



LOUISIANA DEPT. OF HEALTH & HOSPITALS

---

Medicaid Eligibility Data System

# Internal Design: Core Common Modules



# Table of Contents

<b>DOCUMENT INFORMATION</b> .....	<b>5</b>
REVISION SUMMARY .....	5
<b>GLOBAL DATA AREA (MEXGDA)</b> .....	<b>6</b>
<b>SYSTEM COMMON MODULES</b> .....	<b>7</b>
DETERMINE DATE OVERLAP (MEXDDON) .....	7
<b>ADMINISTRATION COMMON MODULES</b> .....	<b>8</b>
GET WORKER NUMBER (MEAWRKN1) .....	8
STANDARD VALUE TABLE LOOKUP (MEASTVN) .....	8
TYPE/CASE LOOKUP (MEATYCN) .....	9
MONTHLY CUTOFF LOOKUP (MEAMOCN) .....	9
<b>CERTIFICATION COMMON MODULES</b> .....	<b>10</b>
FORMAT PERSON NAME FOR DISPLAY (MECFNDN) .....	10
GET CLIENT NAME/SSN AND FORMAT FOR DISPLAY (MECGCNN) (MECGCNN1) .....	11
UPDATE AND ADD AU MEMBERS (MECCCUN) .....	12
DETERMINE IF SENT TO MMIS (MECMISN) .....	13
DETERMINE IF SEGMENTS ARE REQUIRED (MECSGUN) .....	15
GET OPEN CPTC FOR A CERT (MECOCCN) .....	16
GET OPEN CPTC FOR A CERT (MECOCCN1) .....	18
CLOSE DETERMINATION (MECCDNN) .....	19
CHECK FOR ELIGIBLE BUDGETS (MECKBN) .....	20
VALIDATE BUDGET NUMBER WITHIN ELIGIBILITY DETERMINATION (MECVBDN) .....	21
COPY PERSON INCOME VALUES (MECCPIN) .....	22
CLOSE AU MEMBERS (MECAUCN) (MECCAMN1) .....	23
CLOSE SEGMENTS (MECSGCN) (MECCPSN1) .....	24
DETERMINE NUMBER OF DAYS (MECDNDN) .....	26
DETERMINE DATE (MECDDTN) .....	27
POPULATE EARLIEST RENEWAL DATE (MECERDN) .....	29
CHECK IF OPEN CERT EXISTS FOR CASE (MECECCN) .....	31
CHECK IF ADULT (MECADCN) .....	32
UPDATE THE CLIENT RUN NUMBER ON THE CASE FILE (MECUCRN) .....	33
UPDATE THE BENDEX RUN NUMBER ON THE AU-MEMBER FILE (MECUBNN) .....	34
UPDATE THE MMIS RUN NUMBER ON THE PERSON (MECUMPN) .....	35
UPDATE THE MMIS RUN NUMBER ON THE CERT PERIOD (MECUMCN) .....	36
DETERMINE IF SIEVS RUN NUMBER UPDATE IS REQUIRED (MECUSVN) .....	37
UPDATE THE SIEVS RUN NUMBER (MECUSUN) .....	40
CREATE BUDGET WORKSHEET CPTC (MECCBWN) .....	42
RETRIEVE STANDARD VALUE FROM STANDARD VALUE TABLE (MEASTVN) .....	43
MATCH PERSON TO EXISTING MEDS PERSON (MECMEPN) .....	44
COPY 'COMPLETE' TO 'IN PROGRESS' RECORDS (MECYCIN) .....	45
CHECK IF ALL CERTIFIED (MECCICN) .....	47
COPY 'IN PROGRESS' TO 'COMPLETE' (MECYICN) .....	48
PURGE 'IN PROGRESS' (MECPCLN) .....	49
CHECK FOR OVERLAPPING ELIG AT TIME OF WIP (MECOWPN) .....	51
CHECK FOR OVERLAPPING ELIG AT TIME OF CERTIFY (MECOCTN) .....	54
CHECK CUTOFF DATES (MECCCDN) .....	56
CONFIRM COMPLETION OF DETERMINATION (MECDETN1) .....	57

CHECK IF CERT CHANGED (MECCEPN).....	59
CHECK IF DUAL ELIGIBILITY ALLOWED (MECCDAN) .....	61
VALIDATE NAME FOR INVALID CHARACTERS (MECNAMN1) .....	62
VALIDATE ADDRESS AND CITY FOR INVALID CHARACTERS (MECADRN1) .....	63
CHECK CATEGORY COMBINATIONS ON CASES (MECCCCN) .....	64
FIND THE USER-ID FOR A CASE'S CASE WORKER (MEXRLUN) .....	66
DEAD PERSON'S REVIEW LIST CREATOR (MEXDPRN) .....	68
DETERMINE A PERSON'S LAST DEATH CLOSURE (MEXDPDN) .....	69
DETERMINE IF A PERSON IS LINKED TO ACTIVE ELIGIBILITY (MEXDPEN).....	70
BUILD AU MEMBER ELIGIBILITY FOR NEW CPTC (MECAMEN1) .....	71
BUILD AU MEMBER ELIGIBILITY FOR NEW AU MEMBER (MECAMEN2) .....	71
REBUILD AU MEMBER ELIGIBILITY FOR MODIFIED CPTC (MECAMEN3) .....	72
REBUILD AU MEMBER ELIGIBILITY FOR MODIFIED AU MEMBER (MECAMEN4) .....	73
DETERMINE AU MEMBER ELIGIBILITY DATES (MECAMEN5) .....	73
PURGE AU MEMBER ELIGIBILITY FOR PURGED CPTC (MECAMEN6) .....	75
PURGE AU MEMBER ELIGIBILITY FOR PURGED AU MEMBER (MECAMEN7) .....	75
ASSIGN ELIGIBILITY SEQUENCE NUMBER (MECPESN1) .....	76
ADD ELIGIBILITY SEGMENTS (MECPESN2).....	76
BUILD NEW ELIGIBILITY SEGMENTS (MECPESN3) .....	77
LOAD CURRENT ELIGIBILITY SEGMENTS (MECPESN4).....	78
DETERMINE DATE OVERLAP (MECPESN5).....	79
REMOVE SANCTION OVERLAP FROM ELIGIBILITY (MECPESN6) .....	80
REBUILD ELIGIBILITY SEGMENTS (MECPESN7) .....	81
DETERMINE SEGMENT OVERLAP (MECPESN8) .....	84
PURGE ELIGIBILITY SEGMENTS (MECPESN9).....	86
FIND ELIGIBILITY SEGMENTS (MECPESNA).....	86
RETRIEVE LATEST COMPLETED BUDGET FOR CERT (MECRLBN) .....	87
RETRIEVE LATEST UNCERTIFIED BUDGET FOR CERT (MECRUBN).....	88
DETERMINE INDICATED RENEWAL LEVEL FOR CERT (MECDRLN).....	89
RETRIEVE ACTIVE FOOD STAMP CERTIFICATION (MECRFSN).....	94
CHECK STABLE INCOME (MECCSIN) .....	96
RETRIEVE RESOURCE INFORMATION (MECRRIN) .....	98
DETERMINE PRIOR VALUE OF RENEWAL CODE FOR CERT (MECPRCN).....	100
LASES CASE VALIDATION FOR CERT (MECLCVN).....	102
<i>Restricted Type Cases (MECLCVL)</i> .....	104
<i>OCS Type Cases</i> .....	105

## Document Information

---

**Created for:** Louisiana Dept. of Health & Hospitals  
**Project Name:** Medicaid Eligibility Data System  
**Project Abbreviation:** MEDS  
**Document Title:** Internal Design: Core Common Modules  
**Document Subject:**  
**Revision Number:** 5 (incremented each time the document is saved)  
**Status:** Final  
**Last Saved:** Dec 16, 2011 03:18:00 PM  
**Printed:** Dec 16, 2011 03:18:00 PM  
**Comments:**  
**Document Template:** RedManeSpec\_W2000.dot

## Revision Summary

---

The following revisions have been made to the document since it was first published:

Date	Description of Change	By
4/30/2003	Revise with RedMane template	M. Smutko
2/24/04	Add AU Member Eligibility and Eligibility Segments modules	Jennifer Leslie
3/16/04	Add budget modules	Jackie Flosi
8/12/2005	Remove hospice (SIR1186)	Karl Powers
8/20/2009	Calculate the end of month date (MECDDTN)	Rajesh Cheebur
8/20/2010	Changed program name MEAWKRN1 to MEAWRKN1 – typing error  Add the following:  Standard value table lookup (MEASTVN)  Type/Case lookup (MEATYCN)  Monthly Cutoff lookup (MEAMOCN)	J. Badenhorst

## Global Data Area (MEXGDA)

---

Include the following fields in the MEDS global data area:

Worker Key

Supervisor Key

Budget Checking Required Indicator

Budget Method Indicator

# System Common Modules

---

## *Determine Date Overlap (MEXDDON)*

---

This common module will determine (true/false) if 2 dates overlap and what the overlapping date range is. Note: Close date of zero is interpreted as the maximum date possible (i.e. no end).

### Input

Date 1 Start Date  
Date 1 Close Date

Date 2 Start Date  
Date 2 Close Date

### Output

Overlap (L)  
Overlap Start Date  
Overlap Close Date

## Administration Common Modules

---

### *Get Worker Number (MEAWRKN1)*

---

This common routine will return the first Worker Number found in the Worker file (ME-WORKER) using the provided Security User Id. In addition, a counter of worker numbers is given to identify when multiple worker numbers exist for a single user id (max =2).

#### Input

User Id

#### Output

No-of-Worker-Nums-for-User  
 Worker Number  
 Supervisor Id  
 Budget-Checking-Required-Indicator

#### Initial processing

Reset Output variables.

#### Mainline Processing

Find (2) ME-WORKER file by SC-USER-ID with the Input.UserId

Assign Output. No-of-Worker-Nums-for-User = \*Counter(READ)

On the 1<sup>st</sup> record read

Assign Output.WorkerNumber = ME-WORKER.WorkerNumber

If ME-WORKER.Supvs-Id = ''

Assign Output.Supvs-Id = \*user

Else

Assign Output.Supvs-Id = ME-WORKER.Supvs-Id

Assign Output.Bud-check-reqd-indic = ME-WORKER.Bud-check-reqd-indic

End-Find

### *Standard value table lookup (MEASTVN)*

---

#### Input

Standard Value Code

Effective Date 9C

Index Value

#### Output

Standard Value

Effective date

Initial processing

Use Input Standard Value Code & Effective Date 9C to populate #Key.

Mainline Processing

Read (1) ME-STND-VAL by STND-VAL-CD-EFFCT-DATE-9C using #Key

Move requested index value to output and escape

## *Type/Case lookup (MEATYCN)*

---

Input

Category Code

Type Case

Output

Type Case Description

Initial processing

Use Input Category Code & Type Case to populate #Key.

Mainline Processing

Find (1) ME-TYPE-CASE with CATEGORY-CODE-TYPE-CASE equal to #Key

Move Type Case Description to output and escape

## *Monthly Cutoff lookup (MEAMOCN)*

---

Input

Category Code

Type Case

Output

Type Case Description

Initial processing

Use Input Category Code & Type Case to populate #Key.

Mainline Processing

Find (1) ME-TYPE-CASE with CATEGORY-CODE-TYPE-CASE equal to #Key

Move Type Case Description to output and escape

## Certification Common Modules

---

### *Format Person Name for Display (MECFNDN)*

---

This common routine will format a person's name into the correct display format as defined in the External Design Document.

#### Input

Last Name

First Name

Middle Initial

#### Output

Common error PDA (CXXGNLP)

Name (A30)

#### Initial processing

Reset Output variables. Required: Last name

#### Mainline Processing

Concatenate the I.Last-Name, I.First-Name and I.Middle-Initial into O.Name in the format as defined in the External Design document.

#### Error processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message

## *Get Client Name/SSN and Format for Display (MECGCNN) (MECGCNN1)*

---

These common routines will return the client(s) of the Case as defined in the External Design document.

MECGCNN – Retrieves the client name for completed records

MECGCNN1 – Retrieves the client name for in progress records

### Input

Case Number

Cert Period Number

Application Number

Person Number

### Output

Common error PDA (CXXGNLP)

Client Name

SSN

### Initial processing

Reset Output variables

### Mainline Processing

If the I.Person-Number is not blank, then perform Person Processing, set O.Client-Name and exit program.

If the I.Cert-Period-Number and I.Case-Number are not blank, read the Certification Period with Case Number / Cert Period Number to get the client number. Perform Person Processing for the client number, set O.Client-Name and exit the program.

If the I.Application-Number and I.Case-Number are not blank, read the Application file with Case Number / Application Number. Perform Person processing for Probable Client 1 and Probable Client 2. Concatenate the two probable clients' names into O.Client-Name and exit program.

If the I.Case-Number is not blank, read the Cert Period file with Case Number / Start Date 9C using the current date to get the active or else the most recent Cert Period for the Case.

If a record was found, then perform Person Processing for the Client Number. If there is more than one active Cert Period, perform Person Processing for the second Cert Period as well. Set O.Client-Name and exit the program.

If no records were found on the Cert Period file, then read the application file with Case Number / Appl Num 9C to get the most recent application. Perform Person processing for Probable Client 1 and Probable Client 2. Concatenate the two probable clients' names into O.Client-Name and exit the program.

If all the input fields are blank, then perform Error Processing.

## Person Processing

Read the Person file with the Person Number to get the Person record. Call the common routine 'Format Person Name for Display' to get the client name in the correct format.

## Error processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message

## *Update and Add AU Members (MECCCUN)*

---

This common module is used to close an existing certification and close the AU Members and add a new certification with new AU Members for each occurrence of the table passed (i.e. the replace action). It should only be called for in progress records. This will not replace completed certifications.

