



LOUISIANA DEPT. OF HEALTH & HOSPITALS

Medicaid Eligibility Data System

Internal Design: MMIS Interface

Table of Contents

DOCUMENT INFORMATION	5
REVISION SUMMARY	5
MMIS INTERFACES	7
QUICK REFRESHER: MEDS DATABASE STRUCTURE.....	7
OVERVIEW OF MMIS INTERFACE FILE	8
INTERFACE RECORD INFO.....	9
MMIS INTERFACE RECORD LAYOUTS	10
HEADER RECORD	10
PERSON RECORD	11
<i>Note about Race Code</i>	13
SWIPE CARD REQUEST RECORD.....	14
LOCK-IN/COMMUNITY CARE RECORD - OBSOLETE	15
<i>Note about MEDS Lock-In Status</i>	15
<i>Note about Sending Lock-In Records</i>	16
AU MEMBER RECORD.....	17
AU MEMBER PARISH OF RESIDENCE RECORD	18
AU MEMBER ELIGIBILITY RECORD.....	19
<i>Note about Person Records</i>	20
<i>Note about MEDS Sanctions</i>	20
<i>Note about Dual Eligibility</i>	20
<i>Note about Re-determinations</i>	20
AU MEMBER SEGMENT RECORD.....	21
<i>Note about Overlapping Dates</i>	22
PERSON CROSS REFERENCE RECORD.....	23
TRAILER RECORD.....	24
MEDS RECIPIENT INTERFACE EXTRACT PROCESS	25
MMIS RECIPIENT EXTRACT PROGRAM LOGIC	25
<i>Main Extract Logic - MEIMMIZ1</i>	25
<i>Sort Module Logic - MEIMMIZ2</i>	25
<i>Person Extract</i>	27
Get Person Detail routine.....	27
<i>Swipe Card Request Extract</i>	29
<i>Lock-In Extract - Obsolete</i>	30
<i>AU Member Extract</i>	32
<i>AU Member Parish of Residence Extract</i>	34
<i>AU Member Segment Extract</i>	38
<i>Invalid Person Cross Reference Extract</i>	39
MEDS MONTHLY FULL EXTRACT OF CURRENT ELIGIBILITY	42
MMIS MONTHLY FULL EXTRACT PROGRAM LOGIC	42
<i>Main Full Extract Logic – MEIMMIZ7</i>	42
<i>Sort Module Logic – MEIMMIZ8</i>	42
MMIS COMMON ROUTINES	43
DETERMINE IF PERSON HAS BEEN PREVIOUSLY SENT TO MMIS (MEIMMPN1)	43
MMIS RECONCILIATION INTERFACE - OBSOLETE	44

RECONCILIATION MEDS EXTRACT PROCESS	45
<i>Main Program Processing</i>	45
DETERMINE CURRENT AU MEMBER RECORD SUBPROGRAM.....	46
WRITE PERSON RECORD SUBROUTINE	47
WRITE EXTRACT SUBROUTINE	47
WRITE AU MEMBER RECORD SUBROUTINE	48
WRITE ELIGIBILITY SEGMENTS SUBROUTINE	49
CHECK FOR SEGMENTS SUBROUTINE	50
WRITE SEGMENTS SUBROUTINE	51
RECONCILIATION COMPARISON & REPORTING PROCESS - OBSOLETE.....	52
MAIN PROGRAM PROCESSING.....	52
RECORD MISSING FROM MMIS SUBROUTINE	54
RECORD MISSING FROM MEDS SUBROUTINE	54
WRITE RPT LINE SUBROUTINE.....	54
READ MEDS SUBROUTINE.....	55
READ MMIS SUBROUTINE	56
COMPARE RECORDS SUBROUTINE	57
CHECK PERSON FIELDS SUBROUTINE	60
CHECK AU MEMBER FIELDS SUBROUTINE	61
CHECK ELIGIBILITY RECORDS SUBROUTINE	62
CHECK ELIGIBILITY FIELDS SUBROUTINE	63
CHECK SEGMENT RECORDS SUBROUTINE	64
CHECK SEGMENT FIELDS SUBROUTINE	65
MEDS ELIGIBILITY INTEGRITY CHECKER PROCESS (MEIEICZ1) - OBSOLETE	66
<i>Program Logic</i>	66
APPENDIX A - SAMPLE REPORT - OBSOLETE.....	67
APPENDIX B – MMIS EXTRACT LOGIC PRIOR TO FEB 2004	68

Document Information

Created for: Louisiana Dept. of Health & Hospitals
Project Name: Medicaid Eligibility Data System
Project Abbreviation: MEDS
Document Title: Internal Design: MMIS Interface
Document Subject:
Revision Number: 211 (incremented each time the document is saved)
Status: Final
Last Saved: Dec 29, 2011 10:20:00 AM
Printed: Dec 29, 2011 10:20:00 AM
Comments:
Document Template: RedManeSpec_W2000.dot

Revision Summary

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

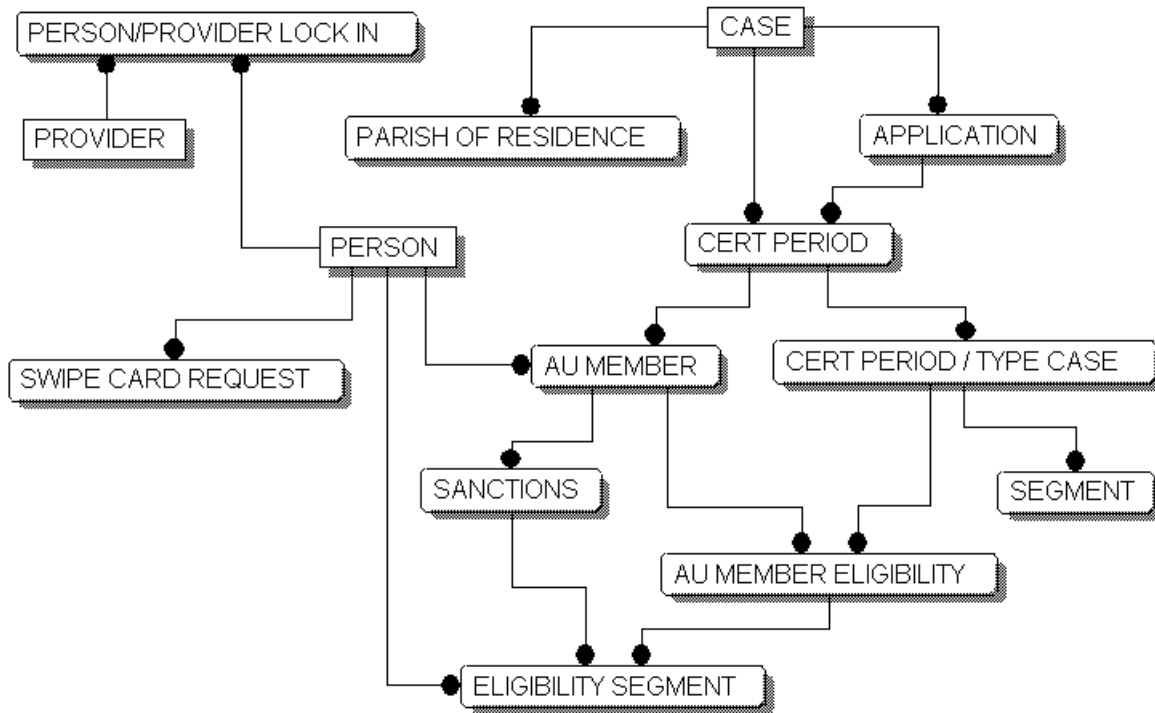
Date	Description of Change	By
4/30/2003	Revised with RedMane template	M. Smutko
2/25/04	Added more detail and removed out of date logic	Jennifer Leslie
1/17/2005	Add LTC Hospice Provider number (SIR1079)	Lizette Nel
1/24/2005	Comment out MEIMMPN2 (SIR1149)	Jonathan Go
11/21/2005	Exclude LIS certs from MMIS (SIR1238)	Karl Powers
11/21/2005	Add Permanent Parish of Residence and Katrina Ind (SIR1224)	Lizette Nel
01/11/2006	Out-of-State evacuee changes (SIR1245)	Jonathan Go
11/22/2006	Add New Race Codes, Ethnicity Code and Medicaid Mother Number (SIR1184)	Jonathan Go
03/11/2008	Remove Katrina/Rita related text (SIR1314)	Lauren O'Neil
07/06/2010	Add Federally Recognized Indicator on Person Record (SIR1620)	Jonathan Go
03/17/2011	Add GNOCHC FPL Percentage on AU Member record(SIR1661)	Rajesh Cheebur
4/1/2011	Updated for SIR1701	Lauren O'Neil

Date	Description of Change	By
<u>9/30/2011</u>	<u>Add Temp/Trans Type Case and Level of Care to Segment (SIR1694)</u>	<u>Jonathan Go</u>

MMIS Interfaces

Quick Refresher: MEDS Database Structure

MEDS Medicaid Eligibility information exists on multiple MEDS business objects. The following diagram illustrates the relationship between the different MEDS business objects (for more information about business objects, refer to the MEDS System Business Object Model in the MEDS External Design document):



Overview of MMIS Interface file

The Medicaid Management Information System (MMIS) Interface File consists of three different record types:

- 1 - Header Record
- 2 - Detail Record(s)
- 3 - Trailer Record

The **Header** record contains file control information. Each Interface file contains one Header record and the Header record is always the first record in the file.

The **Detail** records contain the actual MEDS information to be communicated to the MMIS system. A Detail record may have one of seven different sub-record types:

- 1 - Person Record
- 2 - Swipe Card Request Record
- 3 - Lock-In/Community Care Record
- 4 - AU Member Record
- 5 - AU Member Parish of Residence Record
- 6 - AU Member Eligibility Record
- 7 - AU Member Segment Record (long term care/waiver and hospice segments)
- 8 - Invalid Person Cross Reference Record

The **Trailer** record identifies the physical end of the interface file. Each Interface file contains one Trailer record and the Trailer record is always the last record in the file.

