.

### **Edit Structure:**

*“Logical condition that should be true.”*

If not true, display message “*ERROR MESSAGE*”.

### **EX:**

MDVLHRS = 1-16.

If not true, display message “ERROR MESSAGE IS IN ALL CAPS”.

If the input value(s) trigger the logical condition as “not true”, a separate EDIT window will be displayed that displays the error message and a list of fields involved in the edit check. The interviewer can then select a field from the list which will allow the interviewer to update the input value and correct the error.

The default is to display the fields involved in the edit check in the order the fields are collected in the interview. In the example above, only MDVLHRS will be displayed.

However, **INVOLVES** can be used to override the default display.

### **EX:**

MDVLHRS = 1-16.

If not true, display message “ERROR MESSAGE IS IN ALL CAPS”.

INVOLVES MDVUNIT, MDVLHRS, MDVLMIN.

In this example, MDVUNIT, MDVLHRS, MDVLMIN will be displayed.

### **Testing edits under certain conditions:**

Use IF THEN statement to specify an edit under certain conditions.

### **EX:**

If SPINSTMM=2/February, then SPINSTDD=1-28.

If not true, display message “ERROR MESSAGE IS IN ALL CAPS”.

In this example, the edit is tested only when SPINSTMM=2/February.

### **Using IF THEN, ELSE statements:**

If multiple values need to be tested for the same field, use the IF THEN, ELSE structure:

### **EX:**

If RXDEUNIT=1/PerYear, then RXDEAMT = \$25.00 to \$1,500.00.

Else if RXDEUNIT=2/Quarterly, then RXDEAMT = \$20.00 to \$250.00.

If not true, display “ERROR MESSAGE IS IN ALL CAPS”

**Soft Edits:**

If specifying a soft edit, you can use “should” to make it easier for the programmer to read:

**EX:**

RXDEUNIT should ^= 91/Other.

If not true, display “ERROR MESSAGE IS IN ALL CAPS”

**Specifying date ranges:**

Specify using “on or between” instead of “>=” and “<=”.

**EX:**

SPINSTYY should be on or between (current year – 108) and (current year).

(current year – 108) is read as “Current Year minus 108”.

(current year – 108) and (current year) should be treated as a calculated “fill”.

**Utilizing fills:**

There are times when “Display fills” or “Calculated fills” are used in edits.

**EX:**

Admission date, EVBEGMM/EVBEGDD/EVBEGYY, must be on or between (REFERENCE DATE) and (TODAY/DATE OF DEATH/DATE OF INSTITUTIONALIZATION).

If not true, display message "INVALID DATE. ADMISSION DATE MUST BE ON OR BETWEEN (REFERENCE DATE) AND (TODAY/DATE OF DEATH/DATE OF INSTITUTIONALIZATION)".  
INVOLVES EVBEGMM, EVBEGDD, EVBEGYY.

(REFERENCE DATE) and (TODAY/DATE OF DEATH/DATE OF INSTITUTIONALIZATION) should be substituted with actual dates specified in the global display fills. Any display fill used that is not defined in the global display fills needs to be specified directly under the edit specification using the same standards as display instructions.

- Soft edits are invoked before hard edits.
- Multiple soft and hard edits can be specified.
- Edits are invoked in the order they are specified.
- Error message is specified by design.
- Soft edits are not currently supported by WVS in Rosters.

## **ROSTERS**

In the Community questionnaire, we maintain dynamic lists of related data items.

ROSTER	CHESHIRE SEGMENT	DESCRIPTION	ROSTER POP-UP NAME *Specified in WVS SpecWriter Report under "Roster Name".	POP-UP Description
PERSON ROSTER	ROST	Displays person information for all persons discussed in interview. Ex: Proxy, contacts, helpers, household members, main insured persons, home owners, etc. Displays person's first name, last name, and relationship to SP.	PERSON ROSTER	Allows Persons to be added.
EVENT ROSTERS	EVNT	Displays single date events, including Repeat Visit events. Displays event begin month, day, year, Repeat Visit flag, Number of visits for repeat visit events.	EVENT ROSTER BEGIN DATE RV	Allows events to be added with Repeat Visit.
		Displays single date events, disallows Repeat Visit events. Displays event begin month, day, and year.	EVENT ROSTER BEGIN DATE	Allows events to be added - no Repeat Visit.
		Displays names of medicines prescribed to SP.	PRESCRIPTION MEDICINE ROSTER	Allows PM events to be added.
PROVIDER ROSTER	PROV	Displays names of providers SP visited/went to see.	PROVIDER ROSTER	Allows Providers to be added.
HOUSEHOLD ROSTER	ENUM, ROST	Displays names of persons living in household. Utilizes the Person Roster Pop-Up Window/ROST for adding new household members.	HOUSEHOLD ROSTER	Allows Person in the Household to be added.
PLAN ROSTER	PLAN, PLRO	Displays health insurance plans SP is/was enrolled in. Displays plan name and status.	PLAN ROSTER	Allows Plans to be added.
STATEMENT EVENT ROSTERS	EVNT	Displays events collected in ST series. Displays event type, begin date, end dates.	STATEMENT EVENT ROSTER	Allows events to be added only.
			STATEMENT EVENT EDIT ROSTER	Allows events to be edited only. Also called in UTS.
			STATEMENT OM EDIT ROSTER	Allows OM events to be edited only. Also called in UTS.
SOURCE OF PAYMENT ROSTERS	TSOP	Displays eligible Sources of Payments in ST series. Displays name of SOP.	SOURCE OF PAYMENT ROSTER	Allows SOP to be added only.
STATEMENT CHARGE BUNDLE ROSTERS	COST	Displays charge bundles entered in ST series. Displays type of statement, claim control numbers.	STATEMENT CHARGE BUNDLE ROSTER	Allows Charge Bundles to be added only.
UTILIZATOIN SUMMARY ROSTERS	EVNT	Displays events collected in previous round.	STATEMENT EVENT EDIT ROSTER	Allows events to be edited only.
			STATEMENT OM EDIT ROSTER	Allows OM events to be edited only.

