

**MCBS MAIN STUDY**  
**R55 General Specifications for Blaise/WVS**  
Version 7.01.0001  
8/14/2009

**DATES**

TABLE OF CONTENTS

1. Reference Periods
2. Previous Round
3. Date Entry and Display
4. Date Edits
5. Out of Survey Reference Period Flag
6. Date calculations

## **1. REFERENCE PERIODS**

---

Data is typically collected within a specified reference period. The reference periods are determined prior to the case being fielded and the dates are stored in the database.

### **REFERENCE DATE**

#### **MRES.MREFDATE**

Also referred to as the “current round reference date”.

Refers to the SP’s last community interview date or the date SP was discharged from a facility. New round data is collected from the REFERENCE DATE to the date of the current round interview. If the SP has died or has been institutionalized, data will be collected up to the date of death or the date the SP entered the institution.

### **SUMMARY REFERENCE DATE**

#### **MRES.SUMMBDAT**

Refers to the SP’s Reference date (MREFDATE) from the previous round interview. This date is referenced when collecting/Updating data that was collected in the previous round interview (in summary sections). Data updated/collected in summary sections is collected from the SUMMARY REFERENCE DATE to the REFERENCE DATE.

### **SURVEY REFERENCE DATE**

#### **MRES.SRPBDAT**

Refers to the SP’s Reference date (MREFDATE) from 2 rounds previous (approximately a year prior to current interview date). This date is referenced when collecting data from the past year. Typically, when collecting data from the past year, the reference period will be from the SURVEY REFERENCE DATE to the date of the current round interview (or SP’s date of death/SP’s date of institutionalization).

## 2. PREVIOUS ROUND

The specifications often reference data collected in a previous round interview.

The phrase “previous round” is used to reference data collected in the SP’s previous round interview. The primary reason for setting this value is to properly locate data for SP’s who have skipped the previous round. However, if an SP’s previous round interview was conducted in a facility, the previous round number will not be set to the SP’s most recent community interview round. Instead, it will be set to “current round minus 1”.

Previous round number should be evaluated for each interview type as follows:

INTTYPE=1/StandardHadPrev	Current round minus 1
INTTYPE=2/NewFromFacility	No previous round – Set to Current round minus 1
INTTYPE=3/NewFromSupplement	No previous round – Set to Current round minus 1
INTTYPE=4/StandardSkippedPrev	Current round minus 2
INTTYPE=5/LastRndFacSum	No previous round – Set to Current round minus 1
INTTYPE=6/LastRndFacBase	No previous round – Set to Current round minus 1
INTTYPE=7/SupSmp1stTimeUtil	Current round minus 1
INTTYPE=8/ExitInterviewHadPrev	Current round minus 1
INTTYPE=9/ExitInterviewSkippedPrev	Current round minus 2
INTTYPE=10/SupSmp1stTimeUtilSkipped	Current round minus 2

### **Example of specification:**

IF THIS PUBLIC PLAN WAS "CURRENT" AT THE TIME OF THE PREVIOUS ROUND INTERVIEW, GO TO BOX HI11A - (HIQ5890).

ELSE GO TO HI15 - COVBEGMM ( HIQ5820 ).

### **Technical Note:**

This Plan was "current" at the time of the previous round interview = MRES.INTTYPE^=2,5,6 and (there is a PLRO where (PLRO.PLROPLAN=this PLAN.PLANUM and PLRO.PLRORND=previous round and (PLRO.COVTIME=1/WholeTime or PLRO.COVNOW=1/Yes).

If Previous Round has been set to “41” for this SP and there is no PLRO where PLRO.PLRORND = 41, this statement would be evaluated as “False”.