NOTE: The MMIS Monthly Reconciliation and MMIS Recipient Interfaces (neither currently are run) use the same record format. These two record formats must be kept in sync.

Interface Record Info

Every record in the interface file will contain **Record Info**. The Record Info will always be the first 32 characters in the record. The Record Info provides the following:

- Identifies the format layout of a record.
- Provides a single file sequence number.
- Provides a grouping of detail records by person number.

<u>Position</u>	<u>Length</u>	<u>Data Type</u>	<u>Picture</u>	<u>Data Element/Description</u>
001	001	AN	X(001)	Record Type: 1 - Header (one per file). 2 - Detail. 3 - Trailer (one per file).
002	001	AN	X(001)	Sub-Record Type: 0 - Control Record (Header/Trailer). 1 - Person Record. 2 - Swipe Card Request Record. 3 - Lock-In/Community Care Record. 4 - AU Member Record. 5 - AU Member Parish of Residence Record. 6 - AU Member Eligibility Record. 7 - AU Member Segment Record. 8 - Invalid Person Cross Reference Record.
003	009	Group		Record Key: Record Sequence.
003	009	N	9(009)	Record Sequence: Incremental number from 1 by 1 through all records in the file.
012	021	Group		Person Key: Person Number and Person Sequence. Note: the Person Key will be all zeroes on a control record.
012	013	N	9(013)	Person Number: Unique identifier of a MEDS person.
025	008	N	9(008)	Person Sequence: Incremental number from 1 by 1 through all records with the same Person Number.

MMIS Interface Record Layouts

Header Record

The Header record is the first record in the file and contains job control information. There is only one header record per file. This is a required record (i.e., every file will have one).

<u>Position</u>	<u>Length</u>	<u>Data Type</u>	<u>Picture</u>	<u>Data Element/Description</u>
001	001	AN	X(001)	Record Type: 1 - Header.
002	001	AN	X(001)	Sub-Record Type: 0 - Control Record.
003	009	Group		Record Key: see Record Info field descriptions.
012	020	Group		Person Key: see Record Info field descriptions.
033	67	Group		File Control Information.
033	025	AN	X(025)	File Name: MMIS RECIPIENT EXTRACT Or MMIS RECON EXTRACT
058	010	AN	X(010)	Creation Date (format: YYYY/MM/DD).
068	008	AN	X(008)	Creation Time (format: HH:MM:SS).
076	008	AN	X(008)	MEDS Process Id.
084	008	N	9(008)	MEDS Run Control Number.
092	008	N	9(008)	MEDS Cutoff Date (Recon only)
100	001	AN	X(001)	'*' End of record marker.

Person Record

The Person record contains the MEDS person's current information. If required, only one Person record is sent for each MEDS person extracted.

A MEDS person record is sent to MMIS:

- if they have any eligibility (open, closed or future) or are a client on a cert period (open, closed or future)
- if any of the following fields change: Last name, first name, middle initial, title, suffix, SSN, SSN Claim Number, birth date, mailing address lines, city, state, zip code, extended zip code, phone number, sex, race code, date of death, preferred language code, lock in code
- If they are not invalid persons

Technical notes:

The MMIS run number on the Person file is updated in MECPERU

<u>Position</u>	<u>Length</u>	<u>Data Type</u>	<u>Picture</u>	<u>Data Element/Description</u>
001	001	AN	X(001)	Record Type: 2 - Detail.
002	001	AN	X(001)	Sub-Record Type: 1 - Person Record.
003	009	Group		Record Key: see Record Info field descriptions.
012	020	Group		Person Key: see Record Info field descriptions.
033	241	Group		Person Detail Information.
033	009	N	9(009)	Social Security Number.
042	012	AN	X(012)	SSN Claim Number.
054	025	AN	X(025)	Last Name.
079	020	AN	X(020)	First Name.
099	001	AN	X(001)	Middle Initial.
100	008	N	9(008)	Birth Date (format: YYYYMMDD).
108	035	AN	X(035)	Address Line 1.
143	035	AN	X(035)	Address Line 2.
178	035	AN	X(035)	Address Line 3.
213	020	AN	X(020)	City.(note: MMIS only accepts 1st 18 characters)
233	002	AN	X(002)	State.
235	005	N	9(005)	Zip Code.
240	004	N	9(004)	Zip Code Extension.
244	010	N	9(010)	Phone Number (including area code).
254	001	N	9(001)	Sex Code.

<u>Position</u>	<u>Length</u>	<u>Data Type</u>	<u>Picture</u>	<u>Data Element/Description</u>
				Refer to External Design - Reference Tables.
255	001	N	9(001)	Race Code.
				Refer to External Design - Reference Tables.
256	008	N	9(008)	Date of Death (format: YYYYMMDD).
264	001	AN	X(001)	New Person Code: 0 - Person known to MMIS 1 - New Person (never before sent to MMIS)
265	1	N	9(001)	Lock In Code - Refer to External Design - Lock In Maint
266	003	AN	X(003)	Title
269	003	AN	X(003)	Suffix
272	002	N	9(002)	Preferred Language Indicator
				Refer to External Design - Reference Tables
275	004	N	9(004)	Permanently stored Parish of Residence
279	001	N	9(001)	Race-code-1 0 - Not White or Race unknown 1 - White
280	001	N	9(001)	Race-code-2 0 - Not Black or African American or Race unknown 1 - Black or African American
281	001	N	9(001)	Race-code-3 0 - Not American Indian or Alaskan Native or Race unknown 1 - American Indian or Alaskan Native
282	001	N	9(001)	Race-code-4 0 - Not Asian or Race unknown 1 - Asian
283	001	N	9(001)	Race-code-5 0 - Not Native Hawaiian or Other Pacific Islander or Race unknown 1 - Native Hawaiian or Other Pacific Islander
284	001	N	9(001)	Ethnicity-code 0 - Not Hispanic or Latino 1 - Hispanic or Latino 9 - Ethnicity Unknown
285	013	N	9(013)	Medicaid Mother Person Num
298	001	AN	X(001)	Federally Recognized Indicator
299	001	AN	X(001)	'*' End of record marker.

Note about Race Code

MMIS can now accept MEDS (MSIS) Race codes: (1) White, (2) Black, (3) American Indian or Alaskan Native, (4) Asian, (5) Hispanic or Latino, (6) Native Hawaiian or Other Pacific Islander, (7) Hispanic or Latino and one or more races, (8) More than 1 race (Hispanic or Latino not indicated), and (9) Unknown.

The new race codes are saved as a numeric value 1, 2, 3, 4 and 5 and are translated into the appropriate fields on the extract. The ethnicity code is saved as a 'N', 'Y' or a 'U' and is translated to 0, 1 or 9 on the extract.

Swipe Card Request Record

The Swipe Card Request record identifies a Person's swipe card request information. Each record identifies a single swipe card request. One Person record may have multiple Swipe Card Request records. No Swipe Card Requests are ever generated for persons who are exclusively in certs where the Type Case where the 'Swipe Card Required' indicator is set to 'N' on the Type Case Table (TCAS).

When multiple Swipe Card Request records exist for the same Person, the Swipe Card Request records will appear in ascending Request Date order (i.e., oldest request first).

A record is sent to MMIS when a Swipe Card Request is added in MEDS and when the reason code or mother is changed on an existing Swipe Card Request.

Technical notes:

The MMIS run number on the Swipe Card Request file is updated in MECSWPU

<u>Position</u>	<u>Length</u>	<u>Data Type</u>	<u>Picture</u>	<u>Data Element/Description</u>
001	001	AN	X(001)	Record Type: 2 - Detail.
002	001	AN	X(001)	Sub-Record Type: 2 - Swipe Card Request Record.
003	009	Group		Record Key: see Record Info field descriptions.
012	020	Group		Person Key: see Record Info field descriptions.
033	022	Group		Swipe Card Request Information.
033	008	N	9(008)	Request Date (format: YYYYMMDD).
041	001	AN	X(001)	Request Reason Code.
				Refer to External Design - Reference Tables.
042	013	N	9(013)	Mother's Person Number (Unique identifier).
055	001	AN	X(001)	'*' End of record marker.

Note: Swipe card records are not created for the Reconciliation interface.

Lock-In/Community Care Record - Obsolete

The Lock-In/Community Care record identifies a Person's lock-In information. Each record identifies a single lock-in period. One Person record may have multiple Lock-In/Community Care records.

When multiple Lock-In/Community Care records exist for the same Person, the Lock-In/Community Care records will appear in ascending Start Date order (i.e., oldest lock-in first).

A lock-in record will be sent to MMIS when the lock in becomes approved and if the close date changes once it is approved.

Technical notes:

The MMIS run number on the MEDS Lock In file is updated in MECPLIU

NOTE: The lock-in function is not currently in use in MEDS. No lock-in records are sent to MMIS from MEDS.

<u>Position</u>	<u>Length</u>	<u>Data Type</u>	<u>Picture</u>	<u>Data Element/Description</u>
001	001	AN	X(001)	Record Type: 2 - Detail.
002	001	AN	X(001)	Sub-Record Type: 3 - Lock-In/Community Care Record.
003	009	Group		Record Key: see Record Info field descriptions.
012	020	Group		Person Key: see Record Info field descriptions.
033	027	Group		Person Provider Lock-In Information.
033	007	N	9(007)	Provider Number.
040	008	N	9(008)	Start Date (format: YYYYMMDD).
048	008	N	9(008)	Close Date (format: YYYYMMDD).
056	002	AN	X(002)	Provider Lock-In Type.
				Refer to External Design - Reference Tables.
058	002	Group		Provider Information.
058	002	N	9(002)	Provider Type Code.
060	001	AN	X(001)	'*' End of record marker.

Note: Lock-in records are not created for the Reconciliation interface.