**EX:**

Lists of persons, health insurance plans, medical visits, etc.

A Roster is a display of related data items in a table format with multiple rows and columns.

Each row in the roster represents a unique item in the list.

**EX:**

One row per Health Insurance plan reported.

Each column in the roster displays common data.

**EX:**

Column 1 = Plan Name, Column 2 = Plan type, Column 3 = Plan Status.

A Roster can be used to collect new items or as a method for linking an item to a particular question.

When responding to a roster question, the interviewer must enter the response by selecting the appropriate row or rows that apply.

Rosters can be specified to allow adding, editing or deleting rows.

The Roster is display only. The interviewer cannot navigate between rows and columns.

When adding or editing a row, the interviewer selects a function that invokes a separate Roster Pop-Up Window. Item details are then collected in the Roster Pop-Up Window.

Data entered/updated in Roster Pop-Up Window is automatically reflected in the roster display once Roster Pop-Up Window is exited.

Typically, when an item is added, it is automatically displayed as “selected” once the roster is refreshed.

When a roster is first displayed, existing items can be loaded into the roster or the roster can be empty. Roster display instructions specify which items are loaded in roster and how they will be displayed.

Items can be specified as a protected row, disallowing it to be selected as a response to the question.

No items in a roster should be displayed as a pre-selected row unless otherwise specified by design.

**Example of Roster instructions:**

Display all orthopedic items that have been reported in the current round where:

EVNT.EVNTDFLG^=1/Yes, and

EVNT.EVNTRNDC = current round, and

EVNT.EVNTTYPE = 'OM', and

EVNT.OMETYPE = 3/Orthopedic.

Display all loaded events as protected rows.

Display in order of entry.

#### Example of Roster display:

COL #	HEADER	INSTRUCTIONS
1	First Name	Display ROST.ROSTFNAM.
2	Last Name	Display ROST.ROSTLNAM.
3	Relationship to SP	Display relationship: If ROST.ROSTREL=91/OtherRelative or 92/OtherNon-Relative, display ROST.ROSTREOS. Else display ROST.ROSTREL relationship code.
4	In Household	If there is a current round ENUM for this person and ENUM.NOTHHRN=empty, display "YES". Else if person just added, display "YES". Else do not display.

#### Roster Column Headers

Currently WVS only supports UPPER CASE column headers. If column headers are specified in Upper Lower case, the column headers should be implemented as UPPER CASE in Roster display.

#### ROSTER vs ROSTER POP-UP WINDOW.

It is important to differentiate between a Roster and a Roster Pop-Up Window.

**ROSTER** References a Roster Question display. Rosters are often categorized by the type of data being displayed:

##### EX:

Event Roster  
Plan Roster  
Person Roster  
Household Roster  
Person Roster

The Roster Category does not describe what columns or rows are displayed; the columns and rows can be customized at each instance of the Roster Question display. (In the future, we may standardize and label each unique Roster Question display).

#### ROSTER POP-UP WINDOW

References a separate pop-up window used for adding/editing items in a roster. The Pop-Up Window specification is a separate specification report and is identified by a Roster Pop-Up Window name:

##### EX:

Event Roster Begin Date  
Event Roster Begin Date RV  
Plan Roster  
Person Roster

The Roster Pop-Up window specification describes what fields are collected in the pop-up window, the pop-up window layout and text, the route between multiple fields, as well as the fields set in the background. A single Roster Pop-Up window may be linked to multiple Roster Question displays.

**EX:**

The Event Roster has 2 types of Pop-Up Roster Specifications:

- Event Roster Begin Date Pop-Up
- Event Roster Begin Date RV Pop-UP
- Prescription Medicine Pop-Up
- Statement Event Edit Pop-Up
- Statement Event Pop-Up

**Roster Name:** Specifies the name of the Roster Pop-Up Window Specifications linked to the Roster Question. (Should be labeled “Roster Pop-Up Window Name”).

## **SINGLE ITEM VS MULTI ITEM SELECT ROSTER**

### **Roster Type:**

A roster is specified as a Single Item Select or Multi Item Select Roster.

Single Item Select =                      When exiting Roster, only one item may be selected.

Multi Item Select =                      When exiting Roster, more than one item may be selected.  
Although Multi Item Select implies unlimited number of items may be selected, there is a maximum number of total records allowed in the roster that is applied to each roster based on project needs. \*This number has not yet been specified. Design is currently accepting the maximum number implemented in the current instrument release. This is a future design issue.

## **ADDING BUT NOT SELECTED**

The ADDED BUT NOT SELECTION function has been added to the WVS system functionality. If the interviewer adds and unselects items in a Roster, WVS will display a warning message when pressing NEXT PAGE to exit Roster. The WVS system message will be displayed for each item ADDED BUT NOT SELECTED and will prompt the interviewer to either DELETE the item or SELECT the item. The DELETE option will be enabled at all Rosters for Items just added. More details to be included in General Specifications as we gain experience with this feature.



## **ROUTE BOXES**

Route Boxes are identified by an Item Tag that matches the Cheshire Box Name.

Route boxes are used to direct complex routing in the section flow. They may include calculations, background variable specifications.

A Route Box specifies a series of conditions, with each condition (or set of conditions) specified with a route instruction. The conditions are specified using an “IF, ELSE ” structure.

**EX:**

IF MC2 – WHATWRNG = 1/EnrolledNewPlan, GO TO MC5 - PLAN\_MHMOMCA.

ELSE GO TO HIMC16 - MHMOMORE.