### Input:

#existing PDA  
 #new PDA  
 #AU member PDA (1:30)

### Output:

Common error PDA

### Processing:

Move by name #existing to the Maintenance PDA's (for Cert Period)  
 CALLNAT 'MECCRTU' to close the existing Certification, CPTC and AU members  
 Move by name #new to the Maintenance PDA's for Cert Period and CPTC.  
 CALLNAT 'MECCRTU' to add the new Certification Period and CPTC  
 For each input.AU-member  
   Move by name #new to the AU MEMBER maintenance PDA  
   Move new Cert Period Number to Au MEMBER maintenance PDA  
   CALLNAT 'MECCAMV'  
   If error  
     Perform escape-prog  
   CALLNAT 'MECCAMU'

If an error

Perform escape-prog

## *Determine if sent to MMIS (MECMISN)*

---

This common routine will determine if a Person or a Cert Period has been sent to MMIS as defined in the External Design document.

### Input

Person Number  
Case Number  
Cert Period Number  
AU Member Number  
Segment Number  
Segment Close Date  
Hospice Segment Number  
Hospice Segment Close Date

### Output

Common error PDA (CXXGNLP)  
Sent to MMIS (L)

### Initial processing

Reset Output variables.

### Mainline Processing

If the I.Person-Number is not blank, then

Read the Person file with I.Person-Number to get the MMIS-Person-Run-Num.

Call the Run Control object subprogram to get the next run number for the MMIS Person Extract.

Compare the Person.MMIS-Person-Run-Num to the Run-Control.Run-Number. If they are the same set O.Sent-to-MMIS equal to FALSE else set O.Sent-to-MMIS equal to TRUE.

If the I.Case-Number and I.Cert-Period-Number are not blank, then

Read the Certification Period file with Case Number / Cert Period Number to get the MMIS-Cert-Run-Num of the Cert Period.

Call the Run Control object subprogram to get the next run number for the MMIS Cert Period Extract.

Compare the Cert-Period.MMIS-Cert-Run-Num to the Run-Control.Run-Number. If they are the same set the O.Sent-to-MMIS equal to FALSE, else set it equal to TRUE.

If I.Segment-Num not blank

Read the Segment Log file by entry to get the MMIS Run Num when the segment was added.

Compare the Segment-Log.MMIS-Run-Num to the Run-Control.Run-Number. If they are the same set the O.Sent-to-MMIS equal to FALSE, else set it equal to TRUE.

If I.Segment-Close-Date not blank

Read the Segment Log file by entry to get the MMIS Run Num when the close was first sent to MMIS.

Compare the Segment-Log.MMIS-Run-Num to the Run-Control.Run-Number. If they are the same set the O.Sent-to-MMIS equal to FALSE, else set it equal to TRUE.

If I.Hospice-Segment-Num not blank

Read the Hospice Segment Log file by entry to get the MMIS Run Num when the segment was added.

Compare the Hospice-Segment-Log.MMIS-Run-Num to the Run-Control.Run-Number. If they are the same set the O.Sent-to-MMIS equal to FALSE, else set it equal to TRUE.

If I.Hospice-Segment-Close-Date not blank

Read the Hospice-Segment Log file by entry to get the MMIS Run Num when the close was first sent to MMIS.

Compare the Hospice-Segment-Log.MMIS-Run-Num to the Run-Control.Run-Number. If they are the same set the O.Sent-to-MMIS equal to FALSE, else set it equal to TRUE.

#### Error processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message

## *Determine if Segments are required (MECSGUN)*

---

This common routine checks whether an in progress segment was created for each CPTC in a particular certification period that requires a segment.

### Input

Case number

Certification period number

### Output

Common error PDA (CXXGNLP)

Segments-Required (L)

Segments-Exist (L)

### Initial processing

Reset Output variables

Required: Case Number, Certification Period number

### Mainline Processing

Read the CPTC file with the case number and certification period number.

For each CPTC:

Read the Type Case file with Category / Type Case. The LTC-segs-allow-indic on the Type Case file will indicate if segments are required for this type case. If LTC-segs-allow-indic is equal to 'Y', set O.Segments-required equal to TRUE and then read the segment file with the CPTC prime key to see if segments exist for the CPTC. If no records found, then set O.Segments-exist equal to FALSE else set Segments-exist equal to TRUE.

If O.Segments-required is equal to TRUE and O.Segments-exist is equal to FALSE then exit the common routine.

End of CPTC read.

### Error processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message

## *Get Open CPTC for a cert (MECOCCN)*

---

This common routine returns all open CPTCs for a particular certification period.

### Input

Case-number  
 Certification period number (Optional)  
 Certify Process (L)

### Output

Common error PDA (CXXGNLP)  
 Type-case1  
 Category1  
 Cert-Period-Num-1  
 Type-case2  
 Category2  
 Cert-Period-Num-2  
 None-current

### Initial processing

Reset Output variables  
 Required: Case Number  
 Optional: Certification Period Number

### Mainline Processing

Read-cert-period.

Read Cert period file with case-start-date-9c where case-number = I.case-number and start-date = 99999999 - \*datn

If Cert Period.Case-num ne I.Case-num

Escape

If I.Certify-process = False

Reject if Cert-period.elig-indic ne 'Y'

Read CPTC with case-cert-start-date-9c where case-num = Cert-period.case-num and Cert-period-num = Cert-period.cert-period-num and start-date-9c = (99999999 - \*datn)

If I.Certify-process = False

Reject if CPTC.elig-indic ne 'Y'

If CPTC is active

```
    If CPTC.Cert period num = I.Cert-period-num and I.Cert-period-num > 0
        Assign O.Cat-1 = CPTC.Category-code
        Assign O.Type-case-1 = CPTC.type-case
    Else
        Assign O.Cat-2 = CPTC.category-code
        Assign O.Type-case-2 = CPTC.type-case
    If I.Cert-period-num = 0
        If O.Cat-1 = 0
            And O.Type-case-1 = 0
                Assign O.Cat-1 = CPTC.category-code
                Assign O.Type-case-1 = CPTC.type-case
            Else
                Assign O.cat-2 = CPTC.category-code
                Assign O.Type-case-2 = CPTC.type-cas
    End-read /* Read-cert-period.
    If O.Cat-1 = 0
        And O.Type-case-1 = 0
        And O.Cat-2 = 0
        And O.Type-case-2 = 0
        Assign O.None-current = true
```

#### Error processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message

## *Get Open CPTC for a Cert (MECOCCN1)*

---

This common routine returns all open CPTCs for a particular Certification Period.

### Input

Case Number

Certification Period Number

### Output

Category1

Type-Case1

Category2

Type-Case2

None Current indicator

Common error PDA (CXXGNLP)

### Initial processing

Reset Output variables

Required: Case Number, Certification Period number

### Mainline Processing

CHECK ELIGIBLE/ACTIVE CERT-PERIOD TOO?????

Read the CPTC file with the I.Case number, I.Certification period number and find the CPTCs that are currently active. A CPTC is current if the Eligible Indicator is set to "Y" and the close date of the CPTC is after current date or there is no close date. For the first CPTC found assign O.Type-case1, O.Category1.

As "there can never be more than one active CPTC per Cert Period (ever!)", if there is a second current CPTC, assign O.Type-case2 and O.Category2, but also populate the error message. Continue to read CPTCs and output the total number of current active ones in the error message.

If no current CPTCs are found, flag the None Current indicator.

If no records found, return blank outputs.

### Error processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message

## *Close Determination (MECCDNN)*

---

This common routine purges all those in progress records linked to the determination and closes the determination

### Input

Case-number

Determination number

### Output

Common error PDA (CXXGNLP)

### Initial processing

Reset Output variables

Required: Case Number, Determination number,

### Mainline Processing

Histogram ME-ELIG-CERT-PERIOD for case-num = I.case-num and elig-deter-num = I.elig-deter-num and for each cert period found, call common module 'Purge in progress'.

Update determination status to 'Completed'

Update determination date to current date

*Technical Note: The calling program is responsible for issuing an ET.*

### Error processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message

## *Check for eligible budgets (MECCKBN)*

---

This common routine checks whether budgets exist for a particular determination and whether any eligible budgets exist for a determination.

### Input

Case-number

Determination number

### Output

Common error PDA (CXXGNLP)

No-buds (L)

Elig-buds (L)

Incomplete-buds (L)

### Initial processing

Reset Output variables

Required: Case Number, Determination number

### Mainline Processing

Read the Budget-worksheet with I.Case number and I.Determination number

If no records found

Set O.No-buds to true

Escape

End-if

If Bud det outcome indic = eligible

Set O.Elig bud to true (*Technical Note: Read must continue – incomplete budgets must be found*)

End-if

If Budget status = Incomplete

Set O.Incomplete buds to true

Escape read (*Technical Note: Only one incomplete budget need be found*)

End-if

### Error processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message

## *Validate budget number within eligibility determination (MECVBDN)*

---

This common routine checks whether a budget exists within a particular eligibility determination.

### Input

Case number

Determination number

Budget number

### Output

Common error PDA (CXXGNLP)

Valid-Budget (L)

### Initial Processing

Reset output variables. Required: Case number, determination number, budget number

### Mainline Processing

Find the I.Case number, I.Determination number and I.Budget number on the budget worksheet file.

If no records are found, then set O.Valid-Budget equal to FALSE else set O.Valid-Budget equal to TRUE.

### Error processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message

## *Copy Person Income values (MECCPIN)*

---

This common routine updates the Person Income Value file with values copied from a particular budget.

### Input

Case number

Certification Period Number

Budget number

### Output

Common error PDA (CXXGNLP)

### Initial Processing

Reset output variables. Required: Case number, certification period number, budget number

### Mainline Processing

Read the budget worksheet section file with the I.case number, I.certification period number and I.budget number. With each budget section read for the budget, read the budget worksheet line file with Case number, certification period number, budget number and budget section number. For each income type, update the income value on the Person Income Value file.

*Technical Note: The calling program is responsible for issuing an ET.*

### Error processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message

## *Close AU Members (MECAUCN) (MECCAMN1)*

---

These common routines close all AU members for a particular case and certification period.

MECAUCN – Closes completed AU members

MECCAMN1 – Closes in progress AU members

### Input

Case Number

Certification Period Number

Close Date

Close Code

### Output

Common error PDA (CXXGNLP)

### Initial Processing

Reset Output variables.

Required: Case Number, Certification Period Number, Close Date, Close code

### Mainline Processing

Read all the AU members for the I.certification period in the I.case. For each AU member, call the object subprogram of AU Member Maintenance with the I.close date and I.close code.

*Technical Note: The calling program is responsible for issuing an ET.*

### Error Processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message.

## *Close Segments (MECSGNC) (MECCPSN1)*

---

These common routines close all Segments for a particular case and certification period.

MECSGNC – Closes completed segments

MECCPSN1 – Closes in progress segments

### Input

Case Number

Certification Period Number

CPTC Number

Close Date

Closure Code

Discharge Date

### Output

Common error PDA (CXXGNLP)

### Initial Processing

Reset Output variables.

Required: Case Number, Certification Period Number, CPTC number, Close Date, Closure Code

### Mainline Processing

Read all the Segments for the I.certification period in the I.case. For each segment, call the object subprogram of Segment Maintenance with the I.close date and I.closure code.

*Technical Note: The calling program is responsible for issuing an ET.*

### Error Processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message.



## *Determine Number of Days (MECDNDN)*

---

This common routine calculates the difference in the number of days between two dates.

### Input

From Date

To Date

### Output

Number of Days

Common error PDA (CXXGNLP)

### Initial Processing

Reset Output variables.

Required: From Date, To Date

### Mainline Processing

Calculate the difference between the two dates in number of days.

### Error Processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message.

## *Determine Date (MECDDTN)*

---

This common routine calculates a date using the From Date as the base date. The number of days are either added to or subtracted from the from date depending on the indicator.

### Input

From Date  
 Number of days  
 Number of months  
 Number of years  
 Indicator

### Output

Calculated Date  
 Common error PDA (CXXGNLP)

### Initial Processing

Reset Output variables.  
 Required: From Date, Number of Days, Number of months or Number of years, Indicator  
 Validate that I.Indicator is equal to 'L' (Less), 'M' (More) or 'E' (End of month).

### Mainline Processing

```

If I.Number-of-years > 0
  If I.Indicator = 'L'ess
    Compute O.Calculated-date = I.From-date-yyyy - I.Number-of-years
  If I.Indicator = 'M'ore
    Compute O.Calculated-date = I.From-date-yyyy + I.Number-of-years

If I.Number-of-months > 0
  Determine if leap year
  If O.Calculated date > 0
    Assign #Local.date = O.Calculated-date
  Else
    Assign #Local,.date = I.From-date
  If I.Indicator = 'M'ore
    Add I.Number-of-months to #Local.date
  
```

Else /\* indicator = less

Subtract I.Number-of-months to #Local.date

Assign O.Calculated-date = #Local.date

If I.Number-of-days > 0

If O.Calculated-date = 0

Assign O.Calculated-date = I.From-date

If I.Indicator = 'M'ore

Add I.Number-of-days to O.Calculated-date

Else

Subtract I.Number-of-days from O.Calculated-date

If I.Indicator = 'E'nd of month

Assign #Local.date = I.From-date

Assign #Local.date.day = last day of month

If leap year assign correct last Feb day if I.From-date-month is Feb

Calculate the end of month date

If the I.indicator is 'L' then the calculation will be  $O.Calculated\ Date = I.From\ Date - I.Number\ of\ days$ .

If the I.indicator is 'M' then the calculation will be  $O.Calculated\ Date = I.From\ Date + I.Number\ of\ days$ .

#### Error Processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message.

## *Populate Earliest Renewal Date (MECERDN)*

---

This common routine will populate the earliest re-determination date on case with the correct value

### Input

Case Number

Cert Period Number

Renewal Date

Close Date

### Output

Common error PDA (CXXGNLP)

### Initial Processing

Reset Output variables.

Required: Case Number

Cert Period Number

Setup the start value for the earliest date. If there is a I.Close-Date and it is in the past, set the Earliest Date to '99999999', otherwise if the I.Close-Date is in the future or absent set the Earliest Date to the I.Renewal-Date Passed.

### Mainline Processing

Read the CPTC file with the key Case-Cert-Start-Date-9C.