## SP’s LAST TWO COMMUNITY INTERVIEWS

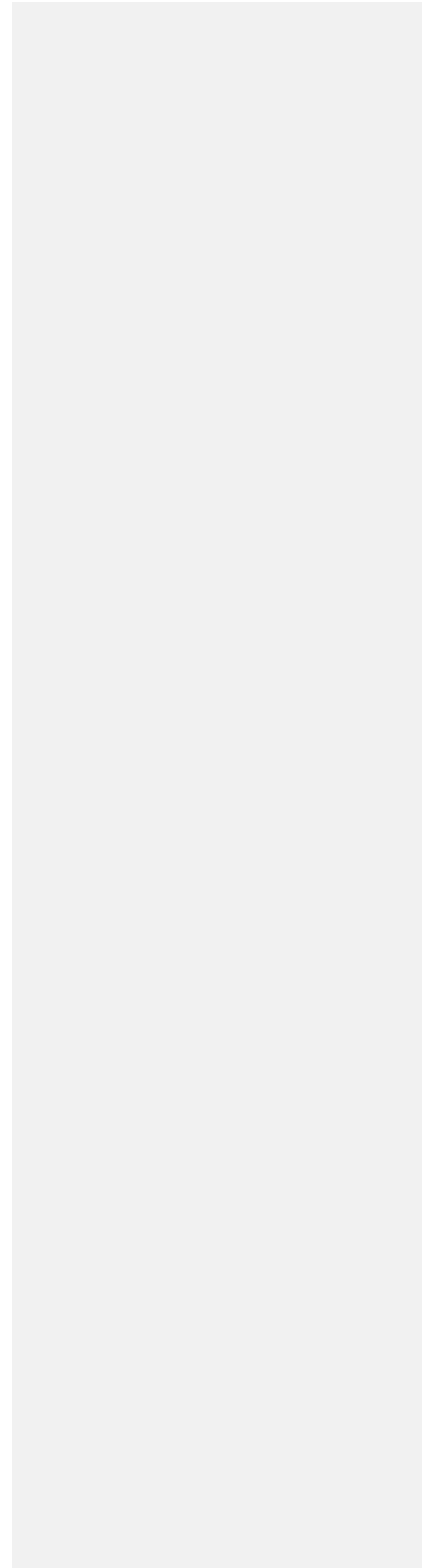
When a case is prepared for the current round, the SP’s most recent community interviews are stored on the current round MRES. The Round numbers stored in these database fields should not be referenced when determining the value of “previous round”. See “Previous Round” above.

**MRES.PREVCOM1**

The Round number of the SP's most recent community interview.

**MRES.PREVCOM2**

The Round number of the SP's 2<sup>nd</sup> most recent community interview.



### 3. DATE ENTRY AND DISPLAY

MCBSWVS, General Date Formatting Requirements

~~(Revised 2/16/06)~~

#### DATE ENTRY:

The following principle should govern **date entry** throughout the questionnaire, unless specific override exception is noted in the questionnaire specifications.

“MM”, “DD”, “[YY]YY” formats:

Entry of dates at most screen types (e.g., single date entry screens, grids and roster pop-up windows), generally:

3) Entry of dates, generally:

Collected as three distinct numeric fields;

example: “10” “31” “04”

“MM”	= Collect 2-digit Month.
“DD”	= Collect 2-digit Day.
“YY”	<u>= Collect 2-digit year.</u>
“YYYY”	<u>= Collect 4-digit year.</u>

If Year field is specified as a 2-digit integer, only 2-digit year is required, however a 4-digit year will be accepted.

If Year field is specified as a 4-digit integer field, only a 4-digit year is acceptable.

\* This is currently not supported by application.

Allow “DK” and “RF” in Month, Day or Year field if specified as Field Attributes.

Purpose: ease of keying by users

- When entering dates, a leading zero on otherwise single digit month and day values is acceptable in operation, but should be stripped away in conversion to other date formats and usages.
- In roster programming, a SQL template is necessary where three numeric field date entry occurs on roster addition pop-up screens, yet date display in the roster window displays date as on field in #2 format. [SQL: user function toMonDDYYYY() for display on roster]

Formatted: Bullets and Numbering

Formatted: Indent: Left: 0"

## DATE DISPLAY:

For the MCBSWVS instrumentation, the following principles should govern **date display and entry** throughout the questionnaire, unless specific override exception is noted in the questionnaire specifications.

Formatted: Font: Bold, Underline

### 1) "Month DD, YYYY" format:

Display of dates as inline text, such as in question text, context headers, report headers and instructions or otherwise read to respondents:

Context Headers, Interviewer Instructions, and Error Messages: Display in UPPER CASE; example: OCTOBER 31, 2004.