References to input fields will be specified in the following format:

IF *ITEM TAG – FIELD NAME* = *NUMERIC VALUE/LABEL*, GO TO *ITEM TAG – FIELD NAME*.

**EX:**

IF MC2 – WHATWRNG = 1/EnrolledNewPlan, GO TO MC5 - PLAN\_MHMOMCA.

CODE ALL fields hold all possible responses in one field. Multiple response are referenced using “includes”.

**EX:**

IF RESPONSE TO AC9 – OPDREAS INCLUDES 1/MedCondNamed or 6/Surgery, GO TO....

Complex logic may be specified using a generic statement that is further defined by Technical Note(s):

**EX:**

IF (SP IS IN THE SUPPLEMENTAL, ORD OR DUAL ELIGIBLE SAMPLE)  
AND (SP HAS A LOADED CMS MEDICARE MANAGED CARE PLAN),  
GO TO MC1 - LOADCORR.

ELSE IF (SP IS NOT IN THE SUPPLEMENTAL, ORD OR DUAL  
ELIGIBLE SAMPLE) AND (SP HAS A MEDICARE MANAGED CARE  
PLAN THAT WAS "CURRENT" AT THE TIME OF THE PREVIOUS  
ROUND INTERVIEW), GO TO HIMC1A – MHMOSAME.

ELSE GO TO HIMC1 - MHMOCOV.

### **Technical Notes:**

**SuppORD**

Supplemental, ORD, or Dual Eligible Sample SP =

MRES.INTTYPE=3/NewFromSupplement.

**DeletedPlans**

Deleted plans that are not valid for displays or Checks =  
Any PLAN where (PLAN.PLANDFLG=1/Yes or PLAN.MHMODFLG=1/Yes or PLAN.PLANHIDE=1/Yes or PLAN.LOSEPLFG^=EMPTY) is a deleted PLAN.

**MHMOCurrPrevRnd**

Medicare Managed Care plan "current" at time of previous round interview=  
MRES.INTTYPE^=2,5,6 and there is (a PLAN where PLAN.PLANTYPE=5/MHMO and a PLRO where (PLROPLAN=this PLAN.PLANNUM and PLRO.PLRORND=previous round and PLRO.COVCURNT=1/Yes)).

**LoadedMHMO**

Loaded CMS Medicare Managed Care plan =  
There is a PLAN where PLAN.PLANTYPE=5/MHMO and PLAN.MHMOLRND = current round.  
Plan name = PLAN.PLNAME.

A Route Box can specify one level of nested conditions:

**EX:**

IF FALL ROUND THEN

IF SP NEVER ENROLLED IN A MEDICARE MANAGED CARE PLAN  
AND SP IS ALIVE, GO TO HIMC1INT - MHMOINT.

ELSE GO TO BOX HIMC4.

ELSE GO TO BOX HI1.

**Specifying Loops using Route Boxes:**

Route Boxes are used to specify loops in questionnaire. In Cheshire, loops were often specified with one box, including wording such as:

**EX:**

CYCLE THROUGH MP7 TO MP9 FOR EACH MP EVENT ENTERED AT MP6 THAT WAS ENTERED AS A REPEAT VISIT. WHEN COMPLETE, GO TO MP10.

The convention in WVS will be to specify a "Begin Loop" and "End Loop" box.

**EX: MP6** (Roster – collect multiple events)

**BOX 1:** <begin loop>  
IF AT LEAST ONE EVENT ENTERED AT MP6 WAS ENTERED AS A REPEAT VISIT, GO TO MP7 - FIELD NAME.

ELSE GO TO MP10 – FIELD NAME.

**MP7**  
**MP8**  
**MP9**

**BOX 2:** <end loop>  
IF ANOTHER EVENT ENTERED AT MP6 AS A REPEAT VISIT, GO TO MP7 - FIELD NAME.

ELSE GO TO MP10 – FIELD NAME.

**MP10**

If the Loop is based solely on the existence of a list, e.g. Roster, a “Begin Loop” box may not be needed.

**EX: MP6** (Roster – collect multiple events)  
**MP7**

**MP8**  
**MP9**

**BOX 2:** <end loop>  
IF ANOTHER EVENT ENTERED AT MP6, GO TO MP7 - FIELD NAME.

ELSE GO TO MP10 – FIELD NAME.

**MP10**

### **Complex End Loops:**

Sometimes there is one Route Box that handles the “End Loop” for multiple loops, a series of nested conditions may be specified:

**EX:**  
IF REVIEWING PRIVATE PLANS THAT THE SP WAS COVERED BY AT THE TIME OF THE PREVIOUS ROUND INTERVIEW THEN

IF SP COVERED BY ANOTHER PRIVATE PLAN AT THE TIME OF THE PREVIOUS ROUND INTERVIEW, GO TO HI21 – COVTIME.

ELSE GO TO HI17 - PRVCOVER.

ELSE IF REVIEWING PRIVATE PLANS THAT WERE REPORTED NEW OR RESTARTED IN THE CURRENT ROUND THEN

IF ANOTHER PRIVATE PLAN REPORTED NEW OR RESTARTED IN THE CURRENT ROUND, GO TO HI21 – COVTIME.

ELSE GO TO BOX HI19A.

**End Loops Route Box used with Instance Navigator:**

When Instance Navigator is invoked, there will be an End Loop Route Box specified that directs route back to Instance Navigator screen for next item in List.

**EX:**

**BOX DU4**

IF ANOTHER DENTAL VISIT SELECTED AT DU6, GO TO DU6\_IN - NAVIGATOR ( DUQ1145 ).

ELSE GO TO DU14 - DUMORE ( DUQ1330 ).