Upon a change in Case-Num exit the read.

If the Cert-Period-Num read is equal to the I.Cert-Period-Num.

If I.Close-Date is in the past

Subtract 1 from the read counter.

End-if

Escape to the top of the read.

End-If

If the Close-Date read is in the past.

Subtract 1 from the read counter

Escape to the top of the read.

End-if

If the Renewal-Date read is less then the Earliest-Date

Move the Renewal-Date read to the Earliest-Date.

End If

End-Read

Get the Case Record using I.Case-Num

If the I.Close-Date is not zero, in other words the CPTC is being closed/ is closed

And the read counter is equal to zero, in other words no open records were found.

Move Zero to the Earliest-Renewal-Date field on the Case Record

Else

Move Earliest-Date to the Earliest-Renewal-Date field on the Case Record

End-if

Update the Case Record

*Technical Note: The calling program is responsible for issuing an ET.*

Error Processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message.

## *Check if Open Cert exists for Case (MECECCN)*

---

This routine will determine if a Case has an open certification period.

### Input

Case Number

### Output

Open Cert (L)

Common error PDA (CXXGNLP)

### Initial Processing

Reset Output variables.

Required: Case Number

### Mainline Processing

Find the Cert Period for the I.Case Number using the current date.

If a record was found, check the Close date against the current date.

If the Cert Period is current, set the O.Open Cert variable to TRUE else set the variable to FALSE.

If no records were found set the O.Open Cert output variable to FALSE.

### Error Processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message.

## *Check if Adult (MECADCN)*

---

This routine will determine if a person is an adult or not.

### Input

Person Number

Date

### Output

Adult (L)

Common error PDA (CXXGNLP)

### Initial Processing

Reset Output variables.

Required: Person Number, Input Date

### Mainline Processing

Read the person file for the I.Person Number. Add 18 to person's birth date giving #local.date

If #local.date > I.Date set the O.Adult variable to FALSE else set it to TRUE.

### Error Processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message.

## Update the Client Run Number on the Case File (MECUCRN)

---

This routine will update the client run number on the Case File so that the case records will be retrieved during the Client Interface extract process.

### Input

Case Number

### Output

Common error PDA (CXXGNLP)

### Initial Processing

Reset Output variables.

Required: Case Number

### Mainline Processing

Call the object subprogram to get the run number for the Client Interface Extract from the Run Control File.

Find the record in the Case File where Case number = I.Case Number

Call the Case Object Subprogram to update the Case record with client run number = Run Control.run number

*Technical Note: The calling program is responsible for issuing an ET.*

End-Find

### Error Processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message.

## *Update the BENDEX Run Number on the AU-Member File (MECUBNN)*

---

This routine will update the BENDEX run number on the AU-Member File so that the record will be retrieved during the BENDEX Interface extract process.

### Input

Case Number  
Cert Period Number  
AU Member Number

### Output

Common error PDA (CXXGNLP)

### Initial Processing

Reset Output variables.

Required: Case Number AND Cert Period Number AND AU Member Number

### Mainline Processing

Call the object subprogram to get the Run Number for the BENDEX Interface Extract from the Run Control File.

Call the object subprogram to find the record in the AU Member file  
where Case Number = I.Case Number  
and Cert Period Number = I.Cert Period Number  
and AU Member Number = I.AU Member Number

Call the AU Member Object Subprogram to update the AU-Member record with BENDEX Run Number = Run Control.Run Number

*Technical Note: The calling program is responsible for issuing an ET.*

### Error Processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message.

## *Update the MMIS Run Number on the Person (MECUMPN)*

---

This routine will update the MMIS Person run number on the Person File so that the record will be retrieved during the MMIS Interface Person extract process.

### Input

Person Number

### Output

Common error PDA (CXXGNLP)

### Local

Sent-to-MMIS (L)

### Initial Processing

Reset Output and Local variables.

Required: Person Number

### Mainline Processing

Read the AU Member file with the Person number / Start date 9C. If the Person has Medicaid eligibility, set Sent-to-MMIS to TRUE, else check if the Person is a client on any Cert Periods by reading the Cert Period file with I.Person-Number. If the Person is a client for any Cert Periods, set Sent-to-MMIS to TRUE.

If Sent-to-MMIS is TRUE then

Call the Run Control object subprogram to get the run number for the MMIS Person Extract process.

Find the record in the Person File where person number = I.Person Number.

Call the Person Object Subprogram to update the Person record with MMIS Person run number = Run Control.run number.

*Technical Note: The calling program is responsible for issuing an ET.*

### Error Processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message.

## Update the MMIS Run Number on the Cert Period (MECUMCN)

This routine will update the MMIS Cert run number on the Certification Period File so that the record will be retrieved during the MMIS Interface Certification Period extract process.

### Input

Cert Period Number

### Output

Common error PDA (CXXGNLP)

### Initial Processing

Reset Output variables.

Required: Cert Period Number

### Mainline Processing

Call the Run Control object subprogram to get the run number for the MMIS Cert Period Extract.

Find the record in the Cert Period File where cert period number = I.Cert-Period-Number.

Call the Cert Period Object Subprogram to update the Cert Period record with MMIS Cert run number = Run Control.run number.

*Technical Note: The calling program is responsible for issuing an ET.*

### Error Processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message.

## Determine if SIEVS Run Number Update is Required (MECUSVN)

This routine will determine if the SIEVS run number should be updated, and if so it calls the SIEVS run number update module. It's called from Person, Person Gross Income, CPTC, AU Member, Case and Caseload objects.

### Input

Type of Change – P for Person, A for AUM, C for CPTC, I for income, W for worker location  
 Person Number  
 SSN  
 Case Number  
 Cert Period Number  
 AU Member Number

### Output

Common error PDA (CXXGNLP)

### Initial Processing

Reset Output variables.

Required: Type of Change

If Type of Change = 'P'

Person Number and SSN are required

End If

If Type of Change = 'A'

Case Number, Cert Period Number and AU Member Number are required

End If

If Type of Change = 'C'

Case Number and Cert Period Number are required

End If

If Type of Change = 'I'

Person Number, Case Number and Cert Period Number are required

End If

If Type of Change = 'W'

Case Number is required

End If

### Mainline Processing

If I.Type of Change = 'P' /\* person

/\* check for AU Members

Read active AU Members for the input Person Number

Check if the Type Case should be sent to SIEVS

Call the SIEVS Run Number Update Subprogram

End Read

/\* check for IU Members

Read Budget Worksheet Applicants for the input Person Number

Accept if the budget member is an IU Member only (not an AU Member)

Read the Budget Worksheet CPTC and Budget Worksheet for the Budget Applicant

Accept if the budget is eligible

```

        Find the Cert Period for the Budget Worksheet CPTC
            Accept if the Cert Period is open and same type case as the Budget Worksheet
            Check if the Type Case should be sent to SIEVS
            Call the SIEVS Run Number Update Subprogram
        End Find
    End Read
End Read
End If

If I.Type of Change = 'A' /* AU Member
    /* check for AU Members only
    Find AU Members for the input Case, Cert Period and AU Member Numbers
        Check if the Type Case should be sent to SIEVS
        Call the SIEVS Run Number Update Subprogram
    End Find
End If

If I.Type of Change = 'C' /* CPTC
    /* only need to check for IU Members (AU Members will have type of change 'A')
    Read CPTC for the input Case and Cert Period Numbers
        Check of the Type Case should be sent to SIEVS
        Read Eligibility Cert Period for input Case and Cert Period Numbers
            Read the Budget Worksheet CPTC for the Eligibility Cert Periods Case, Cert
                Period and Eligibility Determination Number
            Read the Budget Worksheet Applicants for the Budget CPTCs Case, Cert Period,
                Eligibility Determination numbers and IU Indicator.
                Accept if the budget member is an IU Member only
                Call the SIEVS Run Number Update Subprogram
            End Read /* budget applicant
        End Read /* budget worksheet CPTC
    End Read /* Eligibility CP
End Read /* CPTC
End If

If I.Type of Change = 'I' /* income
    /* check for AU Members
    Read active AU Members for the input Case and Cert Period Numbers
        Accept if the AU Member.Person Number = I.Person Number
        Check if the Type Case should be sent to SIEVS
        Call the SIEVS Run Number Update Subprogram
    End Read
    /* check for IU Members
    Find active CPTC for input Case and Cert Period Numbers
        Check of the Type Case should be sent to SIEVS
        Read Budget Worksheet Applicants for the input Case and Person Numbers
            Accept if the budget member is an IU Member only (not an AU Member)
            Read the Budget Worksheet CPTC and Budget Worksheet for the Budget Applicant
                Accept if the budget is eligible and the same type case as the CPTC
                Call the SIEVS Run Number Update Subprogram
            End Read /* Budget Worksheet CPTC
        End Read /* Budget Applicants
    End Read /* CPTC
End If

```

```
If I.Type of Change = 'W' /* worker location
  Read active CPTCs for the input Case Number
  Check if the Type Case should be sent to SIEVS
  /* check for AU Members
  Read active AU Members for the CPTC Case and Cert Period Numbers
  Call the SIEVS Run Number Update Subprogram
  End Read
  /* check for IU Members
  Read Eligibility Cert Period for input Case and Cert Period Numbers
  Read the Budget Worksheet CPTC for the Eligibility Cert Periods Case, Cert
    Period and Eligibility Determination Number
  Read the Budget Worksheet Applicants for the Budget CPTCs Case, Cert Period,
    Eligibility Determination numbers and IU Indicator.
  Accept if the budget member is an IU Member only
  Call the SIEVS Run Number Update Subprogram
  End Read /* budget applicant
  End Read /* budget worksheet CPTC
  End Read /* Eligibility CP
  End Read /* CPTC
End If
```

#### Error Processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message.

## Update the SIEVS Run Number (MECUSUN)

This routine will update the SIEVS run number on the AU-Member file or the Budget Worksheet Applicant file so that the record will be retrieved during the SIEVS Interface extract process.

### Input

Member Type – 'A' for AU Members, 'I' for IU Members  
 Case Number  
 Cert Period Number  
 Person Number  
 SSN  
 AU Member Number (for AU only)  
 Eligibility Determination Number (for IU only)  
 Budget Number (for IU only)

### Output

Common error PDA (CXXGNLP)

### Initial Processing

Reset Output variables.

Required: Member Type, Case Number, Cert Period Number, Person Number and SSN.

If Member Type = 'A'  
     AU Member Number is required  
 End If  
 If Member Type = 'I'  
     Eligibility Determination Number and Budget Number are required  
 End If

### Mainline Processing

Call the object subprogram to get the Run Number for the SIEVS Interface Extract from the Run Control File.

If I.Member Type = 'A'  
     Call the object subprogram to find the record in the AU Member file  
     where Case Number = I.Case Number  
     and Cert Period Number = I.Cert Period Number  
     and AU Member Number = I.AU Member Number  
  
     If I.SSN = 9..... /\* pseudo SSN  
         Reset AU Member.SIEVS Run Number  
     Else  
         Assign AU-Member.SIEVS Run Number = Run Control.Run Number  
     End If  
  
     Call the AU Member Object Subprogram to update the AU-Member record  
 End If  
  
 If I.Member Type = 'I'

Call the object subprogram to find the record in the Budget Worksheet Applicant file  
where Case Number = I.Case Number  
and Eligibility Determination Number = I.Eligibility Determination Number  
and Budget Number = I.Budget Number  
and Person Number = I.Person Number

If I.SSN = 9..... /\* pseudo SSN

    Reset Budget Worksheet Applicant.SIEVS Run Number

Else

    Assign Budget Worksheet Applicant.SIEVS Run Number = Run Control.Run Number

End If

Call the AU Member Object Subprogram to update the AU-Member record

End If

*Technical Note: The calling program is responsible for issuing an ET.*

#### Error Processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message.

## *Create budget worksheet CPTC (MECCBWN)*

---

This routine is used to check if a budget worksheet CPTC exists and if not to create this relationship.

### Input

Case number  
Determination number  
Budget number  
Cert period number  
Category code  
Type case

### Output

Common error PDA (CXXGNLP)

### Initial processing

Reset output variables

Required: Case number, determination number, cert period number. Category code, type case

### Mainline processing

Concatenate I.Category-code and I.Type-case.

Read the Budget worksheet CPTC with I.Case-number, I.Determination-number, I.Cert-period-number, cat-type-case. If no records exist, then add a record with I.Case-number, I.Determination-number, I.Cert-period-number and cat-type-case to the file.

*Technical Note: The calling program is responsible for issuing an ET.*

### Error Processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message.

## *Retrieve standard value from Standard Value table (MEASTVN)*

---

This routine is used to retrieve the standard value from the standard value table.

### Input

Standard Value Code

Effective Date 9C

Index Value

### Output

Standard Value

Common error PDA (CXXGNLP)

### Initial processing

Reset output variables

Required: Standard Value Code, Effective Date 9C, Index Value

### Mainline processing

*Read the first standard value record with the Standard Value Code and Effective Date 9C equal to or greater than the I.Standard Value Code and I.Effective Date 9C. If there are no standard value records with the same value as I.Standard Value Code escape routine with error logical set to true. When a standard value record is found, find the first O.Standard Value that corresponds to the I.Index Value and escape routine. Also, if there are no matching O.Standard Values escape routine with error logical set to true.*

### Error Processing

If any errors occur during processing, set O.Error to true and populate O.Error Message with an appropriate message.

## General Common Routines

## *Match person to existing MEDS Person (MECMEPN)*

---

This common module will take a set of person information and determine whether or not an existing MEDS person is a match for the given data based on a set of rules provided by DHH. Any errors returned from this routine are either caused by faulty input parameters or data integrity problems in the PERSON file.