Note about MEDS Lock-In Status

MEDS Lock-In records will only be sent to MMIS when they are in "Approved" status (i.e., lock-ins waiting for approval or those rejected will never be sent to MMIS).

Note about Sending Lock-In Records

When a new Lock-In record is added or an existing Lock-In record changes (closes), the new/changed Lock-In record will be sent along with all other "open" Lock-In records.

AU Member Record

The AU (assistance unit) Member record contains the MEDS Person's Application, Certification Period, and AU Member information. .

An AU Member record will be sent to MMIS from MEDS when an AU Member is added and when EPSDT code or Adult Indicator change

Technical notes:

The MMIS run number on the MEDS AU Member file is updated in MECAUMU

<u>Position</u>	<u>Length</u>	<u>Data Type</u>	<u>Picture</u>	<u>Data Element/Description</u>
001	001	AN	X(001)	Record Type: 2 - Detail.
002	001	AN	X(001)	Sub-Record Type: 4 - AU Member Record.
003	009	Group		Record Key: see Record Info field descriptions.
012	020	Group		Person Key: see Record Info field descriptions.
033	008	Group		Application Information.
033	008	N	9(008)	Application Date (format: YYYYMMDD).
041	046	Group		Certification Period Information.
041	025	AN	X(025)	Client Last Name.
066	020	AN	X(020)	Client First Name.
086	001	AN	X(001)	Client Middle Initial.
087	008	Group		Calculated Values.
087	005	N	9(05)	Gross Family Income. Calculated at extract time. Sum of Person Income Value.
092	003	N	9(03)	Family Size. Calculated at extract time. Sum of AU Members (using unique person numbers).
095	002	Group		AU Member Information.
095	001	AN	X(001)	Adult Indicator: Y - AU Member is an Adult. N - AU Member is not an Adult.
096	001	AN	X(001)	EPSDT Code. Refer to External Design - Reference Tables.
097	006	N	9(3.2)	FPL Percentage FPL Percentage per line 6 on the budget
103	001	AN	X(001)	'*' End of record marker.

AU Member Parish of Residence Record

The AU Member Parish of Residence record contains the MEDS person's Parish of Residence information. Each record identifies a single occurrence of a Parish of Residence (accompanied by an effective date). One AU Member record may have many AU Member Parish of Residence records.

When multiple AU Member Parish of Residence records exist for the same AU Member, the AU Member Parish of Residence records will appear in ascending Effective Date order (i.e., oldest parish first).

Parish records will be sent to MMIS from MEDS when an AU Member is initially added and when the parish of residence (location type or id) changes for that AU Member.

Technical notes:

For Adds the MMIS interface checks the AU Member log file for AU Members marked to go to MMIS.

For changes there is a MMIS run number on the MEDS Case file that is updated in MECCASU when parish of residence changes.

<u>Position</u>	<u>Length</u>	<u>Data Type</u>	<u>Picture</u>	<u>Data Element/Description</u>
001	001	AN	X(001)	Record Type: 2 - Detail.
002	001	AN	X(001)	Sub-Record Type: 5 - Parish of Residence Record.
003	009	Group		Record Key: see Record Info field descriptions.
012	020	Group		Person Key: see Record Info field descriptions.
033	012	Group		Parish of Residence Information.
033	008	N	9(008)	Parish of Residence Effective Date (format: YYYYMMDD). The date the Parish of Residence was added or changed.
041	004	N	9(004)	Parish of Residence (Location Id). *Note: MMIS may have to compress into 2 bytes.
045	001	AN	X(001)	'*' End of record marker.

Note: Parish of residence records are not created for the Reconciliation interface.

AU Member Eligibility Record

The AU Member Eligibility record contains the MEDS Person's eligibility information. Each record identifies a single occurrence of a Person's eligibility. Eligibility is uniquely identified by Person Number and Eligibility Sequence Number. One AU Member record (within a single Certification Period) may have multiple AU Member Eligibility records.

When multiple AU Member Eligibility records exist for the same Person, the AU Member Eligibility records will appear in Eligibility Sequence Number order (i.e., oldest eligibility first).

These eligibility records are sent to MMIS when eligibility is added and when start date, close date, closure code or deleted indicator change on any eligibility segments.

If eligibility is extended rather than adjusting the dates on existing segments new segments are created and sent to MMIS. This is so MMIS can maintain the original start dates of eligibility.

<u>Position</u>	<u>Length</u>	<u>Data Type</u>	<u>Picture</u>	<u>Data Element/Description</u>
001	001	AN	X(001)	Record Type: 2 - Detail.
002	001	AN	X(001)	Sub-Record Type: 6 - AU Member Eligibility Record.
003	009	Group		Record Key: see Record Info field descriptions.
012	020	Group		Person Key: see Record Info field descriptions.
033	005	Group		AU Member Eligibility Key
033	005	N	9(005)	Eligibility Sequence Number. Unique within Person Number.
038	040	Group		Eligibility Information
038	013	N	9(013)	Case Number.
051	003	N	9(003)	Category Code.
054	003	N	9(003)	Type Case Code.
057	008	N	9(008)	Start Date (format: YYYYMMDD).
065	008	N	9(008)	Close Date (format: YYYYMMDD).
073	002	N	9(002)	Approval Code.
075	002	N	9(002)	Closure Code.
077	001	AN	X(001)	Delete Indicator. Identifies a logically purged eligibility segment. Y - Purged
078	001	AN	X(001)	'*' End of record marker.

NOTE: At this point MMIS does not process our delete indicator.

Note about Person Records

Whenever any records are sent to MMIS the record will always be accompanied by a Person record.

Note about MEDS Sanctions

MEDS supports the notion of a Sanction Period. A Sanction Period is a period of ineligibility with a MEDS AU Member's eligibility start and end date. When Sanction Periods are found in MEDS, the MMIS Recipient Interface will create multiple AU Member Eligibility records in the extract file. Each record will represent an actual period of eligibility (start and end dates) within the MEDS AU Member's eligibility start and end date. The period of eligibility whose close date is the result of a sanction will have a closure code of 57. Logically purged AU Member Eligibility records (which are created from changes to future sanctions) have a closure code of 27 (cancel future eligibility).

Note about Dual Eligibility

It has been decided that for dual eligibility, the MEDS extract will be required to send an AU Member Eligibility record for each Type Case.

Note about Re-determinations

When a MEDS re-determination is done and eligibility does not change, no information will be sent to MMIS.

AU Member Segment Record

The AU Member Segment record contains the MEDS Person's Segment information (i.e., long-term care information) and the Persons Hospice Segment information. Each record identifies a single occurrence of a person's LTC or Hospice segment. One AU Member Eligibility record (single Type Case) may have multiple AU Member Segment records.

When multiple AU Member Segments records exist for the same AU Member Eligibility, the AU Member Segment records will appear in ascending Start Date order (i.e., oldest segment first).

A Segment record will be sent to MMIS from MEDS when a Segment (LTC, waiver) is added and when start date, close date, discharge date, provider number, level of care, admission code, admission date, PLI amount, OSS amount or diagnosis code change on an existing segment change.

Technical notes:

The MMIS run number on the MEDS Segment file is updated in MECSGMU

<u>Position</u>	<u>Length</u>	<u>Data Type</u>	<u>Picture</u>	<u>Data Element/Description</u>
001	001	AN	X(001)	Record Type: 2 - Detail.
002	001	AN	X(001)	Sub-Record Type: 7 - AU Member Segment Record.
003	009	Group		Record Key: see Record Info field descriptions.
012	020	Group		Person Key: see Record Info field descriptions.
033	005	Group		AU Member Eligibility Key
033	005	N	9(005)	Eligibility Sequence Number. Unique within Person Number.
038	072	Group		Segment Information.
038	008	N	9(008)	Segment Start Date (format: YYYYMMDD).
046	008	N	9(008)	Segment Close Date (format: YYYYMMDD).
054	007	N	9(007)	Provider Number.
061	002	N	9(002)	Level Of Care Code.
				Refer to External Design - Reference Tables.
063	001	AN	X(001)	Admission Code.
				Refer to External Design - Reference Tables.
064	008	N	9(008)	Admission Date (format: YYYYMMDD).
072	008	N	9(008)	Discharge Date (format: YYYYMMDD).
080	013	N	9(11)V9(02)	PLI Amount.
093	005	AN	X(005)	Primary Diagnosis Code
098	005	AN	X(005)	Secondary Diagnosis Code
103	007	N	9(007)	LTC Hospice Provider number
<u>110</u>	<u>003</u>	<u>N</u>	<u>9(003)</u>	<u>Temp/Trans Type Case</u>

<u>Position</u>	<u>Length</u>	<u>Data Type</u>	<u>Picture</u>	<u>Data Element/Description</u>
<u>113</u>	<u>002</u>	<u>N</u>	<u>9(002)</u>	<u>Temp/Trans Level of Care</u>
<u>1150</u>	001	AN	X(001)	'*' End of record marker.

Note about Overlapping Dates

AU Member Segment records cannot have overlapping dates.

Person Cross Reference Record

The Person Cross Reference record contains the MEDS person number with all its cross-referenced invalid person numbers. This record is only sent to MMIS if the valid Person and at least one of the invalid Persons have previously been sent to MMIS.

A Person Cross Reference record is sent to MMIS:

- if they have any eligibility (open, closed or future) or are a client on a cert period (open, closed or future)
- if they have been cross-referenced with another person that had eligibility or were a client on a cert period and this other person has been marked as invalid.

Technical notes:

The MMIS xref run number on the MEDS Person file is updated in MECGIPN.

<u>Position</u>	<u>Length</u>	<u>Data Type</u>	<u>Picture</u>	<u>Data Element/Description</u>
001	001	AN	X(001)	Record Type: 2 - Detail.
002	001	AN	X(001)	Sub-Record Type: 8 - Person Record.
003	009	Group		Record Key: see Record Info field descriptions.
012	020	Group		Person Key: see Record Info field descriptions.
033	67	Group		Person Cross Reference Information.
033	1	AN	X(001)	Record action A - Add cross reference link R - Remove cross reference link
034	013	N	9(013)	Invalid Person number 1
047	013	N	9(013)	Invalid Person number 2
060	013	N	9(013)	Invalid Person number 3
073	013	N	9(013)	Invalid Person number 4
086	013	N	9(013)	Invalid Person number 5
099	001	AN	X(001)	'*' End of record marker.