It will be important to specify that the ELSE CONDITION is not invoked until CONTINUE INTERVIEW is selected at Instance Navigator screen. A Design Note should be included:

**EX:**

**NOTE ON ELSE CONDITION:**

Once all items at DU6\_IN Instance Navigator screen are DONE, routing will return to DU6\_IN until Interviewer selects CONTINUE INTERVIEW. When DU6\_IN CONTINUE INTERVIEW is selected, BOX DU4 specifies routing to the next item.

## **BACKGROUND VARIABLE ASSIGNMENTS**

### **Example of Background Variable Assignments:**

For Event(s) added, see Event Roster Begin Date specifications for pop-up window programming instructions.

Variables populated in Event Roster Begin Date:

EVNT.EVNTNUM Event number

EVNT.EVNTRNDC Round

EVNT.EVBEGMM Event Month

EVNT.EVBEGDD Event Day

EVNT.EVBEGYY Event Year

BASE.LASTEVT Highest EVNT.EVNTNUM.

For each event added, set additional EVNT fields as instructed below.

<b>EVNTTYPE</b>	EVNT.EVNTTYPE = OM.
<b>OMETYPE</b>	EVNT.OMETYPE = 1/EyeGlasses.
<b>STOMTYPE</b>	EVNT.STOMTYPE = 1/EyeGlasses.
<b>EVNTPROV</b>	EVNT.EVNTPROV = 01.

The following list the rules for specifying Background Variables.

Background Variable specifications are used to specify when a field is assigned a value “behind” a Question or Route Box. The field specified will be the name of the WVS SQL field, but should also be the same as the Blaise input field. In addition, the Cheshire Segment name was included to help locate field/variable in Cheshire. Typically, the corresponding WVS SQL table name will be Tbl + Cheshire Segment name.

Specifications are stored in SpecWriter in the following fields:

### **Assignment Summary**

Summarizes what fields are collected in corresponding Roster Pop-Up Window.

Summarizes fields that will be assigned a value “behind” Question or Route Box.

### **Variable + Assignment Instructions**

Blaise/SQL Field name + what value is assigned to field.

### **EX:**

The question displayed at PM1A is repeated in other utilization sections. Only show this probe for prescription medicine bottles once during the current round interview.

If PM1A - PMAPMMEDS is asked, set flag as instructed below:

<b>GETPMMEDS</b>	Set GETPMMEDS=1/Yes.
------------------	----------------------

### **Specifying Background Variables for Rosters:**

Specify the following in the Assignment Summary:

- 1) Reference Pop-Up window specifications.
- 2) List variables populated in Pop-Up window.
- 3) List all records that need to be generated, include fields that make up the record's KEY. This is typically specified using Cheshire Segment/Key notation, however, can be applied to the WVS SQL table key.
- 4) List instructions for other fields that are assigned a value "behind" Roster.

### **Specify the following in the Variable + Assignment Instructions:**

- 1) Each field that is assigned a value "behind" Roster.

### **EX:**

#### **Assignment Summary:**

If adding a new condition, see Condition Roster Pop-Up window specifications.

Variables populated at the Condition Roster:

COND.CONDNUM

COND.CONDRNDC

COND.CONDTION

All Conditions selected at IP11 should be linked to the Inpatient Stay being asked about. Link Conditions to IP visit on XCON.

XCON key = XCON.XCONBASE + XCON.XCONEVNT + XCON.XCONBAS2 +  
CON.XCONCOND.

For each condition selected at IP11, create an XCON where XCONEVNT=EVNT.EVNTNUM of Inpatient Stay and XCONCOND = COND.CONDNUM of Condition. XCONBASE and XCONBAS2 both equal the Cheshire BASE.BASEID. Set additional XCON variables as specified below.

#### **Variable + Assignment Instructions:**

<b>XCONEVNT</b>	XCON.XCONEVNT = EVNT.EVNTNUM of Inpatient Stay being asked about.
<b>XCONCOND</b>	XCON.XCONCOND = COND.CONDNUM of Condition selected at IP11.
<b>XCONRNDC</b>	XCON.XCONRNDC = current round.

Q. Which fields from Roster Pop-Up Window should be specified in Question specifications?

- A. Typically all fields that are only associated with the Roster Pop-Up window will be specified in the Roster Pop-Up specifications and will be listed in the Background Variable Assignment Summary.

However, if the field is that that gets assigned is determine by which question is being administered, the background variable should be specified at the Question under Background Variable Assignments.

EX: When an event is added in the Event Roster Pop-Up window, EVNT.EVNTTYPE is usually set based on which instance of the Event roster is being asked (DU, OP, MP, etc.)

## **TECHNICAL NOTES**

In Route Boxes, it is important to maintain a level of readability. When the Route Box instruction is too complex, typically the condition is simplified in the specification and is then further specified by a Technical note. Technical notes provide technical specifications regarding how to implement the Route Box instruction.

Although we have more flexibility in writing display instructions and edit checks, Technical Notes are also used to define logic that is repeated throughout the questionnaire.

### **Example of a simplified condition used in specification:**

If SP covered by a Managed Care Plan anytime during the current round, then...

### **Example of Technical note linked to specification:**

#### **ManagedCarePlan**

SP covered by a Managed Care Plan anytime during the current round =  
There is a PLRO where (PLRO.PLRORND=current round and (PLRO.PPRVHMO=1/Yes or PLRO.COVANYTM=1/Yes or PLRO.MCAIDHMO=1/Yes)).

Technical notes are labeled with a Technical Note Name and are numbered automatically by SpecWriter. The Technical Note Name is an abbreviation of the statement being defined - no spaces are allowed. **The first statement in the technical note should match the condition specified.**

Technical notes reference Cheshire Segment.FieldNames and must be translated to the SQL Table.FieldName by the user. Parenthesis is used to define the order that the conditions should be evaluated.

### **Round specific database records:**

If a technical note references a Cheshire Segment/SQL Table that has round specific records, if the round number is not specified, the default is to reference the current round record.