**Input**

Last Name  
First Name  
Social Security Number  
Date of Birth

**Output**

Common error PDA (CXXGNLP)  
Match Found (L)  
Match Person Number

**Initial Processing:**

Reset the output parameters  
Validate that all required parameters have been supplied

**Mainline Processing:**

Retrieve the Person entity with a SSN equal to I.SSN  
If none found, go to next step  
Else If Last Name = I.Last Name OR First Name = I.First Name  
Set the O.Match Found Logical to TRUE. Assign the Person Number to O.Match Person Number  
Else  
Return an error stating that the I.SSN is in use by a different person

Retrieve the Person record(s) with LAST-FIRST-BIRTH = I.LastName, I.FirstName, I.DateOfBirth  
If any records are returned with a SSN different from I.SSN return an error stating that the person exists with a different SSN

**TechNote:** The second check is for data integrity checking only; no matches will be found.

**Error Processing:**

If any errors occur during processing, assign O.Error = TRUE. O.Error Message should be assigned an appropriate message.

## *Copy 'Complete' to 'In Progress' records (MECYCIN)*

---

This common module copies the Cert period and all of the related child records across from the 'complete' files to the 'in progress' files. This module is executed when creating an elig-cert-period record

### Input

Case Num  
Cert period num

### Output

CXXGNLP (common error PDA)

Required fields: Case number cert period num

### Mainline processing

Read ME-CERT-PERIOD with case-cert-period-num where case-num = I.case-number and cert period num = I.cert-period-num.

Verify that all 'In Progress' files are empty (for input Cert Period Number)

ME-CERT-PERIOD-IP  
ME-SEGMENT-IP  
ME-AU-MEMBER-IP  
ME-LASES-CASE-IP  
ME-LASES-CASE-CHILD

Copy 'Complete' Cert Period to "In Progress" Cert Period

Move by name ME-CERT-PERIOD to #ME-CERT-PERIOD-IP  
CALLNAT 'MECCRTU' with Copy action.

Copy 'Complete' Segments to "In Progress" Segments

Read ME-SEGMENT by case-cert-cptc-seg-num

Move by name ME-SEGMENT to #ME-SEGMENT-IP  
CALLNAT 'MECCPSU' with Copy action

End-read /\* SEGMENT

Copy 'Complete' AU members to "In Progress" AU Members

Read ME-AU-MEMBER by case-cert-au-mbr

Move by name ME-AU-MEMBER to #ME-AU-MEMBER-IP  
CALLNAT 'MECCAMU' with Copy action

End-read /\* AU-MEMBER

Copy 'Complete' LASES Cases to "In Progress" LASES Cases

Read ME-LASES-CASE by case-cert-ap

Move by name ME-LASES-CASE to #ME-LASES-CASE-IP  
CALLNAT 'MECLSPU' with display action and then add action

End-read /\* LASES-CASE

Copy 'Complete" LASES Case Child to "In Progress" LASES Case Child

Read ME-LASES-CASE-CHILD by ap-case-cert-au-mbr

Move by name ME-LASES-CASE-CHILD to #ME-LASES-CASE-CHILD-IP

CALLNAT 'MECLCPU' with display action and then add action

End-read /\* LASES-CASE-CHILD

## *Check if all certified (MECCICN)*

---

This common module reads through all certifications linked to the determination. If there is at least one certification that has not been certified (i.e. has no deter date), then return a logical true

### Input

Case num

Elig deter num

### Output

#not-all-certified (I)

Required: Case num, elig deter num

### Mainline processing

Read ME-ELIG-CERT-PERIOD with case-deter-cert-num where case-num = I.case-num, deter-num = I.elig-deter-num

If there is one certification that has not determination date, then set O.#not-all-certified = true

### Error processing

If any errors occur during processing, assign O.Error = TRUE. O.Error Message should be assigned an appropriate message.

## Copy 'In progress' to 'Complete (MECYICN)

When the Certification gets certified, the 'In progress' records will be copied across to the 'Completed' file. Only those certifications that have been marked for certify are to be copied. This module should only be called once the certifications that have been marked have been validated.

### Input

Case num  
Cert period num

### Output

Common error PDA

Required fields: Case num, cert period num

### Mainline processing

```
CALLNAT 'MECCRTU' with display action to retrieve in progress record
Move by name #ME-CERT-PERIOD-IP to #ME-CERT-PERIOD /* to retrieve completed record
CALLNAT 'MECCPEPU' with display action
If no records found,
Add #ME-CERT-PERIOD
Else,
    If #me-cert-period-ip.PURGE-INDIC = true
        Purge #ME-CERT-PERIOD
        Purge #ME-CERT-PERIOD-IP
    Else
        Modify #ME-CERT-PERIOD
Update MECCRTU with CHANGE-INDIC = 'C'
```

With the same logic, copy the following:

Copy all lases case and lases case child records for I.Case and I.Cert period

Copy all segment records for I.Case and I.cert period

Copy all AU member records for I.case and I.cert period

Copy cert period record for I.case-num and I.cert period

## *Purge 'In progress' (MECPCLN)*

---

When completing the determination, all the In progress certification records linked to the determination must be purged from the In progress files

### Input

Case number

Cert period num

### Output

CXXGNLP (common error PDA)

Required fields: Case number, cert period num

### Mainline Processing

Read ME-CERT-PERIOD-IP with case-cert-period-num where case-num = I.Case number and Cert-period-num = I.cert-period-num

Move by name ME-CERT-PERIOD-IP to #ME-CERT-PERIOD-IP

CALLNAT 'MECCRTU' with purge action

Move by name ME-CERT-PERIOD-IP to #segment-key

Read ME-SEGMENT with case-cert-cptc-seg-num = #segment-key

Move by name ME-SEGMENT to #ME-SEGMENT

CALLNAT 'MECCSPU' with purge action

End-read /\* ME-SEGMENT

Move by name ME-CERT-PERIOD to #AU-member-key

Read ME-AU-MEMBER with case-cert-au-mbr = #au-member-key

Move by name ME-AU-MEMBER to #ME-AU-MEMBER-IP

CALLNAT 'MECCAMU' with purge action

End-read /\* ME-AU-MEMBER

Move by name ME-CERT-PERIOD to #lases-case-key

Read ME-LASES-CASE with case-cert-ap = #lases-case-key

Move by name ME-LASES-CASE to #ME-LASES-CASE-IP

CALLNAT 'MECLSPU' with purge action

Move by name ME-LASES-CASE to #lases-case-child-key

Read ME-LASES-CASE-CHILD with ap-case-cert-au-mbr = #lases-case-child-key

Move by name ME-LASES-CASE-CHILD to #ME-LASES-CASE-CHILD-IP

CALLNAT 'MECLCPU' with purge action

End-read /\* LASES-CASE-CHILD

End-read /\* LASES-CASE

End-read /\* CERT-PERIOD

## *Check for overlapping elig at time of WIP (MECOWPN)*

---

### Rules:

1. If no completed record overlaps– add WIP
2. If completed overlaps – get WIP of completed and check if WIP overlaps. If WIP doesn't overlap – add WIP. *(Technical Note: We need to check if this WIP overlaps to prevent two certified records from overlapping. If there was no overlapping completed record, then the overlapping WIPS would just be ignored until certify time)*

### Input

#me-au-member (this represents the WIP record being maintained)

### Output

CXXGNLP (Common error output)

Required fields: Person Number

### Local

#effct-close-date

#overlapping

#count

#date-overlap

### Mainline processing

Read ME-AU-MEMBER (this represents the Certified record for the person – completed file)

for person-num, start-date-9c where person-num = #me-au-member.person-num.

If ME-AU-MEMBER.person-num ne #me-au-member.person-num

Escape bottom /\* No other completed AU member records exist for the person.

End-if

Set up #effct-close-date from close date and suspend date

If #count > 1 /\* Dual eligibility already exists.

Escape bottom (ME-AU-MEMBER read)

End-if

/\* Check for certified records that may overlap

CALLNAT common module to check for overlapping dates (pass start date and #effct-close-date of certified and WIP records)

If #date-overlap = true

/\* we have found a completed record that overlaps. Get the WIP record for this completed record and check if this WIP overlaps with the WIP being maintained. If it does then check if dual certs are allowed, else the changes to the WIP being maintained are valid

Move by name ME-AU-MEMBER ( C ) to #me-au-member-ip

CALLNAT 'MECCAMU' /\* In progress update subprogram

#me-au-member-ip

If #error /\* No WIP exists for the completed record

Assign #overlapping = true

Assign #ISN = \*isn

Add 1 to #count

Else /\* A WIP was found for the certified record

If #me-au-member.changed-indic = 'C' /\* If WIP is certified, then WIP = completed record and we have established that the completed record overlaps therefore this WIP will overlap

Assign #overlapping = true

Assign #ISN = \*isn

Add 1 to #count

Else /\* At this point we have a completed record overlapping and the WIP of this record is not certified yet. We must check if this WIP overlaps with the WIP being maintained

CALLNAT common module to check if dates overlap passing the start date and close date of both WIP and certified WIP

If #date-overlaps = true

Assign #overlapping = true

Assign #ISN = \*isn

Add 1 to #count

End-if /\* Date of WIP overlaps

End-if /\* WIP found not certified

End-if /\* WIP found

End-if /\* Date of completed record overlaps

End-read

If #count > 1 /\* Two completed records found there can be no other eligibility

Error – Dual eligibility already exists

End-if

If #overlapping = true

Check dual indicator on type case for completed overlapping record found (use #isn to get record)

If no dual allowed

Check for dual indicator on type case for in progress record validating

If no dual allowed

If #ME-AU-MEMBER.close-date = 0 and ME-AU-MEMBER.close-date = 0

Perform error-processing

else

Check cutoff dates to determine whether retro-active dual cert.

*Technical Note: Call common module 'Check cutoff dates'*

*Technical Note: Send the start date of the AU member being modified (i.e. #ME-AU-MEMBER.start-date. Compare the close date of the AU member being modified (#ME-AU-MEMBER) and the close date of the overlapping AU member record found (ME-AU-MEMBER). Pass the smallest close date (OTHER THAN 0) to the common module*

Error processing

If error, assign O.error = true and set up appropriate error message. Escape program

## *Check for overlapping elig at time of certify (MECOCTN)*

---

When certifying the certification, check that no AU members within that certification have eligibility that overlaps with any existing (certified) eligibility

### Input

Case Num

Cert period num

### Output

CXXGNLP (common error PDA)

Required fields: Case num and cert period num

### Local

#isn

#count

#overlapping (L

### Mainline processing

Read ME-AU-MEMBER-IP for case-cert-au-mbr where case = I.case-num and cert = I.cert-period-num

Check that the start date for this AU member is not greater than the next months end date (according to the monthly cutoff table)

*Technical Note: Callnat common module to determine date for current month year plus 1 month. Read ME-MONTHLY-CUTOFF with year-month where year = calculated year and month = calculated month. If ME-AU-MEMBER-IP.Start-date > ME-MONTHLY-CUTOFF.cutoff-date then there is an error as the AU member's eligibility starts after the reasonable cutoff date. (CR 13)*

Read ME-AU-MEMBER with person-start-date-9c where person-num = ME-AU-MEMBER-IP.person-num

Escape bottom on change of person-num

Escape top if ME-AU-MEMBER.case-num = I.case-num and ME-AU-MEMBER.cert-period-num = I.cert-period-num /\* Ignore certified record for the in progress record being processed

If #count > 1 /\* Dual eligibility already exists and is certified

Escape bottom

End-if

CALLNAT common module 'Check if dates overlap'

If #date-overlap = true

Add 1 to #count

Assign #isn

```
        Assign #overlapping = true
    End-if
End-read /* ME-AU-MEMBER
If #count > 1
    Error /* Dual eligibility already exists
End-if
If #overlapping
    Get ME-AU-MEMBER with #isn
    Check for dual eligibility allowed for certified AU member
    Technical Note: Read ME-TYPE-CASE and check dual-certs-allowed-indic
    If dual not allowed
        Check for dual eligibility allowed for in progress AU member
        If no dual allowed
            If #ME-AU-MEMBER.close-date (record being modified) = 0 and ME-AU-
            MEMBER.close-date (overlapping record found) = 0
                Prompt Error – dual eligibility in the future will exist.
            else
                Check cutoff dates to determine whether retroactive dual cert.
                Technical Note: Call common module 'Check cutoff dates'
    Technical Note: Send the start date of the AU member being modified (i.e. #ME-AU-MEMBER.start-date.
    Compare the close date of the AU member being modified (#ME-AU-MEMBER) and the close date of the
    overlapping AU member record found (ME-AU-MEMBER). Pass the smallest close date (OTHER THAN 0)
    to the common module
```

## *Check cutoff dates (MECCCDN)*

---

### Input

Start date (of record checking)  
Close date (of overlapping record)

### Output

Dual-elig-allowed (L)  
CXXGNLP (Standard error PDA)

### Mainline processing

/\* If start date is less than cutoff date the eligibility is retro-active and is allowed.

Read ME-MONTHLY-CUTOFF with current year and current month

If current date > ME-MONTHLY-CUTOFF.cutoff-date

Escape top /\* Get next month's cutoff date

Else

Escape bottom /\* we now have the correct cutoff date

End-read

/\* Both the start date and the close date must be before the cutoff date found, otherwise there will be current or future eligibility that overlaps.

If I.start-date < ME-MONTHLY-CUTOFF.cutoff-date

And I.close date < ME-MONTHLY-CUTOFF.cutoff-date

Assign O.dual-elig-allowed = true

Else

Escape routine

## Confirm completion of determination (MECDET1)

If all the certifications that are linked to a determination have been certified, then the worker can choose to complete the determination. If the worker chooses to complete the determination, then the last validations for the certify process are performed.

### Input

MECCKPP (Certify Key processing PDA)

### Output

CXXGNLP (Common error output PDA)

### Local

#Confirm (A1)

### Mainline processing

Prompt the worker with a pop up confirming the completion of the determination.