Trailer Record

The Trailer record is the last record in the file and identifies the physical end of the file. There is only one trailer record per file. This is a required record (i.e., every file will have one).

<u>Position</u>	<u>Length</u>	<u>Data Type</u>	<u>Picture</u>	<u>Data Element/Description</u>
001	001	AN	X(001)	Record Type: 3 - Trailer.
002	001	AN	X(001)	Sub-Record Type: 0 - Control Record.
003	009	Group		Record Key: see Record Info field descriptions.
012	020	Group		Person Key: see Record Info field descriptions.
033	001	AN	X(001)	'*' End of record marker.

MEDS Recipient Interface Extract Process

The MEDS MMIS Recipient Interface consists of a single extract process and produces 2 extract files. The second (alternate) file is created for SLMB, Q1, Q2 and QDWI type cases only. MEDS information is communicated to MMIS using several different sub-record types (e.g., Person, AU Member, etc.). Each sub-record type contains MEDS information relating to that specific sub-record type object. All sub-record types are grouped together by use of a unique MEDS Person Number (or MMIS Recipient Number). Except where otherwise noted, the MEDS information (contained in a sub-record type) is only extracted when it changes in MEDS. Whenever records are sent to MMIS, the entire record is populated (i.e., if a Person's last name changes, all Person information on the Person detail record is populated including name, address, SSN, etc.).

Each group (or package) of records must contain a corresponding person record (sub-record type 1). This Person record will be sorted to be the first record in the group.

When eligibility is initially added to MEDS for a person MMIS expects a full package of records in the interface (that is, all relevant record types),

MMIS Recipient Extract Program Logic

Main Extract Logic - MEIMMIZ1

Get Current / Restart Run Number

Build Header Record

Perform Person Extract

Perform Swipe Card Extract

Perform Lock In Extract – OBSOLETE

Perform AU Member Extract

Perform AU Member Parish of Residence Extract

Perform AU Member Eligibility Extract

Perform AU Member Segment Extract

Perform Invalid Person Cross Reference Extract

Build Trailer Record

Sort Module Logic - MEIMMIZ2

Read work file 1 #EXTRACT-REC

Sort by Record Type, Person number, Sub Record type

At Start of Data

#Rec-Seq = 1

End-Start

Assign Record Seq = #Rec-Seq

Add 1 to #Rec-Seq

If Person-number #SAVE-PERSON

 Assign #Person-Seq = 1

End If

Assign Person-Sequence = #Person-Seq

Add 1 to #Person-Seq

Write WORK FILE 2 #EXTRACT-REC

End-Sort

Subtract 1 from #REC-SEQ

Move #REC-SEQ to #RESTART-DATA

Call run-control #INTERMEDIATE-ET

Call run-control #COMPLETE

Person Extract

Read ME-PERSON by MMIS-PERSON-NUM from #CURR-RUN-NUM

```
If MMIS-Run-Num not = #CURR-RUN-NUM
  Escape Bottom
End If
```

The spec used to say ignore invalid people but we can't as their eligibility may be closing and we need to send that closure to MMIS even though the person is already marked as invalid. Code is correct as of 5/10/2004.

Read ME-AU-MEMBER by PERSON-START-DATE-9C from person-num

```
If ME-AU-MEMBER.PERSON-NUM not = ME-PERSON.PERSON-NUM
  Escape bottom
```

```
End If
CALLNAT 'MECEPU' /* to get cert period type case
```

Perform Get Person Detail

```
If CAT-CODE = 1,2,4 with TYPE-CASE =108 (LIS)
  Escape top
End-If
```

```
If TYPE-CASE = 40, 48, 49, 94
  WRITE ALTERNATE #EXTRACT-REC
Else
  WRITE #EXTRACT-REC
End If
```

```
If person written to both files
  Escape Bottom /* move on to the next person
End If
```

End-Read

End-Read

Get Person Detail routine

Build #Extract-Rec.RECORD-KEY, #Extract-Rec.PERSON-KEY
Move Person data to #EXTRACT-REC

CALLNAT 'MEIMMPN2' /* formats address lines (Commented out due to SIR1149)
Move formatted address lines to #EXTRACT-REC

CALLNAT 'MEIMMPN1' /* sets new person indicator from Person Log file
Move new person indicator to #EXTRACT-REC.New-Person-Indic

/* Translate the ethnicity code and race codes as follows:

- Find the MEDS ethnicity code on METETHL (Examine) and move the corresponding MMIS ethnicity code to #MEIMMIL1.#ETHNICITY-CODE
- Reset race-code-1, race-code-2, race-code-3, race-code-4, race-code-5

- Race-code-1 – White:
 - Check if race-code = 1 is in the RACE-CODES MU on ME-PERSON
 - If race-code = 1 found, then
 - Set #MEIMMIL1.#RACE-CODE-1 = 1
 - End if
- Race-code-2 – Black or African American:
 - Check if race-code = 2 is in the RACE-CODES MU on ME-PERSON
 - If race-code = 2 found, then
 - Set #MEIMMIL1.#RACE-CODE-2 = 1
 - End if
- Race-code-3 – American Indian or Alaskan Native:
 - Check if race-code = 3 is in the RACE-CODES MU on ME-PERSON
 - If race-code = 3 found, then
 - Set #MEIMMIL1.#RACE-CODE-3 = 1
 - End if
- Race-code-4 – Asian
 - Check if race-code = 4 is in the RACE-CODES MU on ME-PERSON
 - If race-code = 4 found, then
 - Set #MEIMMIL1.#RACE-CODE-4 = 1
 - End if
- Race-code-5 – Native Hawaiian or Other Pacific Islander:
 - Check if race-code = 5 is in the RACE-CODES MU on ME-PERSON
 - If race-code = 5 found, then
 - Set #MEIMMIL1.#RACE-CODE-5 = 1
 - End if

Swipe Card Request Extract

Swipe card requests are never generated for type cases 40, 48, 49 or 94. Online logic in MEDS correctly flags swipe card requests to go to MMIS.

Read ME-SWIPE-CARD-REQST by MMIS-PERSON-REQUEST-DATE from #CURR-RUN-NUM

 If MMIS-Run-Num not = #CURR-RUN-NUM

 Escape Bottom

 End if

 WRITE #EXTRACT-REC /* for swipe card request

 /* always write person record with swipe card request – duplicates will be eliminated later

 Find ME-PERSON with PERSON-NUM = ME-SWIPE-CARD-REQST.PERSON-NUM

 Perform Get Person Detail

 WRITE #EXTRACT-REC /* for person

 End Find

End-Read

Lock-In Extract - Obsolete

We must send all Approved Lock In records that are open or that have been closed since the last MMIS run. We must also create them in date order.

LockIn-Hist.

Histogram ME-PERSON-PROV-LOCK-IN by MMIS-PERSON-DATE from #CURR-RUN-NUM

```
If MMIS-Run-Num not = #CURR-RUN-NUM
  Escape Bottom (LockIn-Hist.)
End If
```

```
IF ME-PERSON-PROV-LOCK-IN.PERSON-NUM NE #SAVE-PERSON
  Assign #SAVE-PERSON = ME-PERSON-PROV-LOCK-IN.PERSON-NUM
```

```
Read ME-AU-MEMBER by PERSON-START-DATE-9C from person-num
```

```
  If ME-AU-MEMBER.PERSON-NUM not = ME-PERSON-PROV-LOCK-IN.PERSON-NUM
    Escape bottom
```

```
  End If
```

```
  CALLNAT 'MECCEPU' /* to get cert period type case
```

```
End-Read
```

LockIn-Read.

```
Read ME-PERSON-PROV-LOCK-IN by PERSON-PROV-START-DATE from #SAVE-PERSON
```

```
  If Person-Num not = #SAVE-PERSON
    Escape Bottom (LockIn-Read.)
```

```
  End If
```

```
  If Lock-in-status = A /* accept approved Open or just closed recs
```

```
  And (Close-date = 0 OR MMIS-RUN-NUM = #CURR-RUN-NUM)
```

```
    Build #Extract-Rec.RECORD-KEY, #Extract-Rec.PERSON-KEY
```

```
    Move PROV-NUM, START-DATE, CLOSE-DATE, LOCK-IN-TYPE to #EXTRACT-REC
```

```
    If PROV-NUM not = #SAVE-PROV then
```

```
      Assign #SAVE-PROV = ME-PERSON-PROV-LOCK-IN.PROV-NUM
```

Provider-Find.

```
      Find (1) ME-PROVIDER with PROV-NUM = #SAVE-PROV
```

```
      Move PROV-TYPE to #SAVE-TYPE
```

```
      End-Find /*Provider-Find
```

```
    End-If
```

```
    Move #SAVE-TYPE to #EXTRACT-REC
```

```
    If TYPE-CASE = 40, 48, 49, 94
```

```
      WRITE ALTERNATE #EXTRACT-REC
```

```
    Else
```

```
      WRITE #EXTRACT-REC
```

```
    End If
```

```
  End-If /* lock status A, open or just closed
```

```
End-Read /* LockIn-Read.
```

```
/* write person record if we haven't already
```

```
Find ME-PERSON with PERSON-NUM = #SAVE-PERSON-NUM
  If ME-PERSON.MMIS-RUN-NUM not = #CURR-RUN-NUM
    Perform Get Person Detail
    If TYPE-CASE = 40, 48, 49, 94
      WRITE ALTERNATE #EXTRACT-REC /* for person
    Else
      WRITE #EXTRACT-REC /* for person
    End If
  End If
End Find
End-if /* different person
End-Histogram /* LockIn-Hist
```

AU Member Extract

This routine creates AU Member records, and also parish records for newly added AU Members

AUMbr-Read.