## **DATA TRANSFORMATION AND FIELD NAMES**

The following list examples of Cheshire Variables that need additional specifications to support data transformation.

Input field names reflect the WVS SQL field name. The default is to match the Cheshire variable name. There are several types of variables where WVS collects data differently from Cheshire. Under these circumstances, the WVS SQL field name may not match the Cheshire variable name. (see further instructions below)

All SQL field names that will be transformed back to Cheshire should not exceed 8 characters to match Cheshire conventions.

### **INSTANCE NAVIGATOR:**

In WVS, we are specifying “NAVIGATOR” as a generic field name on all Instance Navigator screens. This field name will not be used by the WVS system. Therefore, CHESHIRE field name is specified as “N/A”

*Data Transformation:*

*These fields will not be stored in WVS or transformed back to Cheshire database.*

### **CHESHIRE “TEMP” VARIABLES:**

In Cheshire, TEMP fields are not assigned a variable name. In WVS, a new field will be generated.

*Data Transformation:*

*These fields will not be transformed back to Cheshire and can be longer than 8 characters. Typically, the new field name does not match an existing Item Tag.*

### **EVOS records:**

In Cheshire, we stored Other Specify Text Response to Questions in Utilization on an EVNT child segment called EVOS.

*Data Transformation:*

*In WVS, these Other Specify Text Response fields were stored on EVNT and will need to be transformed back to the EVOS table.*

### **COSA records:**

In Cheshire, we stored MSN Claim Control Numbers on COSA.

*Data Transformation:*

*In WVS, these MSN claim control numbers were stored on COST and will need to be transformed back to the COSA table.*

### **CHESHIRE CODE ALL THAT APPLY VARIABLES (CODE ALL):**

In Cheshire, the input to CODE ALL screens is stored in separate variables. Each Cheshire variable is associated with a response to the CODE ALL question and is assigned a value based on the response(s) selected.

In WVS, the CODE ALL response is stored in a single field. A new field will be generated and transformed back to the multiple variables in Cheshire. Typically, the name of the new field is similar to the Cheshire variables associated with the CODE ALL question.

***Data Transformation:***

*If DK or RF is entered as the response, all Cheshire variables associated with this CODE ALL THAT APPLY screen are set to DK or RF to match. This DOES NOT include the OTHER SPECIFY TEXT field. No other response is an option.*

*For each response checked, the corresponding Cheshire variable will be set to 1/Indicated. For each response not checked, the corresponding Cheshire variable will be set to 2/Not Indicated. This DOES NOT include the OTHER SPECIFY TEXT field.*

*If “Other” is selected as a response, the Other Specify text will be stored directly into the existing Cheshire Other Specify field. If the “Other” response is unchecked, the Other Specify Text field should be set to empty.*

**CHESHIRE VERBATIM TEXT VARIABLES:**

In Cheshire, Verbatim Text screens collect each line of verbatim text response in a separate variable. Typically, in the input variables are numbered by input line:

PPRVBUS1, PPRVBUS2, PPRVBUS3.

In WVS, Verbatim Text is collected and stored in a single field. A new field will be generated and transformed back to the multiple variables in Cheshire. The length of the new field should be set to the product of the Cheshire variable length and the number of lines collected in Cheshire.

The new field name in WVS is named using the following naming convention:

Cheshire variable name minus line #.

EX:

WVS SQL field name	=	PREVSAT
Cheshire variable names	=	PREVSAT1
		PREVSAT2
		REVSAT3

If we cannot use the current naming convention because the “NEW” name already exists, the second convention would be:

Cheshire variable name minus line # + VB.

EX:

WVS SQL field name	=	JOINHMOVB
Cheshire variable names	=	JOINHMO1
		JOINHMO2
		JOINHMO3

***Data Transformation:***

*The single VB field in WVS needs to be transformed back to the multiple variables in Cheshire. Depending on the length and number of Cheshire fields, the WVS field will be parsed to fill the Cheshire variables in the order of the Cheshire Line #'s.*

**OPTIONAL RADIO BUTTONS TO ENHANCE DATA COLLECTION IN WVS:**

In certain scenarios, we store “reserved values” in Cheshire variables to represent a specific response. For example, in Cheshire we have a variable that stores “Number of Years” the SP was in the facility. If the SP was in a facility “Less than a Year”, the interviewer is instructed to enter “96”. This type of screen has been designed in WVS with an option radio button to improve data collection:

Number of Years: \_\_\_\_\_

☐ Less Than One Year.

In this example, the original Cheshire field, HMONUMYR, is collected on the first line followed by the new field, HMONUM96, on the second line. The new field is an enumerated field with only one response alternative.

Since only one input field should be entered, both fields will be specified with EMPTY attribute. A Hard Edit will be necessary to ensure that only one field is entered.

***Data Transformation:***

*During data transformation, the additional radio button field needs to be transformed back to Field name and value specified under the WVS SPEC Cheshire name.*

***EX:***

Field Name: HMONUM96

Cheshire Name: NONE

*Replaces the following Cheshire Value:*

*If HMONUM96=1/LessThanYear, set*

*ACCS.HMONUMYR=96.*

**DATA TRANSFORMATION REVIEW:**

See the following files

**MCBS WVS Data Transformation Special Fields.xls**

This document lists all of the special data transformation fields that need to be tested to make sure the SQL field is transformed correctly to Cheshire. Any fields not transformed correctly will need to be updated in data editing.

**MCBS WVS Pilot 1 Database Review EMPTY fields.xls**

This document summaries fields that either did not exist in SQL or were set incorrectly to NULL. There fields will need to be reviewed and possibly corrected in data editing. Pilot 1 cleanup comments are included in this document.