Do you wish to complete the determination? : \_ (Y/N)

If #Confirm = 'Y'

All certifications must have been certified before the Determination can be completed

*Technical Note: To determine if changes have been made to a cert period, call the common module 'Check if cert changed' for the case. As soon as one certification has been changed and not certified, prompt the error message*

If I.#budgets-exist = true /\* performing scenario 4

1. Validate that each eligible budget has been used. *Technical Note: For each budget number use the budget worksheet CPTC. Read the budget worksheet file with case number/ determination number/ budget determination outcome to get all the eligible budgets and then read the budget worksheet CPTC with case number / determination number / budget number / category type case to check a record exists for that budget. If not all eligible budgets have been used, perform error processing.*
2. Validate that all APTC's have been set to done. If there are any APTC's that have a status of pending, perform error processing. *Technical Note: Find ME-APPL-TYPE-CASE with APTC-STAT-CASE-APPL where appl-stat = pending, case-num = #mecckpp-io.case-num and appl-num = #mecckpp-io.appl-num*

Else /\* Scenario 1

If the Application Status is Pending, but there are no APTC's linked to the Application, set the Appl Status to blank. If there are APTC's linked to the Application and the Determination reason is 'Application set the APTC's to 'D'one else give an error message telling the user to complete the APTC's before completing the Determination

End-if

Complete determination – *Technical Note: Call common module 'Close determination*

Else

Escape program

*Technical Note:*

*If the worker confirms the completion of the determination, the determination status will be set to complete and the determination date will be updated with the current date. If the worker chooses not to complete the determination, it will remain open. This will allow the worker to add further certifications to the determination for the application at a later date.*

## *Check if cert changed (MECCEPN)*

---

This common module checks if any of the certs that have been certified have subsequently been changed.

### Input

Case num  
Elig Deter num

### Output

Common error PDA (CXXXGNLP)  
Changed (L)

### Required fields:

Case number, elig deter num

### Mainline processing

Read ME-ELIG-CERT-PERIOD with case-num, elig-deter-num where case-num = I.case-num and elig-deter-num = I.elig-deter-num

HISTOGRAM ME-CERT-PERIOD-IP with case-changed-indic where case-num = I.case-num, cert-period = ME-ELIG-CERT-PERIOD.cert-period-num and changed-indic = 'M'

If \*counter > 0  
Assign O.Changed = true  
Escape routine

HISTOGRAM ME-AU-MEMBER-IP with case-cert-changed-indic where case-num = I.case-num, cert-period = ME-ELIG-CERT-PERIOD.cert-period-num and changed-indic = 'M'

If \*counter > 0  
Assign O.Changed = true  
Escape routine

HISTOGRAM ME-SEGMENT-IP with case-cert-changed-indic where case-num = I.case-num, cert-period = ME-ELIG-CERT-PERIOD.cert-period-num and changed-indic = 'M'

If \*counter > 0  
Assign O.Changes = true  
Escape routine

HISTOGRAM ME-LASES-CASE-IP with case-cert- changed-indic where case-num = I.case-num, cert-period = ME-ELIG-CERT-PERIOD.cert-period-num and changed-indic = 'M'

If \*counter > 0  
Assign O.Changes = true

## Escape routine

HISTOGRAM ME-LASES-CASE-CHILD-IP with case-cert- changed-indic where case-num = I.case-num, cert-period = ME-ELIG-CERT-PERIOD.cert-period-num and changed-indic = 'M'

If \*counter > 0

Assign O.Changes = true

Escape routine

## *Check if dual eligibility allowed (MECCDAN)*

---

### Input

Case (N13/1:2)

Cert period num (N4/1:2)

In progress (L/1:2)

### Output

Dual allowed (L)

Required fields: Case-num (\*) cert-period-num (\*) and in progress (\*)

### Mainline processing

If I.in-progress (1)

    Read ME-CERT-PERIOD-IP

    Get type case

    If dual-cert-allow-indic = 'Y'

        Assign O.dual-allowed = true

    End-if

Else

    Read ME-CERT-PERIOD

    Get type case

    If dual-cert-allow-indic = 'Y'

        Assign O.dual-allowed = true

    End-if

End-if

## *Validate name for invalid characters (MECNAMN1)*

---

This common module checks first name and last name for invalid characters.

### Input

First Name (A20)

Last Name (A25)

### Output

First Name (A20)

Last Name (A25)

### Initial processing

Reset Output variables.

### Mainline Processing

Check for invalid characters.

Following are the invalid characters:

- ; (semi-colon)
- \_ (underscore)
- \_\_ (two underscores)
- . (dot)
- .. (dot dot)
- ,
- ,, (comma comma)
- (double dash)
- '' (double apostrophe)

## *Validate address and city for invalid characters (MECADRN1)*

---

This common module checks address and city for invalid characters.

### Input

Addr (A35) (3 occurrences)

City (A20)

### Output

Address (A35) (3 occurrences)

City (A20)

### Initial processing

Reset Output variables.

### Mainline Processing

Check for invalid characters.

Following are the invalid characters:

- ; (semi-colon)
- \_ (underscore)
- \_\_ (two underscores)
- .. (dot dot)
- ,, (comma comma)
- (double dash)
- '' (double apostrophe)

## *Check Category Combinations on Cases (MECCCCN)*

---

### Input

Case Num (N13)

Category Code (N3)

Client Num (N13)

### Output

Common error PDA (CXXXGNLP)

Required fields: Case Num, Category Code and Client Num

### Mainline processing

Get ME-CASE using Input Case Num

If Input Category Code = OCS Category /\* 6, 8, 15, 22

    If Case Location Type not = OCS location type

        Assign error = true

        Escape routine

    End If

Else /\* not OCS category

If Case Location Type = OCS location type

    Assign error = true

    Escape routine

    End If

End If

Read ME-CPTCs using Input Case Num

    If close date > 0 and close date < current date

        Escape top /\* bypass closed certs

    End If

    If Input Category Code = OCS category and

    ME-CPTC category = Child related category /\* 3, 5, 13, 16

        Assign error = true /\* cannot add OCS certs to child related cases

        Escape routine

    End If

If Input Category Code = Child related category and

    ME-CPTC category = OCS category

        Assign error = true /\* cannot add child related certs to OCS case

        Escape routine

End If

If Input Category Code = OCS category and

    ME-CPTC category = BHSF category /\* 2, 4, 14, 17, 20

    Get ME-CERT-PERIOD for that CPTC

```
    If ME-CERT-PERIOD client num not = Input Client Num
        Assign error = true      /* OCS client must be the client on previous certs
        Escape routine
    End If
End If
End Read
```

## *Find the User-Id for a Case's Case Worker (MEXRLUN)*

---

This routine will take a case number and find the case worker's user id. A primary use of this routine is finding the correct person to receive workflow messages for a case.

### Input

CASE-NUM (N13)

### Output

SC-USER-ID (A8)

CXXGNLP-OUT

Required fields: Case-num

### Mainline Processing

Find ME-CASE

If #ME-CASE.CASE-LOAD-NUM = 0

    Reset #ME-WORKER.SC-USER-ID

Else

    Move by name ME-CASE to ME-CASE-LOAD

    Find ME-CASE-LOAD

    Move by name ME-CASE-LOAD to ME-WORKER

    Find ME-WORKER

    If #ME-WORKER.SC-USER-ID = ''

        Assign #ME-WORKER.SC-USER-ID = #ME-WORKER.SUPVS-ID

    End-if

End-if

If #ME-WORKER.SC-USER-ID NE ''

    Assign Output = #ME-WORKER.SC-USER-ID

Else

    Find Location

    If #ME-LOC.SUPVS-ID NE ''

        Assign Output = #ME-LOC.SUPVS-ID

    Else

        Assign Output = \*USER

    End-if

End-if

## *Dead Person's Review List Creator (MEXDPRN)*

---

This routine will take a person number/case number and create a tracking workflow item for the person if:

- 1) the person is dead and
- 2) the person is associated with "active" elig Case-num

### Input

Person Number (N13)  
Case Number (N13) /\* Used to determine who workflow is sent to

### Output

None

Required fields: Person Number and Case Number

### Mainline Processing

Get Person record (using object subprogram)

Check Person's Eligibility (using common routine "MEXDPEN")

Get Workflow user id (using common routine "MEXRLUN")

If Person is not dead (Person.Date of Death = 0)

Tracking value = '' /\* to delete any existing

Else

If Person is associated with "active" eligibility (#MEXDPEN-OUT.#ACTIVE-ELIG = True)

Tracking value = "Y" /\* to delete any existing

Else

Tracking value = '' /\* to delete any existing

End-If

End-If

Create tracking table for Group "DEATH" and reason code "DEADPERS" (using common routine "CNTTFVN").

## *Determine a Person's Last Death Closure (MEXDPDN)*

---

This routine will determine if a person has eligibility closed because of "death" and will return the latest death closure close date.

### Input

Person Number (N13)

### Output

#Death Closure (L)

Close Date (N8)

Required fields: Person Number

### Mainline Processing

Retrieve all AU Members for the input person number (using  
PersonNumber/CloseDate9C/StartDate9C/CaseNumber/CertPeriodNumber)

If active AU Member found, get out (there are no more closed AU Members)

If closed AU Member does not have "Death" closure, get next record.

Assign Output.#DeathClosure = True

Assign Output.CloseDate = AU Member.CloseDate

Escape Routine

End-Retrieve

## *Determine If A Person Is Linked To Active Eligibility (MEXDPEN)*

---

This routine will determine if a person is linked/associated with any "active" eligibility.

### Input

Person Number (N13)

### Output

#Active Eligibility (L)

Required fields: Person Number

### Mainline Processing

Retrieve all AU Members for the input person number (using PersonNumber/StartDate9C)

Ignore canceled AU Members (closure code = #CancelFutureElig or #QMBFullRemoval)

Ignore Closed AU Members (close date > 0)

Assign Output.#ActiveEligibility = True  
Escape Routine

End-Retrieve

Retrieve all Cert Periods where the input person number is the Client (using Client/CaseNumber/CertPeriod)

Ignore canceled Cert Periods (closure code = #CancelFutureElig or #QMBFullRemoval)

Ignore Closed Cert Periods (close date > 0)

Assign Output.#ActiveEligibility = True  
Escape Routine

End-Retrieve

## *Build AU Member Eligibility for new CPTC (MECAMEN1)*

---

This routine builds the appropriate AU Member Eligibility records (intersection of AU Member and CPTC) for a new CPTC. It is called from the CPTC object when action = add.

### Input

Case Number (N13)  
Cert Period Number (N4)  
CPTC Number (N4)  
CPTC Start Date (N8)  
CPTC Close Date (N8)  
Closure Code (N2)

Required fields: Case, Cert Period, CPTC numbers and start date

### Mainline Processing

Read AU Members for the input case and cert period

- Call common routine Determine AU Member Eligibility Dates (MECAMEN5) passing in AU Member dates and closure code and input CPTC dates and closure code
- Call common routine Maintain AU Member Eligibility (MECAMEU) with action of add using dates and code output from MECAMEN5

End-Read

## *Build AU Member Eligibility for new AU Member (MECAMEN2)*

---

This routine builds the appropriate AU Member Eligibility records (intersection of AU Member and CPTC) for a new AU Member. It is called from the AU Member object when action = add

### Input

Case Number (N13)  
Cert Period Number (N4)  
AU Member Number (N3)  
Start Date (N8)  
Close Date (N8)  
Closure Code (N2)

Required fields: Case, Cert Period, AU Member numbers and start date

### Mainline Processing

Read CPTC for the input case and cert period

Call common routine Determine AU Member Eligibility Dates (MECAMEN5) passing in input AU Member dates and closure code and CPTC dates and closure code

Call common routine Maintain AU Member Eligibility (MECAMEU) with action of add using dates and code output from MECAMEN5

End-Read

## Rebuild AU Member Eligibility for modified CPTC (MECAMEN3)

This routine rebuilds the appropriate AU Member Eligibility records (intersection of AU Member and CPTC) for a CPTC. It is called from the CPTC object when start date, close date or closure code change.

Input

Case Number (N13)

Cert Period Number (N4)

CPTC Number (N4)

CPTC Start Date (N8)

CPTC Close Date (N8)

Closure Code (N2)

Required fields: Case, Cert Period, CPTC numbers and start date

Mainline Processing

Read AU Member Eligibility for the input case, cert period and CPTC

Ignore if AU Member Eligibility delete indicator = 'Y' /\* logically deleted eligibility

Get AU Member for AU Member Eligibility record

Call common routine Determine AU Member Eligibility Dates (MECAMEN5) passing in AU Member dates and closure code and input CPTC dates and closure code

Call common routine Maintain AU Member Eligibility (MECAMEU) with action of modify using dates and closure code output from MECAMEN5

End-Read

## *Rebuild AU Member Eligibility for modified AU Member (MECAMEN4)*

This routine rebuilds the appropriate AU Member Eligibility records (intersection of AU Member and CPTC) for an AU Member. It is called from the AU Member object when start date, close date or closure code change.

### Input

Case Number (N13)

Cert Period Number (N4)

AU Member Number (N3)

Start Date (N8)

Close Date (N8)

Closure Code (N2)

Required fields: Case, Cert Period, AU Member numbers and start date

### Mainline Processing

Read AU Member Eligibility for the input case, cert period and AU member

Ignore if AU Member Eligibility delete indicator = 'Y' /\* logically deleted eligibility

Get the CPTC for the AU Member Eligibility record

Call common routine Determine AU Member Eligibility Dates (MECAMEN5) passing in input AU Member dates and closure code and CPTC dates and closure code

Call common routine Maintain AU Member Eligibility (MECAMEU) with action of modify using dates and code output from MECAMEN5

End-Read

## *Determine AU Member Eligibility Dates (MECAMEN5)*

This routine determines AU Member Eligibility start and end dates between an AU Members date range and a CPTC date range.