All other text: Displayed in Upper/lowercase lettering; example:  
October 31, 2004

Formatted: Indent: Left: 1", First line: 0"

Purpose: ease of reading when embedded in other text

- Do not display leading zeros.
- Display all 2-digit year fields as 4-digit year. The default rule for filling two digits to four digits years is: fill with "20(0)" for range 0 to current year, else fill with "19".
- Date display will include "DK" or "RF" as specified below.

### 2) "Mon DD YYYY" format:

Display of other dates in a table (or list) format, such as rosters or reports on screens (such as headers, columns in rosters or grids, in instructions, etc.):

"Mon" should be displayed in upper/lowercase lettering.

Example: Oct 31 2004

Formatted: Bullets and Numbering

This format will be specified as one distinct column. However, it may implemented as one column as specified or as 3 columns if needed to meet WVS system requirements.

If this format is implemented as 3 columns, column header should be displayed over the 3 columns.

Purpose: ease of reading when displaying date for questionnaire operations or visual date comparison or reference, also more compact and consistent for lists

- Do not display leading zeros.
- Display all 2-digit year fields as 4-digit year. The default rule for filling two digits to four digits years is: fill with "20(0)" for range 0 to current year, else fill with "19".
- Date display will include "DK" or "RF" as specified below.

**“DK” and “RF” DISPLAYS:**

**DON'T KNOW and REFUSE date display:**

If missing Month, Day, or Year:

If variable = RF (Refused), display “RF”.

EX: Mar RF 2006  
March RF, 2006  
RF 2 2006  
RF 2, 2006  
Mar 2 RF  
March 2, RF

If variable = DK (Don't Know), display “DK”.

EX: Mar DK 2006  
March DK, 2006  
DK 2 2006  
DK 2, 2006  
Mar 2 DK  
March 2, DK

**REPEAT VISIT date display:**

If EVNT.VISTTYPE = 2/RepeatVisit, store EVNT.EVBEGMM = DK and display “DK” for Event Begin Month, EVNT.EVBEGMM.

EX: Mar DK 2006  
March DK, 2006.

**Date variables that have been implemented differently in WVS:**

Repeat Visit Dates:

Available option for event types: DU, OP, MP event.

Cheshire: A Repeat visit event is identified by EVNT.EVBEGDD=-5.

WVS: A Repeat visit event is identified by EVNT.VISTTYPE=2/RepeatVisit.

Note: EVNT.EVBEGDD will be automatically set to DK.

On-going IP stays:

Cheshire: An on-going IP stay is identified by EVNT.EVENDMM=95.

WVS: An on-going IP stay is identified by EVNT.IPSTATUS=1/StillInHospital.

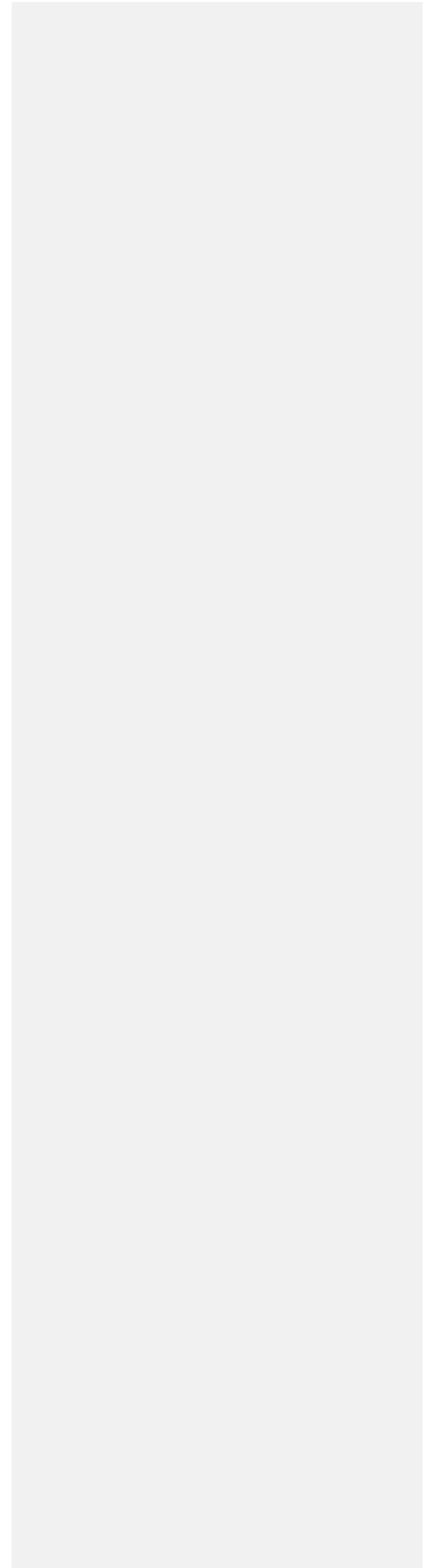
Note: EVNT.EVENDMM/ EVNT.EVENDDD/ EVNT.EVENDYY will be empty.