### **NEW CHESHIRE VARIABLES TO REPLACE MISSING/OUT OF RANGE VALUES:**

In certain scenarios, WVS does not support data types in Cheshire. Therefore, new fields were created in WVS to store data previously stored in a different field in Cheshire. The new fields were also created in the Cheshire Database. Data Editing will need to be updated to check the new logic for these fields.

Examples of Cheshire data types:

#### **MISS(-5)**

Event Begin Day = miss(-5) represents a Repeat visit event.

Certain Household Enumeration fields = miss(-5) to flag that question will “no longer be asked”.

#### **Value = 95**

Event End Month = 95 represents on ongoing IP stay.

Alteration Event Begin Month = 95 to represent an ongoing alteration.

In order to collect these data values, a new field is generated in the Cheshire database and the WVS SQL database to replace the use of the special Cheshire data types. For existing data in the Cheshire database, the original value will be translated to the new field prior to data transformation.

### **REPEAT VISIT:**

New Cheshire Variable =

<b>EVNT.VISTTYPE</b>	<b>1 SingleVisit</b>
	<b>2 RepeatVisit</b>

EVNT.EVBEGDD = miss(-5) will be translated to:

EVNT.EVBEGDD = miss(-1) and EVNT.VISTTYPE = 2/RepeatVisit.

All other EVNTs where Repeat Visit is a valid option will be translated to:

EVNT.VISTTYPE = 1/SingleVisit.

### **HOUSEHOLD DATA:**

New Cheshire Variable =

<b>ROST.ENSFLAG</b>	<b>1 Yes</b>
---------------------	--------------

ROST.ENSFLAG = 1/Yes indicates that ENS10, which reviews missing household enumeration data from the previous round, has already been asked and should not be asked in future rounds.

All ENUM data that is currently set to miss(-5) will be translated to:

ENUM database field = DK and ROST.ENSFLAG = 1/Yes.

### **ON-GOING IP STAYS:**

New Cheshire Variable =

**EVNT.IPSTATUS**                      **1 StillInHospital**  
   **2 Discharged**

EVNT.EVENDMM = 95 will be translated to:

EVNT.EVENDMM = miss(-1) and EVNT.IPSTATUS = 1/StillInHospital.

IP events that have a discharge date entered will be translated to:

EVNT.IPSTATUS = 2/Discharged

### **ON-GOING ALTERATIONS:**

New Cheshire Variable =

**EVNT.OMSTATUS**                      **1 AlterationNotComplete**  
   **2 AlterationComplete**

EVNT.EVBEGMM = 95 will be translated to:

EVNT.EVBEGMM = miss(-1) and EVNT.OMSTATUS = 1/AlterationNotComplete

Alterations that have a date entered will be translated to:

EVNT.OMSTATUS = 2/AlterationComplete.

Data Transformation:

Since the new field will be generated in the Cheshire database, there are no special rules for data transformation.

### **EVENTS “DELETED” IN CHARGE BUNDLES:**

New Cheshire Variable =

**XCEV.DELLINK**                      **1 Yes**

In the Cheshire CAPI instrument, the interviewer could “delete” events linked to a charge bundle in the COST series. In WVS, we will not actually delete XCEV records as it is done in Cheshire. Instead, a “DELETE” flag will be set on the XCEV record.

Data Transformation:

Since the new field will be generated in the Cheshire database, there are no special rules for data transformation.

**NEW CODE ALL FIELD ADDED FOR CHARGE BUNDLES:**

In Cheshire, we collected the following fields which indicated what types of events were added in ST/NS for the charge bundle:

**COST.INCDATES**

**COST.INCPMS**

**COST.INCOMS**

In WVS, we added one more code for Adding Home Health Events:

**COST.INCHHS**

## **5. COST PAYMENT SUMMARY: SETTING COST.CPREASON**

Prior to cases being fielded, COST.CPREASON is set for charge bundles that are eligible for the current round Cost Payment Summary Section. The Cheshire field COST.CPREASON is set at the home office and then transformed to the SQL field tblCOST.CPREASN.

The following is a list of steps used to identify previous round Charge Bundles that are eligible for the current round Cost Payment Summary section.

**STEP 1:** Only previous round Charge Bundles will be eligible for the current round Cost Payment Summary section if the Charge Bundle meets the following criteria:

- 1) Charge Bundle is from the past 2 rounds.
- 2) Charge Bundle is not flagged as "no longer expecting CPS".

### **STEP 1 TECHNICAL NOTES:**

If (COST.COSTRNDNC = (Current Round -1) or (Current Round – 2)) and (COST.EXCPSFLG ^= 2/NotIndicated), go to **STEP 2**.

ELSE CHARGE BUNDLE IS NOT ELIGIBLE FOR THE CURRENT ROUND COST PAYMENT SUMMARY SECTION.

**STEP 2:** Eliminate previous round Charge Bundles that meet the following criteria:

- 1) Eliminate previous round Charge Bundle if at least one event linked to Charge Bundle is an IU Event, an on-going IP Event, a HP Event, a HF Event.
- 2) Eliminate previous round Charge Bundle if at least one event linked to the Charge Bundle is a PM Event and total number of purchases over the past year = 0 or at least one round purchases = DK or RF.
- 3) Eliminate previous round Charge Bundle if at least one event linked to the Charge Bundle is an OM Ostomy Supplies, Incontinence Supplies, or Bandages and total number of purchases over the past year = 0 or at least one round purchases = DK or RF.
- 4) Eliminate previous round Charge Bundle if at least one event linked to the Charge Bundle is flagged as deleted.

### **STEP 2 TECHNICAL NOTES:**

If (EVNT.EVNTTYPE = 'IU'),

OR

If (EVNT.EVNTTYPE = 'IP' and EVNT.IPSTATUS = 1/StillInHospital).