### Input

AU Member Start Date (N8)

AU Member Close Date (N8)

AU Member Closure Code (N2)

CPTC Start Date (N8)

CPTC Close Date (N8)

CPTC Closure Code (N2)

Output

Eligibility Start Date (N8)

Eligibility Close Date (N8)

Eligibility Closure Code (N2)

Required fields: AU Member Start Date and CPTC Start Date

Mainline Processing

If AU Member Close Date = 0

    Move 99999999 to AU Member Close Date

End If

If CPTC Close Date = 0

    Move 99999999 to CPTC Close Date

End If

If CPTC Start Date > AU Member Start Date           /\* find latest start date

    Move CPTC Start Date to Eligibility Start Date

Else

    Move AU Member Start Date to Eligibility Start Date

End If

If CPTC Close Date < AU Member Close Date           /\* find earliest close date

    Move CPTC Close Date to Eligibility Close Date

    Move CPTC Closure Code to Eligibility Closure Code

Else

    Move AU Member Close Date to Eligibility Close Date

    Move AU Member Closure Code to Eligibility Closure Code

End If

If Eligibility Close Date < Eligibility Start Date   /\* no overlap

    Reset Eligibility Start Date, Close Date and Closure Code

End If

If Eligibility Close Date = 99999999

    Reset Eligibility Close Date

End

## *Purge AU Member Eligibility for purged CPTC (MECAMEN6)*

---

This routine purges the appropriate AU Member Eligibility records (intersection of AU Member and CPTC) for a CPTC. It is called from the CPTC object when the CPTC is purged.

### Input

Case Number (N13)

Cert Period Number (N4)

CPTC Number (N4)

Required fields: Case, Cert Period, CPTC numbers

### Mainline Processing

Read AU Member Eligibility for the input case, cert period and CPTC

Ignore if AU Member Eligibility delete indicator = 'Y' /\* logically deleted eligibility

Assign AU Member Eligibility delete indicator = 'Y'

Call common routine Maintain AU Member Eligibility (MECAMEU) with action of modify

End-Read

## *Purge AU Member Eligibility for purged AU Member (MECAMEN7)*

---

This routine purges the appropriate AU Member Eligibility records (intersection of AU Member and CPTC) for an AU Member. It is called from the AU Member object when the AU Member is purged.

### Input

Case Number (N13)

Cert Period Number (N4)

AU Member Number (N3)

Required fields: Case, Cert Period, AU Member numbers

### Mainline Processing

Read AU Member Eligibility for the input case, cert period and AU Member

Ignore if AU Member Eligibility delete indicator = 'Y' /\* logically deleted eligibility

Assign AU Member Eligibility delete indicator = 'Y'

Call common routine Maintain AU Member Eligibility (MECAMEU) with action of modify

End-Read

## *Assign Eligibility Sequence Number (MECPESN1)*

---

**NOTE:** Eligibility Segments keep track of a persons eligibility segments using information from AU Member, CPTC and sanctions. They define a period of time during which a MEDS person actually receives Medicaid, and are sent to MMIS.

MEDS calculates the Eligibility Segments start and close dates by taking the AU Member Eligibility and removing the periods of AU Member sanctions. When eligibility is added, start dates, close dates or closure codes are modified, or sanctions are changed Eligibility Segments are updated, Because MMIS wants to maintain the original dates of eligibility rather than updating start and end dates when more eligibility is added for an AU Member we add new eligibility segments.

MECPESN1 assigns the next eligibility sequence number for a person. It is called from the Eligibility Segment object (MECPESU) when an eligibility segment is added.

### Input

Person Number (N13)

MMIS Run Number (N8)

### Output

Eligibility Segment Sequence Number (N4)

Required fields: Person Number, MMIS Run Number

### Mainline Processing

Find Person for input person number

Assign Output Eligibility Segment Number = Person Next Eligibility Segment Number

Add 1 to Person Next Eligibility Segment Number

Assign Person MMIS Run Number = input MMIS Run Number

Update Person

## *Add Eligibility Segments (MECPESN2)*

---

This routine adds the appropriate eligibility segments (intersection of AU Member Eligibility and sanctions) for a new AU Member eligibility record. It is called from the AU Member Eligibility object (MECAMEU) when an AU Member eligibility record is added.

### Input

AU Member PDA

AU Member Eligibility PDA

Required fields: AU Member Eligibility Start Date

## Mainline Processing

Call common routine Build New Eligibility Segments (MECPESN3) to apply sanction periods to eligibility passing in AU Member and AU Member Eligibility PDAs.

Assign Eligibility Segment case, cert period, CPTC and au member numbers from input AU Member Eligibility

Assign Eligibility Segment person number from input AU Member person number

For 1 to MECPESN3 Number of Segments

Assign Eligibility Segment start date, close date and closure code from MECPESN3 output

Call common routine Maintain Eligibility Segment (MECPESU) with action of Add

End For

---

## *Build New Eligibility Segments (MECPESN3)*

---

This routine builds the appropriate eligibility segments (intersection of AU Member Eligibility and sanctions) for a new or modified AU Member eligibility record. It is called from the Add Eligibility Segments routine (MECPESN2) and the Rebuild Eligibility Segments routine (MECPESN7).

### Input

AU Member PDA

AU Member Eligibility PDA

### Output

Number of Segments

Eligibility Segment Sequence Number (1:10) /\* UNUSED

Start date, close date, closure code (1:10)

Required fields: none

### Mainline Processing

If AU Member Number of Sanctions = 0

Assign output Number of Segments = 1

Assign output start date, close date, closure code (1) = AU Member Eligibility values

Escape routine

End If

For I = 1 to AU Member Number of Sanctions

Call common routine Determine Date Overlap (MECPESN5) to determine the overlap between the sanction and AU Member Eligibility passing in sanction dates and AU Member Eligibility dates.

If no overlap

    Ignore sanction

Else

    Call common routine Remove Overlap (MECPESN6) to remove the sanction overlap from the AU Member eligibility passing in AU Member eligibility dates and the overlap dates returned from MECPESN5

    Move output from MECPESN6 to MECPESN3 output.

End If

End For

Remove empty slots from MECPESN3 output

Sort MECPESN3 output in start date order

## *Load Current Eligibility Segments (MECPESN4)*

---

This routine retrieves and loads the appropriate current eligibility segments into an array for an AU Member eligibility record. It is called from the Rebuild Eligibility Segments routine (MECPESN7).

Input

    Case Number (N13)

    Cert Period Number (N4)

    CPTC Number (N4)

    AU Member Number (N3)

Output

    Old Number of Segments

    Old Eligibility Segment Sequence Number (1:25)

    Old Start date, close date, closure code (1:25)

Required fields: case, cert period, CPTC and AU Member numbers

Mainline Processing

    Read Eligibility Segment with Case Cert CPTC AU Member number from input

        If Eligibility Segment Deleted Indicator = 'Y'

            Escape top /\* ignore deleted segments

        End if

        Load start date, close date, closure code, eligibility segment sequence number to output array

End Read

Sort MECPESN3 output in start date order

## *Determine Date Overlap (MECPESN5)*

---

This routine determines the period of overlap between AU Member Eligibility dates and sanction dates. It is called from the Build New Eligibility Segment routine (MECPESN3) only.

### Input

AU Member Eligibility Start Date (N8)

AU Member Eligibility Close Date (N8)

Sanction Start Date (N8)

Sanction Close Date (N8)

### Output

Overlap Start Date (N8)

Overlap Close Date (N8)

Required fields: AU Member Eligibility Start Date and Sanction Start Date

### Mainline Processing

If AU Member Eligibility Close Date = 0

    Move 99999999 to AU Member Eligibility Close Date

End If

If Sanction Close Date = 0

    Move 99999999 to Sanction Close Date

End If

If Eligibility Start Date > Sanction Start Date                   /\* find latest start date

    Move Eligibility Start Date to Overlap Start Date

Else

    Move Sanction Start Date to Overlap Start Date

End If

If Eligibility Close Date < Sanction Close Date               /\* find earliest close date

    Move Eligibility Close Date to Overlap Close Date

Else

```

    Move Sanction Close Date to Overlap Close Date
End If
If Overlap Close Date < Overlap Start Date /* no overlap
    Reset Overlap Start Date and Close Date
End If
If Overlap Close Date = 99999999
    Reset Overlap Close Date
End

```

## *Remove Sanction Overlap from Eligibility (MECPESN6)*

This routine removes a period of ineligibility (sanction) from a period of eligibility to give true eligibility segment dates. It is called from the Build New Eligibility Segment routine (MECPESN3) only.

### Input

```

    AU Member Eligibility Start Date (N8)
    AU Member Eligibility Close Date (N8)
    AU Member Eligibility Closure Code (N2)
    Overlap Start Date (N8)
    Overlap Close Date (N8)

```

### Output

```

    Number of Segments
    Segment Start Date (1:2)
    Segment Close Date (1:2)
    Segment Closure Code (1:2)

```

Required fields: AU Member Eligibility Start Date and Overlap Start Date

### Mainline Processing

```

If AU Member Eligibility Start Date = Overlap Start Date
    If AU Member Eligibility Close Date = Overlap Close Date
        Assign output Number of Segments = 0 /* sanction applies for entire eligibility period
    Else
        Assign output Number of Segments = 1

```

Assign output Segment Start Date (1) = Overlap Close Date + 1 /\* day after sanction ends  
Assign output Segment Close Date (1) = AU Member Eligibility Close Date  
Assign output Segment Closure Code (1) = AU Member Eligibility Closure Code  
End If  
Else /\* start dates are not the same  
Assign output Number of Segments = 1  
Assign output Segment Start Date (1) = AU Member Eligibility Start Date  
Assign output Segment Close Date (1) = Overlap Start Date - 1 /\* day before sanction starts  
Assign output Segment Closure Code (1) = 57 /\* code meaning closed due to sanction  
If AU Member Eligibility Close Date not = Sanction Code Date  
Assign output Number of Segments = 2  
Assign output Segment Start Date (2) = Overlap Close Date + 1 /\* day after sanction ends  
Assign output Segment Close Date (2) = AU Member Eligibility Close Date  
Assign output Segment Closure Code (2) = AU Member Eligibility Closure Code  
End If  
End If

## Rebuild Eligibility Segments (MECPESN7)

This routine rebuilds the appropriate eligibility segments (intersection of AU Member Eligibility and sanctions) for a modified AU Member eligibility record or a modified sanction. It is called from AU Member Eligibility object (MECAMEU) when start date, close date or closure code change, the Add Eligibility Segments routine (MECPESN2) and the Find Eligibility Segments routine (MECPESNA), which is called from the AU Member object when sanctions change.

### Input

AU Member Eligibility PDA

AU Member PDA

Required fields: none

### Mainline Processing

Call common routine Load Current Eligibility Segments (MECPESN4) passing in AU Member Eligibility fields to get the current Eligibility Segments

Call the common routine Build New Eligibility Segments (MECPESN3) passing in AU Member Eligibility and AU Member fields to get the new eligibility dates

/\* now match the old segments to the new dates. When adding eligibility a new eligibility segment is  
/\* created rather than adjusting the old one. This is so MMIS can maintain a history of eligibility.

For I = 1 to current number of segments (output from MECPESN4)

For J = 1 to new number of segments (output from MECPESN3)

Call common routine Determine Segment Overlap (MECPESN8) passing in one occurrence of  
the start and close dates from MECPESN4 and MECPESN3

If no overlap

Escape top /\* get next new segment from MECPESN3

End If

**If new start date (from MECPESN3) < old start date (from MECPESN4)**

/\* eligibility has been moved back – add a new segment for the new period

Add segment with date from new start date to old start date - 1

Assign Eligibility Segment Closure Code = 65 /\* code telling MMIS this is due to a date  
change

/\* adjust remaining new eligibility array (from MECPESN3) to reflect eligibility still to be  
/\* added and do this comparison again

Assign new start date = old start date

End If

**If new start date (from MECPESN3) = old start date (from MECPESN4) and new  
close date (from MECPESN3) = old close date (from MECPESN4)**

/\* no change in dates – modify the segment for new closure code

Modify Eligibility Segment Closure Code to new closure code

End If

**If new start date (from MECPESN3) = old start date (from MECPESN4) and new  
closure code = 48, 89 or 90 and old closure code = 48, 89 or 90 /\* death closures**

/\* date of death change only – modify the segment for new close date

Modify Eligibility Segment Close Date = new close date

Assign Eligibility Segment Closure Code = new closure code

End If

**If new start date (from MECPESN3) = old start date (from MECPESN4) and new  
close date > old close date**

/\* more eligibility has been added to the end

/\* add a segment for the original period that matches

Modify Eligibility Segment Closure Code = 65 /\* date change

/\* this could be an extension due to date of death being removed. If so set the segment  
/\* close date to be the end of the month (MMIS cannot handle middle of month dates)

If old closure code = 48, 89 or 90 /\* death

Assign Eligibility Segment Closure Code = 45 /\* undead code for MMIS

Assign Eligibility Segment Close Date = last day of month of old close date

End If

/\* adjust remaining new eligibility array (from MECPESN3) to reflect eligibility still to be  
/\* added and do this comparison again

Set new start date = Eligibility Segment Close Date + 1

End If

**If new start date (from MECPESN3) = old start date (from MECPESN4) and new  
close date < old close date**

/\* eligibility has been removed from the end

/\* update segment for the period that matches

Modify Eligibility Segment Close Date = new close date and closure code = new closure  
code

End If

**If new start date (from MECPESN3) > old start date (from MECPESN4) and new  
close date > old close date**

/\* more eligibility has been added to the end but eligibility taken away from the  
beginning

/\* modify the segment for the new later start

Modify Eligibility Segment Start Date = new start date, close date = old close date and  
closure code = 65 /\* date change

/\* adjust remaining new eligibility array (from MECPESN3) to reflect eligibility still to be  
/\* added and do this comparison again

Set new start date = Eligibility Segment Close Date + 1

End If

**If new start date (from MECPESN3) > old start date (from MECPESN4) and new  
close date <= old close date**

/\* eligibility taken away from the beginning and the end

```
        /* modify the segment for the new later start and earlier end

        Modify Eligibility Segment Start Date = new start date, close date = new close date and
        closure code = new closure code

        End If
    End For /* new segments (MECPESN3)

    /* If there are old segments (MECPESN4) unmatched to new segments (MECPESN3) delete them
    If no match
        Modify Eligibility Segment deleted indicator = Y
    End If
End For /* old segments (MECPESN4)

/* if there are new segments not matched to old ones add them
For J = 1 to new number of segments (output from MECPEsn3)
    If eligibility sequence number (J) = 0 /* unmatched
        Add a new segment with start date, close date, closure code =values from MECPEsn3
    End If
End For

For each Eligibility Segment being added or changed
    Call common routine Modify Eligibility Segment (MECPESU) passing in the Eligibility Segment
    fields
End For
```

## *Determine Segment Overlap (MECPESN8)*

---

This routine determines the period of overlap between 2 dates ranges – existing segment and new segment. It is called from the Rebuild New Eligibility Segment routine (MECPESN7) only.