Read ME-AU-MEMBER by MMIS-CASE-CERT-AU-MBR from #CURR-RUN-NUM

If MMIS-Run-Num not = #CURR-RUN-NUM

Escape Bottom (**AUMbr-Read.**)

End If

/* write a parish record for newly added AU Members

Read AU-Member-Log using key log-by-entry

If the log action = 'A'

Write a Parish of Residence record for the newly added AU Members

End If

End-Read Au Member Log

Build #Extract-Rec.RECORD-KEY, #Extract-Rec.PERSON-KEY

Cert-Find.

Find (1) ME-CERT-PERIOD with CASE-CERT-PERIOD-NUM = AU Mbr's Cert

Assign #APPL-NUM = ME-CERT-PERIOD.APPL-NUM

Assign #CLIENT-NUM = ME-CERT-PERIOD.CLIENT-NUM

End-Find /* **Cert-Find**

TCas-Find. /* to get type case

Find (1) ME-CPTC with CASE-CERT-CPTC-NUM = AU Mbr's Cert

End-Find /* **Tcas-Find**

Client-Find.

Find (1) ME-PERSON with PERSON-NUM = AU Mbr's Cert

Move Persons name to #EXTRACT-REC

End-Find /* **Client-Find**

Appl-Find.

Find (1) ME-APPLICATION with CASE-APPL-NUM = Cert's Case-Appl-Num

Move ME-APPLICATION.APPL-DATE to #EXTRACT-REC

End-Find /* **Appl-Find.**

Move ME-AU-MEMBER.ADULT-INDIC, ME-AU-MEMBER.EPSDT-CODE to #EXTRACT-REC

IF ME-AU-MEMBER's CASE-CERT not = #SAVE-CASE-CERT

/* this is the first AU member for this cert period. We need to calculate family size

/* and gross family income

Assign #SAVE-CASE = CASE-NUM

Assign #SAVE-CERT = CERT-PERIOD-NUM

AUMbr-Read2.

Read ME-AU-MEMBER by CASE-CERT-AU-MBR from #SAVE-CASE-CERT

If CASE-NUM not = #SAVE-CASE or CERT-PERIOD-NUM not = #SAVE-CERT

Escape Bottom (**AUMbr-Read2.**)

End If


```
        If CLOSE-DATE = 0 and CLAIMS-SUS-DATE = 0 and ELIGIBLE-INDIC = Y then
            Add 1 to #FAMILY-SIZE
        End If

    End Read      /* AUMbr-Read2

PerIncome-Read.
    Read ME-PERSON-INC-VALUE by CASE-CERT-PERSON-INC-TYPE from #SAVE-CASE-CERT
        If CASE-NUM NOT = #SAVE-CASE or CERT-PERIOD-NUM Not = #SAVE-CERT
            Escape Bottom (PerIncome-Read.)
        End If
        Add INC-AMT to #GROSS-FAMILY-INCOME
    End-Read      /* PerIncome-Read.
End If

Move #FAMILY-SIZE, #GROSS-FAMILY-INCOME to #EXTRACT-REC

If TYPE-CASE = 40, 48, 49, 94          /* from CPTC
    WRITE ALTERNATE #EXTRACT-REC
Else
    WRITE #EXTRACT-REC
End If

/* write person record if we haven't already
Find ME-PERSON with PERSON-NUM = ME-AU-MEMBER.PERSON-NUM

    If ME-PERSON.MMIS-RUN-NUM not = #CURR-RUN-NUM

        Perform Get Person Detail

        If TYPE-CASE = 40, 48, 49, 94
            WRITE ALTERNATE #EXTRACT-REC /* for person
        Else
            WRITE #EXTRACT-REC /* for person
        End If

    End If

End Find

End Read      /* AUMbr-Read.
```

AU Member Parish of Residence Extract

For each active AU member for the case whose parish was changed, send the complete parish history. Each Person-num should only be sent once. This routine extracts parish changes and resends only

Loop1.

Read ME-CASE by MMIS-CASE from #CURR-RUN-NUM

```
If MMIS-Run-Num >= #START-RUN-NUM and < #END-RUN-NUM
  Ignore
ELSE
  Escape Bottom
End If
```

Loop2.

Read ME-CASE-MEMBER by CASE-NUM-PERSON-NUM from ME-CASE.Case-Num

```
If Case-Num not = ME-CASE.Case-Num
  Escape Bottom (Loop2.)
End If
```

Loop3.

Read ME-AU-MEMBER by PERSON-START-DATE-9C from ME-CASE-MEMBER.Person-Num

```
If Person-Num not = ME-CASE-MEMBER.Person-Num
  Escape Bottom (Loop3.)
End If
```

```
If Case-Num not = ME-CASE.Case-Num
  Escape Top
End If
```

```
If Close-Date > 0 and < #TODAY
  Or Claims-Sus-Date > 0 and < #TODAY
```

```
  /* records closed in the past could be resends. Resends must have Person stamped
  /* to go to MMIS as well
```

```
  Find (1) ME-PERSON with Person-Num = ME-AU-MEMBER.PERSON-NUM
```

```
  If ME-PERSON.MMIS-RUN-NUM = #CURR-RUN-NUM and ME-CASE.MMIS-RUN-NUM
  = #CURR-RUN-NUM
```

```
    Ignore
```

```
  Else
```

```
    Escape Top    /* (Loop3.)
```

```
  End If
```

```
End If
```

```
Find (1) ME-CPTC with CASE-CERT-CPTC-NUM from AU-MEMBER
End-Find
```

```
Build #Extract-Rec.RECORD-KEY, #Extract-Rec.PERSON-KEY
```

```
For #I = 1 to C*Parish-Res-History
```

```
  Move Parish of Residence and Effective Date to #Extract-Rec
```

```
    If CAT-CODE = 1,2,4 with TYPE-CASE =108 (LIS)
      Escape top
    End-If

    If TYPE-CASE = 40, 48, 49, 94          /* from CPTC
      WRITE ALTERNATE #EXTRACT-FILE
    Else
      WRITE #EXTRACT-RECORD
    End If
  End-For

  /* write person record if we haven't already
  Find ME-PERSON with PERSON-NUM = ME-AU-MEMBER.PERSON-NUM

  If ME-PERSON.MMIS-RUN-NUM not = #CURR-RUN-NUM
    Perform Get Person Detail

    If CAT-CODE = 1,2,4 with TYPE-CASE =108 (LIS)
      Escape top
    End-If

    If TYPE-CASE = 40, 48, 49, 94
      WRITE ALTERNATE #EXTRACT-REC /* for person
    Else
      WRITE #EXTRACT-REC /* for person
    End If

  End If

  End Find

  Escape Bottom (Loop3.)

  End-Read /* (Loop3.)

  End-Read /* (Loop2.)

  End-Read /* (Loop1.)
```

AU Member Eligibility Extract

Send all eligibility segment changes to MMIS, unless there are deleted segments that have never been sent to MMIS.

Loop1.

Read ME-ELIG-SEGMENT by MMIS-PERSON-ELIG from #CURR-RUN-NUM

```
If MMIS-Run-Num not = #CURR-RUN-NUM
    Escape Bottom
```

```
End If
```

```
/* If segment is marked as deleted check if it has ever been sent to MMIS by comparing current
/* run number on the segment to the run number when the segment was added. If it has never
/* gone to MMIS don't send it now.
```

```
If delete-indicator = 'Y'                                /* segment has been deleted
    If MMIS-Run-Num = MMIS-Add-Run-Num                    /* segment added since last went to MMIS
        Escape top                                        /* read next record
```

```
End if
```

```
End if
```

```
/* Get the Approval code and Type Case from the CPTC, if different from last ELIG-SEGMENT
```

```
If Cert-Period-Num not = #SAVE-CERT or CPTC-Num not = #SAVE-CPTC
```

```
    Assign #SAVE-CASE = Case-Num
```

```
    Assign #SAVE-CERT = Cert-Period-Num
```

```
    Assign #SAVE-CPTC = CPTC-Num
```

```
    CPTC-Find.
```

```
    Find ME-CPTC with CASE-CERT-CPTC-NUM = #SAVE-CCC
```

```
        Move APPV-Code to #Extract-Rec
```

```
        Move Category Type Case to #Extract-Rec
```

```
    End-Find /* CPTC-Find
```

```
End If
```

```
/* Move ELIG-SEGMENT data to record
```

```
Build #Extract-Rec.RECORD-KEY, #Extract-Rec.PERSON-KEY
```

```
Move Elig-Sequence-Num, Case-Num, Cat-Type-Case, Start-Date, Close-Date, Closure-Code to
#Extract-Rec
```

```
/* Closure Code 65 is set in MEDS when a start date is moved back or an end date is moved
/* forward, creating a new eligibility segment. We make these segments look like open/closes to
/* MMIS.
```

```
If ME-ELIG-SEGMENT.CLOSURE-CODE = 65
    Move 67 to #Extract-Rec.Approval Code
    Move 66 to #Extract-Rec.Closure Code
```

```
End If
```

```
If CAT-CODE = 1,2,4 with TYPE-CASE =108 (LIS)
```

```
    Escape top
```

```
End-If
```

```
If TYPE-CASE = 40, 48, 49, 94                                /* from CPTC
```

```
    WRITE ALTERNATE #EXTRACT-FILE
```

```
Else
```

```
        WRITE #EXTRACT-RECORD
    End If

    /* write person record if we haven't already

    Find ME-PERSON with PERSON-NUM = ME-ELIG-SEGMENT.PERSON-NUM

        If ME-PERSON.MMIS-RUN-NUM not = #CURR-RUN-NUM

            Perform Get Person Detail

            If TYPE-CASE = 40, 48, 49, 94
                WRITE ALTERNATE #EXTRACT-REC /* for person
            Else
                WRITE #EXTRACT-REC /* for person
            End If

        End If

    End Find

End-Read /* Loop1
```

AU Member Segment Extract

Since MMIS cannot expand their files to store the full case-cert-cptc key for the segment, we find the first eligibility segment associated with the CPTC and send its sequence number instead. This will allow us to backtrack to the appropriate CPTC later on.