On-going Alteration:

Cheshire: An on-going OM alteration is identified by EVNT.EVBEGMM=95.

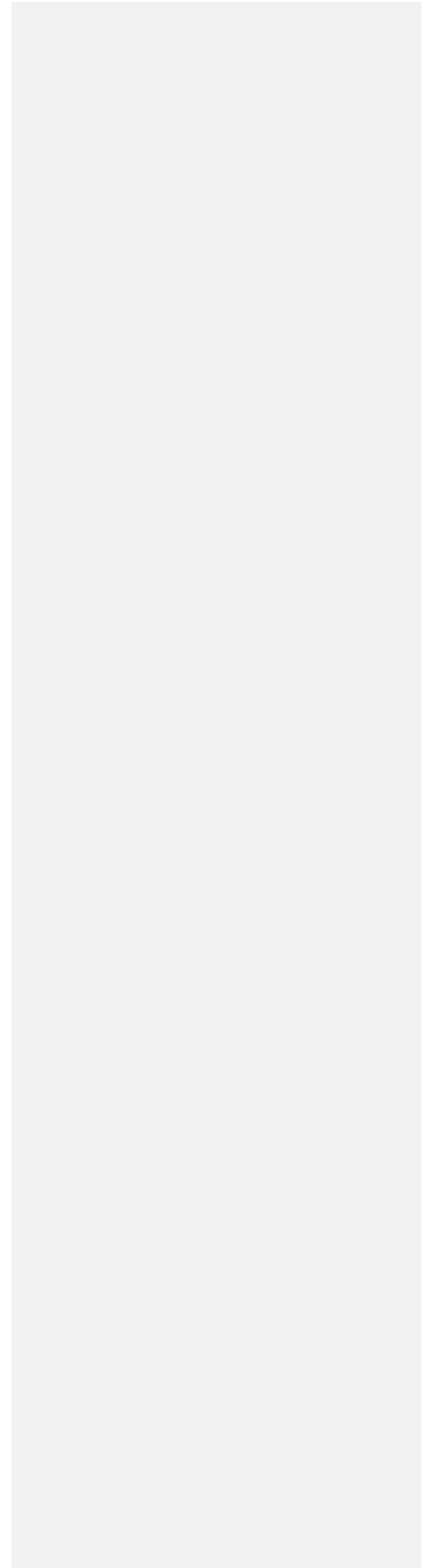
WVS: An on-going OM alteration is identified by EVNT.OMSTATUS =1/AlterationNotComplete.

Note: EVNT.EVBEGMM/  
EVNT.EVBEGDD/EVNT.EVBEGYY will be empty.





|



## MCBSWVS Date Handling

Formatted: Font: Not Bold

### SQL User Functions, Dates

Formatted: Font: Not Bold

Use in SQL for app. date (display) standardization, handles missing values too.

toMM(DateStr)—WVS function, takes date string arg. and strips Mon month from start and replaces with month number

Formatted: Font: Not Bold

toMon(Str)—app. function, converts month number to Mon

Formatted: Indent: Left: 0", First line: 0"

[toMonInt(Int) variation with integer argument]