### Input

Old Start Date (N8)

Old Close Date (N8)

New Start Date (N8)

New Close Date (N8)

Output

Overlap Start Date (N8)

Overlap Close Date (N8)

Required fields: Old and new start dates

Mainline Processing

If Old Close Date = 0

    Move 99999999 to Old Close Date

End If

If New Close Date = 0

    Move 99999999 to New Close Date

End If

If Old Start Date > New Start Date                   /\* find latest start date

    Move Old Start Date to Overlap Start Date

Else

    Move New Start Date to Overlap Start Date

End If

If Old Close Date < New Close Date               /\* find earliest close date

    Move Old Close Date to Overlap Close Date

Else

    Move New Close Date to Overlap Close Date

End If

If Overlap Close Date < Overlap Start Date /\* no overlap

    Reset Overlap Start Date and Close Date

End If

If Overlap Close Date = 99999999

    Reset Overlap Close Date

End

## *Purge Eligibility Segments (MECPESN9)*

---

This routine purges the appropriate eligibility segments (intersection of AU Member Eligibility and sanctions) for a purged AU Member eligibility record. It is called from the AU Member Eligibility object (MECAMEU) when an AU Member eligibility record is purged.

### Input

Case, Cert Period, CPTC, AU Member numbers

Required fields: Case, Cert Period, CPTC, AU Member numbers

### Mainline Processing

Read Eligibility Segments for input case, cert, CPTC and AU member numbers

    If deleted indicator = Y

        Escape top

    End If

    Assign Eligibility Segment Deleted Indicator = Y

    Call common routine Maintain Eligibility Segment (MECPESU) with action of Modify

End Read

## *Find Eligibility Segments (MECPESNA)*

---

This routine finds and updates the appropriate eligibility segments (intersection of AU Member Eligibility and sanctions) for a modified AU Member record. It is called from the AU Member object (MECAUMU) on change of sanctions.

### Input

AU Member PDA

Required fields: Case, Cert Period, AU Member numbers

### Mainline Processing

Read AU Member Eligibility input case, cert and AU member numbers

    Call common routine Rebuild Eligibility Segment (MECPES7) passing in AU Member Eligibility fields and AU Member fields

End For

## *Retrieve Latest Completed Budget for Cert (MECRLBN)*

---

This routine searches through the budgets that are linked to a Cert and returns the latest "Completed" and eligible budget. The latest budget for a cert is one that is linked to a certification. It is of the same type case and for the same determination.

### **Parameters**

#### *Input*

CASE-NUM (N13)  
CERT-PERIOD-NUM (N4)  
CPTC-NUM (N4)  
CAT-TYPE-CASE (N6)  
ELIG-DETER-NUM (N6)

#### *Output*

BUD-NUM (N3)  
BUD-DATE (N6)  
#ME-BUDGET-WRKSHT View (MEBWRKFP)  
Standard Error fields

### **Processing**

- Reset all output parameters.
- Using a suitable superdescriptor, find and return the latest "Completed" (*field BUD-STATUS*) and "Eligible" (*field BUD-DETER-OUTCOME-INDIC*) Budget Worksheet linked to the CPTC specified by the input parameters. The latest Budget Worksheet is the one with the latest Budget Date. If multiple Budget Worksheets with the latest same Budget Date are linked to the CPTC, return the one with the highest matching Determination number.

#### *Technical Notes:*

- *Budget Worksheets are linked to CPTCs via the ME-BUDGET-WRKSHT-CPTC file.*

## *Retrieve Latest Uncertified Budget for Cert (MECRUBN)*

---

This routine searches through the budgets that are linked to a Cert and returns the latest "Completed" budget. The latest budget for a cert is one that is linked to a certification that is not certified/decided on yet. It is of the same type case and for the same determination.

### **Parameters**

#### *Input*

CASE-NUM (N13)  
CERT-PERIOD-NUM (N4)  
CPTC-NUM (N4)  
CAT-TYPE-CASE (N6)  
ELIG-DETER-NUM (N6)

#### *Output*

BUD-NUM (N3)  
BUD-DATE (N6)  
#ME-BUDGET-WRKSHT View (MEBWRKFP)  
Standard Error fields

### **Processing**

- Reset all output parameters.
- Using a suitable superdescriptor, find and return the latest "Completed" (*field BUD-STATUS*) and "Eligible" (*field BUD-DETER-OUTCOME-INDIC*) Budget Worksheet linked to the CPTC specified by the input parameters. The latest Budget Worksheet is the one with the latest Budget Date. If multiple Budget Worksheets with the latest same Budget Date are linked to the CPTC, return the one with the highest matching Determination number. The Budget CPTC link also must be CERTIFIED-LINK NE 'Y'.

#### *Technical Notes:*

- *Budget Worksheets are linked to CPTCs via the ME-BUDGET-WRKSHT-CPTC file.*

## *Determine Indicated Renewal Level for Cert (MECDRLN)*

---

This subprogram determines the Indicated Renewal Level applicable to a particular Cert. This module implements the business rules specified on the "Medicaid Eligibility Program – Renewal Guide" specified as "Indicators" for Level 1 – "Exparte" Renewal.

**Note:** The Renewal Levels are defined in an LDA that is spec'ed in the SIR924 Part 2 spec.

### **Parameters**

#### *Input*

CASE-NUM (N13)  
CERT-PERIOD-NUM (N4)  
CPTC-NUM (N4)

#### *Output*

RENEWAL-LEVEL (N1)  
RENEWAL-RULE (N2)  
DETER-LEVEL-DESC (A79)  
Error Fields

### **Processing**

- Reset all output parameters.
- Validate that input parameters CASE-NUM, CERT-PERIOD-NUM, and CPTC-NUM parameters are populated.
- Retrieve the Certification Period and CPTC. Return an error if the Cert Period or CPTC can't be found.  
*Technical Note: Call the appropriate object subprogram to retrieve them.*
- Retrieve the Type Case identified by the CPTC. *Technical Note: Call the appropriate object subprogram to retrieve it.*
- Set Local.#CURRENT-REDET-CODE to ME-CPTC.REDET-CODE.

Return the Output.RENEWAL-RULE based on the rule we are trying to process at the moment.

Populate the Output.DETER-LEVEL-DESC with a description of the rule used to determine the renewal level. If an error occurred populate Output.DETER-LEVEL-DESC with an appropriate message to indicate what information was being sought to apply the rule.

*/\* The rules listed here correspond to the rules as described in section "Error! Reference source not found." (page Error! Bookmark not defined.).*

*/\* Rule 1: If Renewal Code is '2', determine the prior value.*

- If the CPTC's current Renewal Code is 2:
  - Determine the prior value of REDET-CODE for the current cert. Call module "Determine Prior Value of Renewal Code for Cert (MECPRCN)" (page 100) passing in the Input CASE-NUM, CERT-PERIOD-NUM, CPTC-NUM, and Local.#CURRENT-REDET-CODE.
  - If the output parameters from the call indicate that the prior value of REDET-CODE was found:

- Set Local.#CURRENT-REDET-CODE to MECPRCN-Output.PRIOR-REDET-CODE.
- Else
  - Set Local.#CURRENT-REDET-CODE to zero and continue with following steps. .
- End-If
- End-If

*/\* Rule 2: If review or other action pending, don't determine a level.*

- If the Local.#CURRENT-REDET-CODE is 4, 5, or 9:
  - Set the Output.RENEWAL-LEVEL to zero (no value) and escape this routine.
- End If.

*/\* Rule 3: Renewal Codes 8, => Level '1' Exparte.*

- If the Local.#CURRENT-REDET-CODE is 8:
  - Set the Output.RENEWAL-LEVEL to '1' (Exparte) and escape this routine.
- End If.

*/\* Rule 4: Renewal Codes 10, 12, 13, => Level '2' Regular.*

- If the Local.#CURRENT-REDET-CODE is 10, 12, or 13:
  - Set the Output.RENEWAL-LEVEL to '2' (Regular) and escape this routine.
- End If.

*/\* Rule 5: If LAMI Closure, Level '1' Exparte.*

- If the Local.#CURRENT-REDET-CODE is '6':
  - Set the Output.RENEWAL-LEVEL to '1' (Exparte) and escape this routine.
- End-If.

*/\* Rule 6: If SSA Closure, Level '1' Exparte.*

- If the Local.#CURRENT-REDET-CODE is '7':
  - Set the Output.RENEWAL-LEVEL to '1' (Exparte) and escape this routine.
- End-If.

*/\* Rule 7: If an active Food Stamp certification exists for the Client, Level '1' (Exparte).*

- If the Type Case's "Exparte Renewal – Always if Active FS" field = 'X':
  - Call module MECRFSN, passing in the case number and cert period number, to determine if a current and active Food Stamp Cert exists.

- If MECRFSN-Output.CURRENT-FOOD-STAMP-CERT-FOUND = TRUE and MECRFSN-Output.ACTIVE-FS-CERT-INDICATOR = YES:
  - Set the Output.RENEWAL-LEVEL to '1' (Exparte) and escape this routine.
- End-If.
- End-If.

*/\* Rule 8: If the "Exparte Renewal – Frequency" field on TCAS is blank, Level '2' (Regular).*

- If the "Exparte Renewal - Frequency" field on TCAS is blank:
  - Set the Output.RENEWAL-LEVEL to '2' (Regular) and escape this routine.
- End-If

*/\* Rule 9: If the "Exparte Renewal – Frequency" field on TCAS indicates "Odd years only" and the prior renewal was Exparte.*

- If the "Exparte Renewal – Frequency" field on TCAS indicates "Odd years only" and the Renewal Code on the Cert is '11' (Exparte):
  - Set the Output.RENEWAL-LEVEL to '2' (Regular) and escape this routine.
- End-If

*/\* Rule 10: If the Type Case that income must be stable to allow Exparte.*

- If the TCAS field "Exparte Renewal – Check Stable Income" is 'X' and budgets are required by the Type Case:
  - Retrieve the Certification's latest "Completed" budget. This is the "completed" Budget with the latest Budget Date that is linked to the CPTC.
  - Call the module "Check Stable Income (MECCSIN)" (page 96) passing in the key fields of the budget.
  - If MECCSIN-Output.EARNED-INCOME-FOUND (there is "Earned Income" on the budget):
    - Set the Output.RENEWAL-LEVEL to '2' (Regular) and escape this routine.
  - End-If
  - If NOT MECCSIN-Output.ALL-INCOME-STABLE:
    - Set the Output.RENEWAL-LEVEL to '2' (Regular) and escape this routine.
  - End-If
- End-If.

*/\* Rule 11: If the Type Case indicates that the presence of a Community Spouse prevents Exparte.*

- If the TCAS field "Exparte Renewal – Check no Comm Spouse" is 'X' and budgets are required by the Type Case:
  - Retrieve the Certification's latest "Completed" budget if not previously retrieved.

- If there is a Community Spouse:

*Technical Note: To determine if there is a community spouse, call the object-subprogram of the ME-CASE-BUDGET-INFO file MEBCBIU, passing in the CASE-NUM and the BUDGET-DATE of the latest budget. There is a community spouse if field ANSWER-Q7 = 'Y'.*

- Set the Output.RENEWAL-LEVEL to '2' (Regular) and escape this routine.
- End-If
- End-If

*/\* Rule 12: If resources limits apply, and the client's resources are within \$500 of the Eligibility limit, cannot Exparte.*

- If the Type Case's "Resource Limits" field has a value:
  - Retrieve information about known resources of the client. Call module "Retrieve Resource Information (MECRRIN)" (page 98) passing in the Case Number and the Category/Type Case.
  - If MECRRIN-Output.RESOURCES-FOUND = TRUE:
    - Retrieve the currently effective Standard Value for the Standard Value Type specified in the Type Case's "Resource Limits" field. If no such Standard Value is found, return an error "No Standard Values defined for Standard Type XXXXXXXX" (where XXXXXXXX is the Standard Value Type.) The STND-VAL field of the first occurrence of the VALUE-INFO PE contains the "Single" limit, and the STND-VAL field of the second occurrence of the VALUE-INFO PE contains the "Couple" limit. Keep these as local variables #SINGLE-LIMIT, and #COUPLE-LIMIT respectively.
    - Retrieve the currently effective Standard Value for Standard Type "RSRCCUSH" (Resource Cushion), which defines the cushion required between the actual resources and the applicable limit in order to allow Exparte renewal. The STND-VAL field of the first occurrence of the VALUE-INFO PE contains the cushion amount. Keep this as local variable #CUSHION-AMOUNT.
    - If MECRRIN-Output.SINGLE-OR-COUPLE indicates "Single":
      - Subtract #CUSHION-AMOUNT from #SINGLE-LIMIT giving Local.#EXPARTE-MAX-RSRC.
    - Else
      - Subtract #CUSHION-AMOUNT from #COUPLE-LIMIT giving Local.#EXPARTE-MAX-RSRC.
    - End-If
    - If MECRRIN-Output.TOTAL-RESOURCE-AMT > Local.#EXPARTE-MAX-RSRC:
      - Set the Output.RENEWAL-LEVEL to '2' (Regular) and escape this routine.
    - End-If.
  - End-If.
- End-If.

*/\* Rule 13: Anything left is Level '1' (Exparte).*

- Set the Output.RENEWAL-LEVEL to '1' (Exparte) and escape this routine.

End of Processing

## *Retrieve Active Food Stamp Certification (MECRFSN)*

---

This routine retrieves information about whether there is an "active" Food Stamp certification for a Case. The Food Stamp information is held on the Person record for the client of a certification. MEDS will consider Food Stamp information updated in the last 2 months current for use for on-cycle renewals and for late renewals, but not for future off-cycle renewals.