OR

If (EVNT.EVNTTYPE = 'HF' or 'HP') and there is at least one HERO where  
HERO.HEROEVNT = EVNT.EVNTNUM has (HERO.PROFPROB =  
3/HomeHealthEnteredInError or HERO.FRNDPROB = 3/HomeHealthEnteredInError),

OR

(EVNT.EVNTDFLG = 1/Yes), THEN CHARGE BUNDLE IS NOT ELIGIBLE FOR THE CURRENT  
ROUND COST PAYMENT SUMMARY SECTION.

ELSE GO TO **STEP 3**.

**STEP 3:** Include previous round Charge Bundle if it meets one of the following criteria. Set  
COST.CPREASON based on logic below:

**STEP 3 TECHNICAL NOTES:**

If COST.COSTTYPE = 2/NS and COST.EXMCMAIL = 1/Yes, set COST.CPREASON = 1.

Else if COST.COSTTYPE = 2/NS and COST.EXMCMAIL = 9/FlagCostForCPS, set COST.CPREASON = 8.

Else if COST.COSTTYPE = 2/NS and ((COST.TOTALAMT > 1.00) or (COST.TOTALAMT  
= DK or RF)) and  
(There is no PAYM where PAYM.PAYMCOST = COST.COSTNUM and PAYM.PAYMDFLG  
= empty and PAYM.PAYMAMT > 0, DK or RF), set COST.CPREASON = 2.

Else if COST.COSTTYPE = 1/ST and ((COST.AREMAING > 1.00) or (COST.AREMAING =  
DK or RF)) and  
(COST.MCPAYAMT ^= DK and ^= RF) and (There is no PAYM where PAYM.PAYMCOST  
= COST.COSTNUM and PAYM.PAYMDFLG = empty and PAYM.PAYMAMT > 0, DK or  
RF), set COST.CPREASON = 3.

Else if COST.COSTTYPE = 2/NS and (COST.TOTALAMT ^= DK or ^= RF) and (There is at  
least one PAYM where PAYM.PAYMCOST = COST.COSTNUM and PAYM.PAYMDFLG =  
empty and PAYM.PAYMAMT > 0) and (ALL PAYM.PAYMAMT ^= DK, ^= RF, and ^= 0) and  
((Total of PAYM.PAYMAMT < COST.TOTALAMT - 1)) or (Total of PAYM.PAYMAMT <  
(COST.TOTALAMT \* 0.98))), set COST.CPREASON = 4.

Else if COST.COSTTYPE = 1/ST and (COST.AREMAING ^= DK and ^= RF) and  
(COST.MCPAYAMT ^= DK and ^= RF) and (There is at least one PAYM where  
PAYM.PAYMCOST = COST.COSTNUM and PAYM.PAYMDFLG = empty and  
PAYM.PAYMAMT > 0) and (ALL PAYM.PAYMAMT ^= DK, ^= RF, and ^= 0) and ((Total of  
PAYM.PAYMAMT < (COST.AREMAING - 1)) or (Total of PAMS.PAYMAMT < COST.AREMAING  
\* 0.98))), set COST.CPREASON = 5.

Else if (There is a PAYM where PAYM.PAYMCOST = COST.COSTNUM and  
PAYM.PAYMDLFG = empty and PAYM.PAYMTYPE = 1 and PAYM.PAYMAMT > 5.00 and



there is a CORO where (CORO.CORORND = previous round and CORO.EXPPAYBK = 1)), set COST.CPREASON = 6.

Else if (There is a PAYM where PAYM.PAYMCOST = COST.COSTNUM and PAYM.PAYMDLFG = empty and PAYM.PAYMTYPE = 1 and PAYM.PAYMAMT >5.00 and there is a CORO where (CORO.CORORND = previous round and CORO.EXPPAYBK = DK)), set COST.CPREASON = 7.

### **Differences between Cheshire and WVS CPS:**

The CPS specifications indicate that if at least one event is linked to an NS charge bundle in the current round and EXMCMail ^= 1 or 3, do not ask about charge bundle in CPS. However, if the NS scenario skips EXMCMail, then "EXMCMail = null" and is not evaluated as the same as "EXMCMail ^= 1 or 3". Therefore, even though this event is linked to a current round charge bundle, it is still asked about in CPS.

Examples of where events linked to a CPS charge bundle are still asked about in CPS even though event is linked to current round COST data:

- 1) Any OM rent to buy, that goes through PS and you answer YES, goes to NS in the current round and we collect charges without asking EXMCMail. Thus, these will also come up in CPS.
- 2) There is a box in NS where we skip EXMCMail. For these events, if we skip EXMCMail, if the event is also linked to a CPS charge bundle, we will still ask about the charge bundle in CPS.
- 3) If in CPS, you route to NS and you then link events to the NS charge bundle that are also linked to CPS charge bundles (and haven't been asked yet), these charge bundles will still get asked about in CPS (i.e. because the events are linked to a previous round NS COST).
- 4) In Cheshire, CPS Reason was evaluated and set at the beginning of CPS during the interview. In WVS, CPS Reason is set prior to a case being fielded. The logic for excluding events from coming up in CPS includes excluding on-going IP stays and on-going OM Alterations. In Cheshire, if an on-going IP stay or on-going OM alteration is linked to a CPS charge bundle and if the SP was discharged from the IP stay or the OM Alteration was completed in the current round, the CPS charge bundle was still asked about in CPS. However, since CPS Reason is evaluated prior to the interview in WVS, the same scenario would result in the CPS charge bundle not getting asked about in CPS..

Note:

Keep in mind that in R52, issue #1 and #2 did not apply because we did not have the check for EXMCMail in CPS. Therefore, these charge bundles were not asked about in CPS in R52.

In contrary, in R52, we had charge bundles in CPS that were not asked about because the event (PM, OM rental) was asked about in NS and we answered EXMCMail = 1/Yes. However, without the check for EXMCMail, these charge bundles did not come up in CPS because the events were now linked to a current round COST.