Formatted: Font: Not Bold

toYYYY(Str)—app. function, converts YY number to YYYY

Formatted: Font: Not Bold

[toYYYYInt(Int) variation with integer argument]

ConvertMiss(Str)—app. function, converts -7, -8 and Ref in any string to RF and DK as output

Formatted: Font: Not Bold

[ConvertMissInt(Int) variation with integer argument]

toMonDDYYYY(MMStr,DDStr,YYStr)—app. function, takes three numeric as string arguments for MM,DD,YY(YY) and produces a standard "Mon DD YYYY" output string

Formatted: Font: Not Bold

### Blaise Procedures, Dates

Formatted: Font: Not Bold

DateFldsToDate(YYStr,MMStr,DDStr,OutStr)—three numeric string date fields to Blaise DateType

DateFldsToText(YYStr,MMStr,DDStr,OutStr)—three numeric string date fields to Mon-DD-YYYY

Formatted: Font: Not Bold

DateNumsToText(YY,MM,DD,OutStr)—three number date fields to Mon-DD-YYYY

Formatted: Font: Not Bold

DateStdToFullText(InStr,OutStr)—converts string Mon-DD-YYYY to "Month, DD, YYYY"

Formatted: Font: Not Bold

DateToText(InDate,OutStr)—converts Blaise DateType to Mon-DD-YYYY string

Formatted: Font: Not Bold

MonToNum(MMStr,MM)—converts Mon or Month string to month number

Formatted: Font: Not Bold

NumToMon(MM,MMStr)—converts month number to Mon

Formatted: Font: Not Bold

NumToYYYY(YY,YYYYStr)—converts (partial) year number to YYYY

Formatted: Font: Not Bold

StdMonToFullMon(InStr,OutStr)—converts Mon string to Month string

Formatted: Font: Not Bold

StrToYYYY(YYStr,YYYYStr)—converts partial year numeric string to four digit year string

Formatted: Font: Not Bold

TextToDate(InStr,OutDate)—converts Mon-DD-YYYY string to Blaise DateType

Formatted: Font: Not Bold

GetMMDYYFromStd(InStr,MMStr,DDStr,YYStr)—converts Mon-DD-YYYY date string to MM, DD, YY individual date numeric strings [note: rename

Formatted: Font: Not Bold

TextToDateFlds(?)

## **4. DATE EDITS**

### **Single Date Verification**

1. Check for digits only, or DK, RF
2. Check for valid date entry
  - a. Month Check

- Month = 1..12, DK, RF
- b. Day Check
  - If Month=2
    - If Year ^= DK, RF
      - If leap year, Day=1..29, DK, RF
      - Else Day=1..28, DK, RF
    - If Year = DK, RF
      - Day = 1..29, DK, RF
  - If Month=4,6,9,11
    - Day = 1..30, DK, RF
  - If Month=1,3,5,7,8,10,12
    - Day=1..31, DK, RF
  - If Month=DK, RF
    - Day=1..31, DK, RF

### Two Date Verification

Two Date Verification is used to determine if a single date is BEFORE, ON, or AFTER another single date. When discussing Two Date Verification, the Date being checked will be referenced as the TARGET DATE. The date the target date is being checked against will be referenced as the REFERENCE DATE.

The following logic should be used when comparing two dates.

#### Evaluate check in the following order:

	<b>Result</b>
If Target Year = DK,RF or Reference Year = DK, RF	CANNOT CHECK
Else if Target Year < Reference Year	BEFORE
Else if Target Year > Reference Year	AFTER
Else Target Year = Reference Year, continue checking Month.	
If Target Month=DK,RF or Reference Month = DK, RF	CANNOT CHECK
Else if Target Month < Reference Month	BEFORE
Else if Target Month > Reference Month	AFTER
Else Target Month = Reference Month, continue checking Day.	
If Target Day = DK, RF or Reference Day = DK, RF	CANNOT CHECK
Else if Target Day < Reference Day	BEFORE
Else if Target Day > Reference Day	AFTER
Else Target Day = Reference day	ON

#### The Two Date Verification results include:

BEFORE	Target Date is before Reference Date
AFTER	Target Date is after Reference Date
ON	Target Date matches reference Date
CANNOT BE CHECKED	Target Date CANNOT BE CHECKED against the Reference Date. If a Soft/Hard edit is specified and the date verification result = CANNOT BE CHECK, the Soft/Hard edit should not be invoked.