### **Parameters**

#### *Input*

Case Number (N13)

Cert Period Number (N4)

#### *Output*

CURRENT-FS-CERT-EXISTS (L)

ACTIVE-FS-CERT-INDIC (A1)

LAST-FS-CHECKED-DATE (N8)

### **Processing**

Reset all output parameters.

Find MEDS Certification Period Type Case with Case Number and Cert Period Number = input parameters

End Find

If input Person Number = 0

    Find MEDS Certification Period with Case Number and Cert Period Number = input parameters

    End Find

End If

Find MEDS Person with Person Number = Client Number from Certification OR input Person Number

    /\* For on-cycle renewals the food stamp information must be current within last 2 months

    If the Renewal Date of the Certification Period Type Case is within 2 months of cut-off and the Food Stamp Certification Last Date Checked is within 2 months of the current date

        Assign output Current FS Cert Exists flag = true

        Assign output Active FS Cert Indic = Active FS Cert Indic from Me Person

        Assign output FS Last Checked Date = FS Last Checked Date from Me Person

    End If

    /\* For overdue renewals the food stamp information must be current within last 2 months

    If the Renewal Date of the Certification Period Type Case < current date and the Food Stamp Certification Last Date Checked is within 2 months of the current date

```
    Assign output Current FS Cert Exists flag = true
    Assign output Active FS Cert Indic = Active FS Cert Indic from Me Person
    Assign output FS Last Checked Date = FS Last Checked Date from Me Person
End If
/* For off-cycle renewals the LAMI system must be checked manually
End Find
```

## Check Stable Income (MECCSIN)

---

The subprogram determines whether or not all the income on a specified budget is considered "stable". In order to make this determination it looks at all the IU members on the budget and checks to see if any of them has income of a type that is not considered "stable". If all income types on the budget are "stable", this routine returns TRUE in the ALL-INCOME-STABLE parameter, otherwise it returns FALSE. It also checks if any earned income was found. If so, this routine returns TRUE in the EARNED-INCOME-FOUND parameter, otherwise it returns FALSE.

An Income Type is considered "stable" if the "Stable Income" field on the Income Type Maintenance screen is set to 'Y'.

*Technical Note: If the calling program has already retrieved the desired Budget Worksheet record, it should only pass the budget date. It is the only additional piece of information required from the budget worksheet view.*

### Parameters

#### Input

CASE-NUM (N13)  
ELIG-DETER-NUM (N4)  
BUD-NUM (N3)  
BUD-DATE (N6)

#### Output

ALL-INCOME-STABLE (L)  
EARNED-INCOME-FOUND (L)  
Standard Error Fields

### Processing

- Reset all output parameters.
- Validate that the input parameters are populated.
- Set Output.ALL-INCOME-STABLE to TRUE.
- Retrieve all Budget Worksheet Applicants that are IU Member on the budget. *Technical Note: Read by super CASE-DETER-BUD-IU-INDIC.* For each one:
  - If the Income Indicator on the Budget Worksheet Applicant is 'Y' (*Technical Note: field INCM-INDIC*):
    - Retrieve the Person Month Income information for the Person identified on the Budget Worksheet Applicant and the Budget's "Budget Date". *Technical Note: Find the ME-PERSON-MONTH-INCOME using the PERS-MONTH super.*
    - *If a Person Month Income record does not exist. Set Output.ALL-INCOME-STABLE to false and exit this routine. It is assumed that this is an old budget and PIES previously did not exist.*
    - Iterate through the collection of Unearned Income information (*Technical Note: PE UNEARNED-INCOME-INFO*) on the Person Month Income record, and for each entry:

- Retrieve the Income Type code specified on the Unearned Income Information entry.  
*Technical Note: using the appropriate object-subprogram.*
- If the Stable Income field on the Income Type Code is not 'Y':
  - Set Output.ALL-INCOME-STABLE to FALSE and exit this loop.
- End-If
- End Iteration.
- Now check if any earned income exists for this IU member.
  - If earned income exists set the Output. EARNED-INCOME-FOUND to TRUE.
- If Output.ALL-INCOME-STABLE = FALSE and Output.EARNED-INCOME-FOUND = TRUE
  - If both set escape this routine.
- End-If.
- End Retrieval.

End of Processing

## *Retrieve Resource Information (MECRRIN)*

---

This subprogram retrieves information about resources, including amount and whether it is "Single" or "Couple".

The information must come from MAS.

It is not clear whether the information from MAS required in order for this interface to operate will be available in a timely manner. All we may get from MAS is whether or not there are any Resources, or perhaps a list of the Resource Types. The total countable resource amount is not currently stored by MAS, but is recalculated when a report is produced. UNO do not think that they can supply us with this amount. For more information see "**Error! Reference source not found.**" (page **Error! Bookmark not defined.**).

### Parameters

#### *Input*

CASE-NUM (N13)

CATEGORY (N3)

The Type Case Category.

TYPE-CASE (N3)

#### *Output*

RESOURCES-FOUND (L)

TOTAL-RESOURCE-AMT (N9.2)

SINGLE-OR-COUPLE (A1)

This parameter returns either 'S' – Single, or 'C' – Couple.

### Processing

- Reset all output parameters.
- Determine which Resource Group (as defined in the **Error! Reference source not found.** (page **Error! Bookmark not defined.**), the Type Case falls into, if any. (Assuming SIR-xxx is implemented by the time this is implemented, it should be used to make this determination). Escape this routine if the Type Case doesn't belong to one of the Resource Groups.
- READ (1) the ME-MAS-RESOURCE-INTFC file in Descending sequence starting with Input.CASE-NUM and Input.RSRC-GROUP
  - If the Case Number and Resource Group don't match the input values:
    - Escape Bottom.
    - End-If
    - Set Output.RESOURCES-FOUND to TRUE.
    - If the QTY-PEOPLE-RSRC-UNIT = 1:
      - Set output.SINGLE-OR-COUPLE = 'S'
    - Else

- Set output.SINGLE-OR-COUPLE = 'C'
- End-If
- Set other output parameters from the record.
- End-Find.

End of Processing

*Technical Note: It may make sense to just generate an "Object Subprogram" for the ME-MAS-RESOURCE-INTFC file, and call it with a "Display" action when a call to this subprogram is required. However such a subprogram will never be called for update, since this file is only updated by MAS.*

## *Determine Prior Value of Renewal Code for Cert (MECPRCN)*

---

This module looks back through the ME-CPTC-LOG file to determine when the REDET-CODE of the CPTC was last changed, and determine the prior value of the code.

### **Parameters**

#### *Input*

CASE-NUM (N13)  
 CERT-PERIOD-NUM (N4)  
 CPTC-NUM (N4)  
 CURRENT-REDET-CODE (N2)  
     The current value of the CPTC's REDET-CODE

#### *Output*

PRIOR-VALUE-FOUND (L)  
     Set to TRUE if a prior value is found, FALSE otherwise.

PRIOR-REDET-CODE (N2)  
     Set to the value the REDET-CODE of the CPTC had immediately prior to receiving its current value.

DATE-REDET-CODE-CHANGED (N8)  
     This is the date that the REDET-CODE was changed from the prior value to the current value.

TIME-REDET-CODE-CHANGED (N7)  
     This is the time that the REDET-CODE was changed from the prior value to the current value.

### **Processing**

- Reset Output parameters.
- Validate that all input parameters are populated.
- Read ME-CPTC-LOG Descending WITH LOG-BY-ENTRY starting from the Input CASE-NUM, CERT-PERIOD-NUM, CPTC-NUM and high values for the UPDATE-DT and UPDATE-TM fields.
  - Escape Bottom if the Case, Cert, CPTC part of the key is not equal the Input Case, Cert, CPTC.
  - If Input.CURRENT-REDET-CODE not equal ME-CPTC-LOG.REDET-CODE
    - Set Output.PRIOR-REDET-CODE to ME-CPTC-LOG.REDET-CODE
    - Set Output.PRIOR-VALUE-FOUND = TRUE
    - Escape Bottom (Read.)
  - End-If

/\* The date and time that the change occurred is on the earliest log record that has the current value of REDET-CODE.

- Set Local.#SAVE-UPDATE-DT to ME-CPTC-LOG.UPDATE-DT.
- Set Local.#SAVE-UPDATE-TM to ME-CPTC-LOG.UPDATE-TM.

- End-Read.
- If prior value found
  - Set Output.DATE-REDET-CODE-CHANGED to Local.#SAVE-UPDATE-DT.
  - Set Output.TIME-REDET-CODE-CHANGED to Local.#SAVE-UPDATE-TM.

End of Processing

## *LASES Case Validation for Cert (MECLCVN)*

---

This module will contain the logic to determine if a LASES case referral is required for a certification. Depending on whether a minor or adult is being certified an error or pop-up may result.

### **Parameters**

In

PERSON-NUMBER (N13)

AU-MEMBER-NUMBER (N3)

RELATION-TO-CLIENT-CODE (A2)

CLIENT (L)

MINOR (L)

AT LEAST ONE LCHLD FOUND (L)

THIS AUMEM LCHLD FOUND (L)

Out

REFERRAL-REQUIRED (A1)

MANDATORY-REFERRAL (L)

CLIENT-NUM (N13)

CASE-NUM (N13)

CERT-PERIOD-NUM (N4)

CAT-TYPE-CASE (N6)

### **Processing**

Minor Being Certified

13-01 Type Cases

Type cases 13-01 (LIFC Basic) with an approval code = 2 (absence) and a minor exists and deprivation is due to absence a referral is mandatory.

For Client found on same cert

- This is a mandatory referral
- Provide an error message if a LASES case does not exist for the minor child.

Type cases 13-01 (LIFC Basic) with an approval code = 2 (absence) and a minor exists but we did not find the adult on the same cert. If the client is the child's parent, try to locate the client on another cert or case. For these type cases where deprivation is due to absence a referral is mandatory.

If the relationship code is not equal to the value 3, child, then this referral is not Mandatory. The pop-up window discussed in the section below will be displayed.

If the child's relationship code = 3, then the client is the parent.

#### Check for Client/Parent in Another Cert on Same Case

- Read Au Member by person number matching the client's number.
- If the au member is closed do not consider the cert.
- There are type cases that are restricted. These type cases as well as OCS type cases cannot be used when searching for the adult certification. See attached tables for the list of type cases.
- If an active client is found on a different cert then a LASES referral is required.
- This is a mandatory referral
- Provide an error message if a LASES case does not exist for the minor child.

•

#### Check for Client/Parent in Different Cases

- Use the Client from the minor certification and look for this client on another cert. We already know this client is the parent based on the child's relationship code.
- Read Au Member by person number matching the client's number.
- If the au member is closed do not consider the cert.
- There are type cases that are restricted. These type cases as well as OCS type cases cannot be used when searching for the adult certification. See attached tables for the list of type cases.
- If an active client is found on a different cert/case then a LASES referral is required.
- This is a mandatory referral
- Provide an error message if a LASES case does not exist for the minor child.

#### Other Type Cases

The following section constitutes edits for all type cases not on the restricted list. For these type cases we have not established deprivation for the child therefore the referral may not be mandatory. We simply need to inform the worker. The worker will be required to make a decision based on the specific details of the case.

- If a minor is found on a type case and the type case is not on the restricted list, a pop-up will be provided to the worker. This will occur regardless of whether an adult exists on an active certification.
- This pop-up is only provided when certifying a minor.
- This situation is not a mandatory referral.
- Provide a pop-up with the information found.

Pop-Up

```

Message Line ...
MECXXXXX          *** M.E.D.SYSTEM ***          *DATE
                  LASES REFERRAL VALIDATION      *TIME

                  For Au Member N
                  Is a LASES Referral appropriate? _ _

*Tran: _____ Act: _ Key: _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12
      Help      Cont
    
```

A value of Y or N is accepted as input to the question.

If a value is not entered but PF3 is pressed, it is assumed the user wants to complete a referral and the decision is not completed.

PF3 will allow the user to continue.

**Other Notes**

At the time a decision is made the edits described above will occur. The order in which a decision is made on a case and the certifications within a case will have an impact on the execution of the above edits.

We are only considering certified/active cases and certs when looking for an adult with full title 19 eligibility. Therefore, let's look at a case with 2 certs being added. Cert 1 has a child on a 13-01 type case with approval code 2 and Cert 2 has the parent on a non restricted type case. If cert 2 is decided before cert 1; cert 1 requires a mandatory referral. This is because we will find the parent with active eligibility. If cert 1 is decided before cert 2, then the worker will get the pop-up asking if a referral is required. This is because the system doesn't find any active eligibility for the parent.

Also, if a child already exists in the system and we are adding the parent no pop-up will be issued. The pop-up only appears when certifying a child in an unrestricted type case.

**Restricted Type Cases (MECLCVL)**

This is a list of type cases that have limited Medicaid services or programs where mandatory Lases referrals are not required. We are using the type cases to assist with determining Mandatory Lases referrals.

Category Type Case	Description
01/40	Pure SLMB
01/47	EMS for Legal/Illegal Aliens
01/48	QI1
01/63	LTC Co-insurance

02/40	Pure SLMB
02/47	EMS for Illegal/Legal Aliens
02/48	QI1
02/94	QDWI
03/47	EMS for Illegal/Ineligible Aliens
03/69	Rolldown Medicaid
03/71	Transitional Medicaid
04/40	Pure SLMB
04/47	EMS for Illegal/Ineligible Aliens
04/48	QI1
04/94	QDWI
04/63	LTC Co insurance
05/01	Refugee Cash Assistance
05/20	RCA – Regular Medically Needy Program
05/21	RCA – Spend down Medically Needy Program
13/71	Lifc Transitional Medicaid
13/85	Child Support continuance
14/63	LTC Co insurance
16/12	Champ Presumptive Eligibility
17/95	Pure QMB
20/28	Tuberculosis
03/01	LAMI

## OCS Type Cases

OCS Category	Type Cases
06	All
08	All

15	All
22	All