Loop1.

Read ME-SEGMENT by MMIS-CASE-CERT-CPTC-SEG-NUM from #CURR-RUN-NUM

 If MMIS-Run-Num not = #CURR-RUN-NUM
 Escape Bottom

 End If

 Read (1) ME-ELIG-SEGMENT by CASE-CERT-CPTC-AU from #SAVE-CCC
 Move Elig-Sequence-Num to #Extract-Rec

 End-Read

 Find ME-CPTC with CASE-CERT-CPTC-NUM from ME-SEGMENT
 End-Find /* **CPTC-Find**

 Build #Extract-Rec.RECORD-KEY, #Extract-Rec.PERSON-KEY

 Move data from ME-SEGMENT to #Extract-Rec

 If TYPE-CASE = 40, 48, 49, 94 /* from CPTC
 WRITE ALTERNATE #EXTRACT-REC

 Else

 WRITE #EXTRACT-REC

 End If

 /* write person record if we haven't already

 Find ME-PERSON with PERSON-NUM = ME-ELIG-SEGMENT.PERSON-NUM

 If ME-PERSON.MMIS-RUN-NUM not = #CURR-RUN-NUM

 Perform Get Person Detail

 If TYPE-CASE = 40, 48, 49, 94

 WRITE ALTERNATE #EXTRACT-REC /* for person

 Else

 WRITE #EXTRACT-REC /* for person

 End If

 End If

 End Find

End-Read

Invalid Person Cross Reference Extract

Subroutine Person-Xref-Extract

/* Extract Xref records from the me-person file

Read-pers-xref.

Read Person by MMIS-XREF-PERSON-NUM for the current MMIS Run number

Escape out of read loop if run number changes

/* determine what records to create

Set #MECGIPN-IN.#DETAIL = true

Set #MECGIPN-IN.#MMIS-RUN-NUM = #RUN-CONTROL.RESTART-RUN-NU

Call sub program **Error! Reference source not found.** to get the Xref info passing ME-PERSON as a parameter. This sub program will return an array containing any new invalid-person numbers as well as invalid-person numbers that have been removed.

/* Check on which file(s) the record type 8 must be written. First check the valid person.

Set #PERSON-START-DATE-9C.PERSON-NUM = ME-PERSON.PERSON-NUM

Reset #LOGIC.#MAIN-FILE and #LOGIC.#ALT-FILE

Perform Subroutine Check-Which-Work-File (page 40) to determine work file

Set #LOGIC.#PERSON-ON-ALT-FILE = #LOGIC.#ALT-FILE

Set #LOGIC.#PERSON-ON-MAIN-FILE = #LOGIC.#MAIN-FILE

/* Now check the invalid persons and build an array of flags

For-actions.

For both actions (for #i = 1 to 2)

If the record action is not populated (#MECGIPN-OUT.#RECORD-ACTION(#I) = ' '), then
Escape top /* (for-actions.)

End-if

For-inv-pers.

For every invalid person in the array (for #j = 1 to 5)

If #MECGIPN-OUT.#INV-PERSON-NUM(#i,#j) = 0, then
Escape top /* (for-inv-pers.)

End-if

Set #PERSON-START-DATE-9C.PERSON-NUM = #MECGIPN-OUT.#INV-PERSON-NUM(#I,#J)

Reset #LOGIC.#MAIN-FILE and #LOGIC.#ALT-FILE

Perform Subroutine Check-Which-Work-File (page 40) to determine work file

Save the #LOGIC.#ALT-FILE and #LOGIC.#MAIN-FILE flags in a multidimensional array
(#LOGIC.#INV-ALT-FILE(#i,#j) = #LOGIC.#ALT-FILE and #LOGIC.#INV-MAIN-FILE(#i,#j)
= #LOGIC.#MAIN-FILE).

End-for /* (for-inv-pers.)

End-for /* (for-actions.)

/* Now write the record type 8's to the appropriate work file

Perform Subroutine Write-Xref-To-Work-File (page 40) to write the record type 8's to the appropriate file(s)

End subroutine

Subroutine Check-Which-Work-File

/* Check on which file the person should be written. Check for both files because we might have to write the record type 8 to both files.

Read-AUM.

Read me-au-member by person-start-date-9c from #person-start-date-9c

Escape out of read loop if person-num changes

Move by name me-au-member to #me-cert-period

Perform GET-CERT-PERIOD (existing subroutine)

If type-case = 40, 48, 49, 94 (#CONSTANT.#ALTERNATE-TYPE-CASE(*))

Set #logic.#alt-file to true

Else

Set #logic.#main-file to true

End if

/* if both is true, we can stop reading

If #logic.#mainf-file and #logic.#alt-file

Escape out of read loop

End if

End read /* (read-AUM.)

End subroutine

Subroutine Write-Xref-To-Work-File

/* Write a record type 8 to the main file if the valid person and at least one invalid person belongs on the main file.

If #LOGIC.#PERSON-ON-MAIN-FILE

For-actions1.

For both actions (for #i = 1 to 2)

If the record action is not populated (#MECGIPN-OUT.#RECORD-ACTION(#I) = ' '), then

Escape top /* (for-actions1.)

End-if

Reset #MMIS-IDX

For-inv-pers1.

For every invalid person (for #j = 1 to 5)

If #MECGIPN-OUT.#INV-PERSON-NUM(#i,#j) = 0, then

Escape top /* (for-inv-pers1.)

End-if

If the invalid person belongs on the main file (If #LOGIC.#INV-MAIN-FILE(#i.#j)), then


```

        Add 1 to #MMIS-IDX
        #EXTRACT-REC.#INV-PERSON-NUM(#MMIS-IDX) =
            #MECGIPN-OUT.#INV-PERSON-NUM(#i,#j)
    End-if
End-for /* (for-inv-pers1.)

If there are any invalid persons (#MMIS-IDX > 0)
    Create a record type 8 with:
        /* Header Information
        #EXTRACT-REC.#RECORD-TYPE = #MEIMMIL2.#DETAIL
        #EXTRACT-REC.#SUB-RECORD-TYPE = #MEIMMIL2.#PERSON-XREF
        #EXTRACT-REC.#RECORD-SEQUENCE = 0
        #EXTRACT-REC.#PERSON-NUMBER = ME-PERSON.PERSON-NUM
        #EXTRACT-REC.#PERSON-SEQUENCE = 1
        /* Xref detail information

        #EXTRACT-REC.#RECORD-ACTION = #MECGIPN-OUT.#RECORD-ACTION(#i)

        /* Write Xref record
        #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
        Add 1 to Xref counter
        Write record to work file (WRITE-WORK-FILE)
        Write a person record (record type 1 to the work file) – see PERSON-DETAIL.
    End if /* (there are invalid persons

End for /* (for-actions1.)

End if /* Person on main file

/* Write a record type 8 to the main file if the valid person and at least on invalid person belongs on the
main file.

If #LOGIC.#PERSON-ON-ALT-FILE
    Reset #EXTRACT-REC

    Do exactly as above for the main file except check the #LOGIC.#INV-ATL-FILE flag and perform
    WRITE-ALT-WORK-FILE to write the record to the alternate work file. The person record (record type
    1) for the valid person must also be written to the alternate work file.

    End if /* Person on alt file

End Subroutine

```

MEDS Monthly Full Extract of Current Eligibility

This is a monthly process to create a full extract of current eligibility for MMIS. Like the daily job it consists of a single extract process and produces 2 extract files – one for most type cases, and one for SLMB, Q1, Q2 and QDWI type cases only. Sub-record types 1 (Person), 4 (AU Member), 5 (Parish), 6 (Eligibility) and 7 (Segments) are created with the same layouts as the daily process. Sub-record types 2 (Swipe cards) and 3 (Lock-in) are not created in the monthly extract. Person records marked as invalid/duplicate are not extracted. This process replaces the MMIS reconciliation.

NOTE: This interface has not been run since May 2002 in production

MMIS Monthly Full Extract Program Logic

Main Full Extract Logic – MEIMMIZ7

Get Current / Restart Run Number

Build Header Record

Read AU Member by Person, Close Date 9C, Start Date 9C, Case number, Cert Period

 If Person = Previous person and Close Date = 0 /* active dual eligibility

 Perform Get Cert Period /* call MECCEPU

 Perform Write AU Member Eligibility

 Perform Write Segment

 Escape Top

End If

Perform Get Cert Period /* call MECCEPU

Perform Get Person /* call MECPERU

If Person Lock Indicator = Yes

 Escape Top /* ignore these invalid person records

End If

Perform Write Person Detail

Perform Write AU Member

Perform Write AU Member Parish of Residence

Perform Write AU Member Eligibility

Perform Write Segment

End Read

Build Trailer Record

The data is written to 1 of 2 files using the same logic and the same file format at the daily job, with the exception of the New Person Indicator on the person extract. For the monthly extract it is set to 0 to indicate the person is not a new person, as the monthly job is run after a daily file has been extracted and before any new people could have been added to MEDS.

Sort Module Logic – MEIMMIZ8

This is identical to the sort for the daily interface, but using run control related to the monthly process.

MMIS Common Routines

Determine if Person has been previously sent to MMIS (MEIMMPN1)

When Person extract records are created, they are required to have an indicator as to whether the record represents a new person to MMIS. This common routine will be used to make that determination.