### Three Date Verification

Three date verification is used when determining if a single date is BEFORE, AFTER or ON/BETWEEN a specified reference period that includes a BEGIN and END date. When discussing Three Date verification, the Date being checked will be referenced as the TARGET DATE. The specified BEGIN and END dates will be referred to as the REFERENCE PERIOD BEGIN DATE and REFERENCE PERIOD END DATE.

The following logic should be used when comparing two dates.

<b>Evaluate check in the following order:</b>	<b>Result</b>
If Target Year = DK, RF	CANNOT CHECK
Else run Two Date Verification with Target Date and Reference Begin Date	

<b>Two Date Verification with Target Date and Reference Begin Date:</b>	
If Date Verification = BEFORE	BEFORE
Else if Date Verification = ON	ON OR BETWEEN
Else run Two Date Verification with Target Date and Reference End Date	

<b>Date Verification with Target Date and Reference End Date:</b>	
If Date Verification = ON or BEFORE	ON OR BETWEEN
Else if Date Verification = AFTER	AFTER
Else	CANNOT CHECK

**The Three Date Verification results include:**

BEFORE	Target Date is before Reference Period
AFTER	Target Date is after Reference Period
ON OR BETWEEN	Target Date is on or between Reference Period
CANNOT BE CHECKED	Target Date CANNOT BE CHECKED against Reference Period. If a Soft/Hard edit is specified and the Three Date Verification result = CANNOT BE CHECK, the Soft/Hard edit should not be invoked.

**Sample Test Plan for “Target date must be on or between two dates”:**

**BEGIN DATE = 9/14/04**

**END DATE = 5/6/07**

TARGET DATE TESTED	Expect Hard Edit?	TARGET MONTH	TARGET DAY	TARGET YEAR	Can't check DK/RF
10/16/03	YES	Month > Begin Month	Day > Begin Day	Year < Begin Year	
8/16/04	YES	Month < Begin Month	Day > Begin Day	Year = Begin Year	
9/12/04	YES	Month = Begin Month	Day < Begin Day	Year = Begin Year	
2/DK/04	YES	Month < Begin Month	Day = DK or RF	Year = Begin Year	
DK/DK/03	YES	Month = DK or RF	Day = DK or RF	Year < Begin Year	
1/1/DK	NO	Month < Begin Month	Day < Begin Day	Year = DK	X
DK/1/04	NO	Month = DK or RF	Day < Begin Day	Year = Begin Year	X
9/DK/04	NO	Month = Begin Month	Day = DK or RF	Year = Begin Year	X
DK/DK/04	NO	Month = DK or RF	Day = DK or RF	Year = Begin Year	X
5/DK/DK	NO	Month < Begin Month	Day = DK or RF	Year = DK or RF	X
DK/3/DK	NO	Month = DK or RF	Day < Begin Day	Year = DK or RF	X
9/14/04	NO	Month = Begin Month	Day = Begin Day	Year = Begin Year	
10/1/04	NO	Month > Begin Month	Day < Begin Day	Year = Begin Year	
1/1/05	NO	Month < Begin Month	Day < Begin Day	Year > Begin Year Year < End Year	
4/7/07	NO	Month < End Month	Day > End Day	Year = End Year	
6/10/06	NO	Month > End Month	Day > End Day	Year > Begin Year Year < End Year	
7/10/DK	NO	Month > End Month	Day > End Day	Year = DK	X
DK/10/07	NO	Month = DK or RF	Day > End Day	Year = End Year	X
5/DK/07	NO	Month = End Month	Day = DK or RF	Year = End Year	X
DK/DK/07	NO	Month = DK or RF	Day = DK or RF	Year = End Year	X
8/DK/DK	NO	Month > End Month	Day = DK or RF	Year = DK or RF	X
DK/12/DK	NO	Month = DK or RF	Day > End Day	Year = DK or RF	X
5/6/07	NO	Month = End Month	Day = End Day	Year = End Year	
2/2/08	YES	Month < End Month	Day < End Day	Year > End Year	
7/2/07	YES	Month > End Month	Day < End Day	Year = End Year	
5/10/07	YES	Month = End Month	Day > End Day	Year = End Year	
8/DK/07	YES	Month > End Month	Day = DK or RF	Year = End Year	
DK/DK/08	YES	Month = DK or RF	Day = DK or RF	Year > End Year	

**Checking for MATCHING DATES:**

When checking for matching event dates, first check Event Begin and End dates for:

- 1) Valid dates
- 2) Dates are within specified reference period. See Three Date Verification.