## 5. ROSTER REQUIREMENTS

---

- Roster Page that is displayed (if multiple pages)
  - Moving forward
    - Display 1<sup>st</sup> page
  - Moving backwards
    - Display 1<sup>st</sup> page
  - In Roster, adding a new item
    - Display Page where Item is added
    - Item ADDED should be displayed in specified Display Order.
- Display all Roster rows in One Column
- Display all Columns in Roster per specification
- Display Items (Rows) in Roster per Roster Load specifications
  - Display Rows per specifications
  - Display Rows in Order that matches specs
    - Items should be displayed in specified display order when Roster is displayed moving forward and backwards.
    - If Display Order is not specified, the default is to display roster items in “Order of Entry”, matching the order item is stored on related database table.
    - Items added to be roster should be displayed in display order.
    - Person Roster
      - Proxy Roster displayed alphabetically. Pending spec update?
      - MIP Roster displayed
        - SP
        - Spouse
        - Order of Entry
      - Household Roster displayed
        - SP
        - Proxy
        - Order of Entry
    - Plan Roster
      - Display order of entry.
    - Utilization Event Rosters
      - Display in Order of Entry
    - PM Roster
      - Display Alphabetically. Pending spec update?
    - Provider Roster
      - Order of Entry
    - SOP Roster
      - By SOP Type
    - Statement Event Rosters
      - Order of Entry
  - Protected Rows.

- Protected Rows should be “grayed out”.
  - Display Items as Protected Row per specifications moving forward.
  - Do not apply Protected Row specifications to items JUST ADDED.
- ADD Function
  - Enable ADD Function where specified.
  - Single Select and Multi Select Item Rosters should both allow more than one item to be added. However, Single Select Rosters will only allow 1 item to be selected.
  - ADD should invoke specified Pop-Up Window.
  - Roster Pop-UP
    - Pop-Up window display should match specifications
    - Pop-Up input fields and routing should match specifications
    - Don’t Know and Refuse
      - Entered using Right Mouse Button.
      - Can also enter DK and RF into input field.
      - Design has requested a DON’T KNOW and REFUSE “button” similar to MEPS. DEFER TO PILOT 2.
    - ADD and STAY function
      - Add in Pilot 2.
    - Roster Edits
      - Roster Edits are invoked differently than Specified HARD and SOFT edits.
        - WVS does not support SOFT edits in Roster. Need to identify the location of these and consider redesign.
  - Items ADDED should automatically be selected.
  - Data collected in Pop-Up should automatically be displayed in Roster when leaving Pop-Up window.
  - Items ADDED should automatically be displayed in DISPLAY ORDER specified.
- EDIT Function
  - Enable EDIT Function where specified.
  - EDIT should invoke specified Pop-Up Window.
  - Roster Pop-UP
    - Pop-Up window display should match specifications
    - Pop-Up input fields and routing should match specifications
  - Data updated in Pop-Up should automatically be displayed in Roster when leaving Pop-Up window.
- DELETE Function
  - Not currently specified as an enabled option.
  - Default DELETE function will be enabled on all Rosters where ADD is enabled. Will be used only to delete items JUST ADDED.
- Selection
  - Single Item Select rosters should allow 1 Item to be selected.
    - If Multiple Items ADDED, the last item ADDED should be selected.
    - Allow Items to be unselected by selecting/adding another item.

- Multiple Item Select rosters should allow multiple items to be selected.
  - Allow Items to be unselected.
- If NO SELECTION REQUIRED is specified, allow no selection. Otherwise, always force a selection.
- PRESELECTED ROWS
  - Rows should never appear PRESELECTED moving forward
- Paging
  - Navigating to a different PAGE in the Roster should not change what has been selected on that page.
- Added But Not Selected
  - If you ADD but NOT SELECT an item, a verification screen should appear.
  - Should have option to DELETE item ADDED but not selected.
  - WVS currently does not support once you leave Roster and return using PREVIOUS PAGE. Do we expect a system fix to this?
- Instance Navigator
  - Instance Navigator should be invoked after every Roster specified as a Multi-Item select Roster followed by Roster Details.
    - There may be an exception to this agreed upon by programming and design.
  - Instance Navigator should display all Items selected from the Roster in the same display order specified for the Roster.
  - All Items should display a status of NOT STARTED/IN PROGRESS or DONE.
  - Should be able to administer Roster Details for Items “out of order”.
  - Instance Navigator should only display PREFILLED responses to Roster Details already entered.
    - This will occur when an Item is DONE or IN PROGRESS.
    - All Items NOT STARTED should not have pre-filled responses.
  - Instance Navigator should disallow CONTINUE INTERVIEW until all items are DONE.
  - When completing the last item listed in Instance Navigator, the route should return to the Instance Navigator screen. The interviewer should then be required to select CONTINUE INTERVIEW to continue to next item on route.
  - When using PREVIOUS PAGE after a Roster Detail loop, PREVIOUS PAGE should display the Instance Navigator screen and allow access to the Roster details.
- PREVIOUS PAGE
  - When pressing PREVIOUS PAGE from Roster Details, the Roster Function and Display should match what was just displayed moving forward from Roster:
    - Items selected should still be selected and NOT PROTECTED.
    - The Roster should still recognize Items JUST ADDED.
      - If you unselect an Item that was ADDED earlier, the ADDED BUT NOT SELECTED confirmation screen should still be invoked.
    - If ADD is enabled, should allow items to be added to Roster.
      - Items ADDED should be displayed in Instance Navigator moving forward.
      - If you ADD a new item and unselect it, the ADDED BUT NOT SELECTED confirmation screen should be invoked.
  - If you unselect an Item, it should not appear in Instance Navigator moving forward.