Input

Person Number (N13)

#CURR-MMIS-RUN-NUM (N8)

Output

New Person Indic (A1)

Error

Error Message

Main Processing

Move 'Y' to New-Person-Indic

Read ME-PERSON-LOG by LOG-BY-ENTRY from Input.Person-Number

If ME-PERSON-LOG.Person-num not = Input.Person-Number

Escape Bottom

End If

If ME-PERSON-LOG.MMIS-RUN-NUM > 0 and

ME-PERSON-LOG.MMIS-RUN-NUM not = #CURR-MMIS-RUN-NUM

Move 'N' to New Person Indic

Escape Bottom

End If

End Read

On Error

Move TRUE to Error

Move *Error-Msg to Error Message

End-Error

MMIS Reconciliation Interface - Obsolete

This is obsolete – the MMIS monthly interface replaced the reconciliation. The MEDS / MMIS Reconciliation process consists of two extract processes and a comparison and reporting step. An MMIS extract is received from the MMIS system, MEDS information is extracted to a work file, and then the comparison process matches information from the two files. Data that needs to be sent to MMIS is marked as such. Any other discrepancies are listed on a report produced by the comparison process. The format of both files will be the same as the MMIS Recipient Extract. Certain sub-record types will not be utilized as part of the reconciliation process and file extracts.

The reconciliation process uses the following rules:

- Only active AU members and clients and those closed within the last six full months will be compared. The cutoff date used will be carried on the MMIS Reconciliation Extract header record.
- All information typically on a MEDS Recipient Extract will be compared except for Community Care, Swipe Card Requests, and Parish of Residence.
- LTC segments will compare the latest three segments or the last six months' segments, whichever is the least number of records.
- Eligibility records which MEDS believes are closed and MMIS believes are open (and vice-versa) will be flagged with the next available MMIS extract run number to be re-sent to MMIS for updating. All other discrepancies will be listed in the reports.
- It is assumed by the comparison process that no MMIS records will be received without an eligibility sequence number. Any that do not have this data will be rejected as erroneous and reported.
- Data inconsistent between the two systems will be reported as well as information that exists in one system, but is missing from the other.

It is expected that the process flow will be as follows:

1. The MMIS Reconciliation Extract file will be created on the UNISYS machine;
2. MEDS will read the MMIS extract file to get the cutoff date and produce the MEDS Reconciliation extract;
3. The last step will read both of the above files and perform the appropriate comparisons, updating records where necessary and reporting discrepancies.

Reconciliation MEDS Extract Process

The extract of MEDS data for reconciling with MMIS is a similar process to the daily MMIS Recipient extract, but with slightly different criteria. Rather than looking for changed data, the MEDS Reconciliation extract collects all active and recently closed, eligible persons and writes all of their pertinent information to the work file.

Main Program Processing

Set up reports and run control

Read the MMIS Extract header to get cutoff date.

Move #MMIS.Cutoff-Date to #extract-rec.Cutoff-Date /* in Header

Move Header info to #extract-rec.#HEADER-REC

Perform WRITE-EXTRACT-REC

REPEAT-LOOP

REPEAT

Hist-AU

Histogram ME-AU-MEMBER by Person-Start-Date-9c from #Start-Person

Perform Determine-Current-AU-Mbr-Record

If #found then

Perform Write-Person

Perform Write-AU-Member

Perform Write-AU-Member-Eligibility

Perform Write-Segments /* last 3 recs or six months worth whichever is less recs

End If

Move person-num to #start-person

Add 1 to #start-person

Escape bottom (*Hist-AU*)

End-Histogram /* (*Hist-AU*)

If *Counter (*Hist-AU*) = 0

Escape bottom (*REPEAT-LOOP*.)

End-If

End-repeat /* (*REPEAT-LOOP*.)

Determine Current AU Member Record subprogram

Given a Person Number, determine which AU Member record for that person is the current one, if any, based on the MMIS Reconciliation criteria (active or closed within the last six full months).

Input

Person Number
Cutoff Date

Output

Case Number
Cert Period Number
AU member Number
#Found (L)

Data

#Pass (p3)
#Start-Key (B21) Redefined into Person-number, Start-Date-9C

Program Logic

Reset #Pass #Found #Start-Key
Move Person-Num to #START-KEY.Person-Num

REPEAT-LOOP

Repeat

 Add 1 to #Pass

MBR-READ

 Read ME-AU-MEMBER by Person-Start-Date-9C from #START-KEY

 If Person-Num not = #Start-Key.Person-Num then
 Escape Bottom (MBR-READ.)

 End-if

 If Eligible-Indic not = 'Y' then
 Escape Top

 End if

 If #Pass = 1 then /* 1st pass, look only for active records

 If Close-Date > 0 or Claims-Sus-Date > 0 then
 Escape Top

 Else

 Move True to Input.#Found
 Move Case-num, cert-period-num, au-mbr-num to Output fields
 Escape Bottom (REPEAT-LOOP.)

 End If

 Else /* 2nd pass, look only at closed records

 If close-date = 0 then
 Escape top

 Else

 If close-date >= cutoff-date

```
                Move True to Input.#found
                Move Case-num, cert-period-num, au-mbr-num to Output fields
                Escape Bottom (REPEAT-LOOP.)
            End If
        End-if
    End-if
End Read /* (MBR-READ.)
If #pass > 1 then
    Escape Bottom (REPEAT-LOOP.)
End-if
End-Repeat /* (REPEAT-LOOP.)
End
```

Write Person Record subroutine

Given a Person Number, retrieve information necessary to write the Person extract record.

Program Logic

```
CALLNAT 'MECPERU' with person-num and action = display
    If no records found
        Write ERROR message
    End-if

Set up #Extract-Rec key fields
Move by name ME-PERSON to #EXTRACT-REC
Perform Write-Extract
```

Write Extract subroutine

Write the physical work file record for the reconciliation extract.

Program Logic

```
Write work file 1 #extract-rec
Add 1 to #rec-cnt
Reset #extract-rec
```

Write AU Member Record subroutine

Given an AU Member Number, retrieve information necessary to write the extract record.

Program Logic

CALLNAT 'MECAUMU' with Au Mbr key and action = display

If no records found

 Write ERROR message

End-if

Set up #Extract-Rec key fields

Move by name ME-AU-MEMBER to #EXTRACT-REC

Get client info from ME-person, Me-case and move to #extract-rec

Get application date from ME-APPLICATION and move to #extract-rec

CALLNAT 'MEIMMPN3' /* get family size, family gross income common subpgm

Move income, family size to #extract-rec

Perform Write-extract

Write Eligibility Segments subroutine

Find all eligibility for the current person that is active or has been closed since the cutoff date.

Program Logic

/* First read all elig segments closed since the cutoff date (if any)

Move cutoff-date to #person-cutoff.cutoff-date

Move person-num to #person-cutoff.person-num

Read ME-ELIG-SEGMENT by Person-Close-Date from #person-cutoff

 If person-num ne #person-cutoff.person-num

 Escape bottom

 End-if

 Find (1) ME-CPTC with Case-cert-cptc-num = me-elig-segment.case-cert-cptc-num

 If no record found

 Write error message

 End-norec

 Move APPV-CODE to #extract-rec

 End-find

 Move key fields to #extract-rec

 Move by name ME-ELIG-SEGMENT to #Extract-Elig-Segment

 Perform write-extract

 Perform Check-For-Segments

End-read

/* Now read any eligibility segments which are not closed for this person

Move 0 to #person-cutoff.cutoff-date

Read ME-ELIG-SEGMENT by Person-Close-Date from #person-cutoff

 If person-num not = #person-cutoff.person-num or close-date > 0 then

 Escape bottom

 End-if

 Find (1) ME-CPTC with Case-cert-cptc-num = me-elig-segment.case-cert-cptc-num

 If no record found

 Write error message

 End-norec

 Move APPV-CODE to #extract-rec

 End-find

 Move key fields to #extract-rec

 Move by name ME-ELIG-SEGMENT to #Extract-Elig-Segment

Perform write-extract

Perform Check-For-Segments

End-read

Check For Segments subroutine

For the current eligibility segment's case-cert-cptc look for LTC segments that need to be written to the extract file. If found, check the array of segments to see if the current segment is more recent than those already in the array. We only send the latest three records and only that many if their close date is zero or after the cutoff date.

The following data definitions will be used:

01 #max-segs (p3) CONST <3>

01 #seg-array

02 #segments (1:#max-segs)

03 #st-cls-9c (p8)

03 #st-cccs (b25)

03 redefine #st-cccs

04 case-num (n13)

04 cert-prd-num (n4)

04 cptc-num (n4)

04 segment-num (n4)

Program Logic

Move case-cert-cptc-num from ME-ELIG-SEGMENT to #case-cert-cptc

Read ME-SEGMENT by Case-Cert-CPTC-Start-9C from #case-cert-cptc

If case, cert, or cptc-num ne #case-cert-cptc nums

Escape bottom

End if

If ME-SEGMENT.Start-Date < Cutoff-date

And ME-SEGMENT.Close-Date > 0 And ME-SEGMENT.Close-Date < Cutoff-date

Escape bottom

End if

If close-date > 0 then

#close-dt-9c = 99999999 - close-date

End if

For #I = 1 to #max-segments

If #close-date-9c < #St-cls-9c (#I) or #St-cccs (#I) = 0

If #I < #max-segments /* move entries down one to make room for new entry

```
        Move #segments(#I:#max-segments -1) to #segments (#I +1:#max-
        segments)
    End if
    Move by name ME-SEGMENT to #segments(#I)
    Move #close-date-9c to #St-cls-9c (#I)
End if
End for
End read
```

Write Segments subroutine

Using the array built in Write Eligibility Segments, write the extract records for the LTC segments.

Program Logic

/* the array #seg-array should have the #max-segs number of latest LTC segments (if any)

```
For #I = 1 to #max-segments
    If #St-cccc (#I) <1 then
        Escape bottom
    End-if

    Find (1) ME-SEGMENTS with case-cert-cptc-seg-num = #st-cccc(#I)
    If no record found
        Write error message
        Escape bottom
    End-norec
    Move key fields to #extract-rec
    Move by name ME-SEGMENT to #extract-segment
    Perform write-extract
End Find
End for
```

Reconciliation Comparison & Reporting Process - Obsolete

Once the MMIS Extract and the MEDS Extract are complete, the two files can be compared. The Comparison & Reporting process will identify discrepancies between the two systems, mark appropriate MEDS data for transmission to MMIS in the next MMIS Interface run, and report on all discrepancies.