Next, check for matching event dates based on the following rules and additional rules specified at Question or Route Box:

- 1) If event date added is a Repeat Visit event, the event date is a match if the following is true:
  - a. Event added EVNT.VISTTYPE = 2/RepeatVisit.
  - b. Event added EVNT.EVBEGMM and EVNT.EVBEGYY should ^= DK.
  - c. Event added EVNT.EVBEGMM and EVNT.EVBEGYY should ^= RF.
  - d. Existing event EVNT.VISTTYPE = 2/RepeatVisit.
  - e. Event added EVNT.EVBEGMM = existing event EVNT.EVBEGMM.
  - f. Event added EVNT.EVBEGYY = existing event EVNT.EVBEGYY.
  - g. Other rules specified at Question or Route Box are true.
- 2) If event date added is not a Repeat Visit event, the event date is a match if the following is true:
  - a. Event added EVNT.VISTTYPE = 1/SingleVisit or EMPTY.
  - b. Event added EVNT.EVBEGMMM, EVNT.EVBEGDD and EVNT.EVBEGYY should ^= DK.
  - c. Event added EVNT.EVBEGMM, EVNT.EVBEGDD and EVNT.EVBEGYY should ^= RF.
  - d. Existing event EVNT.VISTTYPE = 1/SingleVisit or EMPTY.
  - e. Event added EVNT.EVBEGMM = existing event EVNT.EVBEGMM.
  - f. Event added EVNT.EVBEGDD = existing event EVNT.EVBEGDD.
  - g. Event added EVNT.EVBEGYY = existing event EVNT.EVBEGYY.
  - h. Other rules specified at Question or Route Box are true.

**Checking for OVERLAPPING IP DATES:****Event Begin and End dates overlap existing IP Event begin and end dates:**

IP Admission date = EVNT.EVBEGMM/EVNT.EVBEGDD/EVNT.EVBEGYY

IP Discharge date = EVNT.EVENDMM/EVNT.EVENDDD/EVNT.EVENDYY

If SP has not been discharged from hospital, EVNT.IPSTATUS=1/StillInHospital.

When checking for overlapping IP events, first check Event Begin and End dates for:

- 3) Valid dates
- 4) Dates are within specified reference period. See Three Date Verification.
- 5) Admission date is on or before Discharge date. See Two Date Verification. There is often a soft edit if the Admission Date matches the Discharge date.
- 6) Does the Admission date match an existing IP stay Admission date for the same Provider? The SP cannot be admitted to the same hospital twice on the same day. If any part of date = DK or RF, do not execute this check.

- 7) Does the end date match an existing IP stay end date for the same Provider? The SP cannot be discharged from the same hospital twice on the same day. . If any part of date = DK or RF, do not execute this check.

Next, compare the MP Event date or the IP Admission and Discharge dates against existing IP stays entered in the current round. See Three Date Verification.

An existing IP stay entered in the current round can be identified by:  
EVNT.EVNTRNDC=current round, or  
EVNT.EV95FLG=previous round (previous round event where the discharge date was entered in the current round).

An IP date overlaps an existing IP stay if:

- 1) The admission date is (after the admission date of existing IP stay) and ((before the discharge date of existing IP) or (existing IP is flagged as STILL IN HOSPITAL)).
- 2) The discharge date is (after the admission date of an existing IP stay) and ((before the discharge date of the existing IP) or (existing IP is flagged as STILL IN HOSPITAL)).
- 3) The admission date is before the admission date of existing IP stay and the discharge date is after the discharge date of existing IP stay.
- 4) The admission date is after the admission date of existing IP stay and the discharge date is before discharge date of existing IP stay.
- 5) The admission date matches admission date of existing IP stay and the discharge date matches discharge date of existing IP stay, admission date and discharge dates are not the same dates.

NOTE: We allow the SP to be admitted and discharged from one hospital on one day, and then be admitted to a second hospital on the same day.

**MP Event date overlaps an IP stay:**

MP Event date = EVNT.EVBEGMM/EVNT.EVBEGDD/EVNT.EVBEGYY

An MP event date overlaps an existing IP stay if:

- 1) The MP event date is (on or after the admission date of existing IP stay) and ((on or before the discharge date of existing IP) or (existing IP is flagged as STILL IN HOSPITAL)).

## **5. OUT OF SURVEY REFERENCE PERIOD FLAG**

### **ORP Check for Events with END DATES:**

**If (MRES.SPALIVE ^= 2/AliveAndInstitute and MRES.SPALIVE ^= 3/Deceased)  
and  
(EVNT.EVNTTYPE = IP and EVNT.IPSTATUS ^= 1/StillInHospital) or  
(EVNT.EVNTTYPE = IU) or  
(EVNT.EVNTTYPE = OM and (EVNT.RENTSTIL = 2, DK, or RF)))**

If (Event Begin Date is before MRES.SRPBDAT) and  
(Event End Date is before MRES.SRPBDAT),  
set EVNT.EVORPFLG=1/ORP.

**If (MRES.SPALIVE = 2/AliveAndInstitute or 3/Deceased) and  
(EVNT.EVNTTYPE = IP and EVNT.IPSTATUS ^= 1/StillInHospital) or  
(EVNT.EVNTTYPE = IU) or  
(EVNT.EVNTTYPE = OM and (EVNT.RENTSTIL = 2, DK, or RF)))**

If (Event Begin Date is before MRES.SRPBDAT) and  
(Event End Date is before MRES.SRPBDAT),  
set EVNT.EVORPFLG=1/ORP.

### **ORP check for Events with only BEGIN DATES:**

**If (MRES.SPALIVE ^= 2/AliveAndInstitute and MRES.SPALIVE ^= 3/Deceased)  
and ((EVNT.EVNTTYPE = DU, ER, OP, MP,SD, SL) or  
(EVNT.EVNTTYPE = IP and EVNT.IPSTATUS = 1/StillInHospital) or  
(EVNT.EVNTTYPE = OM and  
(EVNT.RENTSTIL ^= 2, DK, RF) and (EVNT.OMSTATUS ^=  
1/AlterationNotComplete) and (EVNT.OTHRTYPE ^= 4/OstomySupplies,  
5/IncontinenceSupplies, or 6/Bandages)))**

If Event Begin Date is before SRPBDAT, set EVNT.EVORPFLG=1/ORP.

**If (MRES.SPALIVE = 2/AliveAndInstitute or 3/Deceased) and  
(EVNT.EVNTTYPE = DU, ER, OP, MP,SD, SL) or  
(EVNT.EVNTTYPE = IP and EVNT.IPSTATUS = 1/StillInHospital) or  
(EVNT.EVNTTYPE = OM and  
(EVNT.RENTSTIL ^= 2, DK, RF) and (EVNT.OMSTATUS ^=  
1/AlterationNotComplete) and (EVNT.OTHRTYPE ^= 4/OstomySupplies,  
5/IncontinenceSupplies, or 6/Bandages)))**

If Event Begin Date is before SRPBDAT, set EVNT.EVORPFLG=1/ORP.



## **6. DATE CALCULATIONS**

---

If DOB Month, Day and Year ^= DK, RF

Calculated age = (Today's date minus Date of Birth) /365.23

Else if DOB year ^= DK, RF

Calculated age = (Current year minus DOB year)

If DOB month ^= DK, RF

If DOB month > today's month, calculated age = (calculated age minus 1).

Else age = unknown. Not sure how this is displayed.

We collect dates in the 2 separate fields:

We set dates during the between round rollover procedures in the following format:

MMDDYY

We do not set missing values to a "fixed" value during date comparison. If we cannot check a date properly because of a missing value, the date passes the check.