Make sure MMIS extract and MEDS extract are using the same cutoff date and same run numbers(?).

Main Program Processing

/* Get a record from each file

Perform **READ-MEDS**

Perform **READ-MMIS**

Repeat

/* Should always have 1st record for a person from MMIS & MEDS here

/* unless either or both files are at EOF.

 If #MMIS-at-eof

 Perform **RECORD-MISSING-FROM-MMIS**

 Move true to #skip-to-next-meds-person

 Perform **READ-MEDS**

 End if

 If #MEDS-at-eof

 Perform **RECORD-MISSING-FROM-MEDS**

 Move true to #skip-to-next-mmis-person

 Perform **READ-MMIS**

 End if

 If #meds-rec.rec-type = #person then

 If #mmis-rec.rec-type = #person then

 Decide for first condition

 When #meds-rec.person-num = #mmis-rec.person-num

 Perform **COMPARE-RECORDS**

 When #meds-rec.person-num < #mmis-rec.person-num

 Perform **RECORD-MISSING-FROM-MMIS**

 Perform **READ-MEDS**

```

    When #meds-rec.person-num > #mmis-rec.person-num
        Perform RECORD-MISSING-FROM-MEDS
        Perform READ-MMIS
    When None
        Ignore
    End decide
Else
    Write error message 'Incomplete set of MMIS records for
    person number ' #mmis-rec.person-num
    Move true to #skip-to-next-MMIS-person
    Perform READ-MMIS
End if
Else
    Write error message 'Incomplete set of MEDS records for
    person number ' #meds-rec.person-num
    Move true to #skip-to-next-MEDS-person
    Perform READ-MEDS
End if
If #meds-at-eof and #mmis-at-eof
    Escape bottom
End if
End-Repeat
```

Record Missing From MMIS subroutine

Move #meds to #rpt.source

Compress 'Record missing from' #mmis into #rpt.#mmis-value

Reset #rpt.#meds-value

Perform **Write-Rpt-Line**

Record Missing From MEDS subroutine

Move #mmis to #rpt.source

Compress 'Record missing from' #meds into #rpt.#mmis-value

Reset #rpt.#mmis-value

Perform **Write-Rpt-Line**

Write Rpt Line subroutine

If #rpt.source = #meds or #rpt.source = #both then

 Move #meds-rec.person-num to #rpt.person-num

 Move #desc(#meds-rec.rec-type) to #rpt.record

Else

 Move #mmis-rec.person-num to #rpt.person-num

 Move #desc(#mmis-rec.rec-type) to #rpt.record

End if

Write (RPT) #rpt.person-num (is=on) #rpt.source (is=on)

 #rpt.record (is=on) #rpt.field

 #rpt.meds-value #rpt.mmis-value

Read MEDS subroutine

Move #meds-rec.person-num to #save-person

MEDS-REPEAT

Repeat

 Read Work File 1 #meds-rec

 If #meds-rec.rec-type = #trailer then
 Move true to #meds-at-eof
 Escape bottom /* MEDS-Repeat.

 End if

 If #skip-to-next-MEDS-person

 If #meds-rec.person-num = #save-person then
 Escape top (meds-repeat.)

 Else

 If #meds-rec.rec-type ne #person-rec

 Write an error message – person data received without a
 person record!!

 /* skip this person and go on to the next one.

 Move #meds-rec.person-num to #save-person

 else

 Reset #skip-to-next-MEDS-person

 Move true to #meds-at-new-person

 End-if

 End if

 Else

 If #meds-rec.rec-type eq #person-rec then
 Move true to #meds-at-new-person

 End if

 Escape bottom /* MEDS-Repeat.

 End-if

End-Repeat

Read MMIS subroutine

Move #mmis-rec.person-num to #save-person

MMIS-REPEAT.

Repeat

 Read Work File 2 #mmis-rec

 If #mmis-rec.rec-type = #trailer then

 Move true to #mmis-at-eof

 Escape bottom /* MMIS-Repeat.

 End if

 If #skip-to-next-MMIS-person

 If #mmis-rec.person-num = #save-person then

 Escape top (mmis-repeat.)

 Else

 If #mmis-rec.rec-type ne #person-rec

*Write an error message – person data received without a
 person record!!*

*/*Skip this person and go on to the next one.*

 Move #mmis-rec.person-num to #save-person

 Escape top (mmis-repeat.)

 else

 Reset #skip-to-next-MMIS-person

 Move true to #mmis-at-new-person

 End-if

 End if

 Else

 If #mmis-rec.rec-type eq #person-rec then

 Move true to #mmis-at-new-person

 End if

 Escape bottom /* MMIS-Repeat.

 End-if

End-Repeat

Compare Records subroutine

Move #meds-rec.person-num to #check-person

Perform **Check-Person-Fields**

Perform **Read-MEDS**

Perform **Read-MMIS**

Decide for first-condition

```

When #meds-rec.rec-type = #au-mbr
  And #mmis-rec-type = #au-mbr
  And #meds-rec.person-num = #check-person
  And #mmis-rec.person-num = #check-person

```

Perform **Check-AU-Mbr-Fields**

Perform **Read-MEDS**

Perform **Read-MMIS**

```

When #meds-rec.rec-type ne #au-mbr
  OR #meds-at-eof
  OR #meds-at-new-person

```

```

  Move #check-person to #rpt.person
  Move 'Incomplete set of data from MEDS' to #rpt.#meds-value
  Perform write-report-line
  If not (#meds-at-eof OR #meds-at-new-person) then
    If #meds-rec.person-num ne #check-person
      Move #meds-rec.person-num to #rpt.person
      Move 'Incomplete set of data from MEDS' to #rpt.#meds-value
    End if
    Move true to #skip-to-next-MEDS-person
    Move true to #skip-to-next-MMIS-person
    Perform READ-MEDS
    Perform READ-MMIS
  End if
  Escape routine

```

```

When #mmis-rec.rec-type not = #au-mbr
  OR #mmis-at-eof
  OR #mmis-at-new-person

```

```

  Move #check-person to #rpt.person
  Move 'Incomplete set of data from MMIS' to #rpt.#mmis-value
  Perform write-report-line
  If not (#mmis-at-eof OR #mmis-at-new-person) then
    If #mmis-rec.person-num not = #check-person
      Move #mmis-rec.person-num to #rpt.person
      Move 'Incomplete set of data from MMIS' to #rpt.#mmis-value
    End if
    Move true to #skip-to-next-MMIS-person
    Move true to #skip-to-next-MEDS-person

```

```

        Perform READ-MMIS
        Perform READ-MEDS
    End if
    Escape routine

End Decide

/* we have now processed the person and AU member, now do Eligibility /* & Segments

Repeat
    Decide for first condition

    When (#meds-at-eof and #mmis-at-eof)
        Or (#meds-at-eof and #mmis-at-new-person)
        Or (#mmis-at-eof and #meds-at-new-person)
        Or (#meds-at-new-person and #mmis-at-new-person)

        Escape bottom

    When (#meds-at-eof or #meds-at-new-person)
    and (#mmis-rec.rec-type = #elig
        or #mmis-rec.rec-type = #segmt)

        Perform Record-missing-from-MEDS
        Perform Read-MMIS

    When (#mmis-at-eof or #mmis-at-new-person)
    and (#meds-rec.rec-type = #elig
        or #meds-rec.rec-type = #segmt)

        Perform Record-missing-from-MEDS
        Perform Read-MMIS

    When #meds-rec.rec-type = #elig
    And #mmis-rec-type = #elig

        If #meds-rec.person-num = #check-person
            And #mmis-rec.person-num = #check-person

                Perform Check- Eligibility-Records
            Else

                Write 'incomplete set of data for person' #check-person
                Move true to #skip-to-next-MEDS-Person
                Move true to #skip-to-next-MMIS-person

            End if

        Perform Read-MEDS
        Perform Read-MMIS

    When #meds-rec.rec-type = #segmt
    And #mmis-rec-type = #segmt

        If #meds-rec.person-num = #check-person
            And #mmis-rec.person-num = #check-person

```

Perform **Check-Segment-Records**

Else

Write 'incomplete set of data for person' #check-person

Move true to #skip-to-next-MEDS-Person

Move true to #skip-to-next-MMIS-person

End if

Perform **Read-MEDS**

Perform **Read-MMIS**

End Decide

End Repeat

Check Person Fields subroutine

Move #both to #rpt.source

Decide for **EVERY** condition

```
When #meds.SSN <> #mmis.SSN
    Move 'SSN' to #rpt.Field
    Move #meds.ssn to #rpt.meds-value
    Move #mmis.ssn to #rpt.mmis-value
    Perform Write-Rpt-Line

When #meds.SSN-claim-nbr <> #mmis.SSN-claim-nbr
    Move 'SSN CLAIM NBR' to #rpt.Field
    Move #meds.ssn-claim-num to #rpt.meds-value
    Move #mmis.ssn-claim-num to #rpt.mmis-value
    Perform Write-Rpt-Line

When #meds.Last-Name <> #mmis.Last-Name
    Move 'LAST NAME' to #rpt.Field
    Move #meds.ssn to #rpt.meds-value
    Move #mmis.ssn to #rpt.mmis-value
    Perform Write-Rpt-Line

When #meds.First-Name <> #mmis.First-Name
When #meds.Middle-init <> #mmis.Middle-init
When #meds.Birth-date <> #mmis.Birth-date
When #meds.Address-Line-1 <> #mmis.Address-Line-1
When #meds.Address-Line-2 <> #mmis.Address-Line-2
When #meds.Address-Line-3 <> #mmis.Address-Line-3
When #meds.City <> #mmis.City
When #meds.State <> #mmis.State
When #meds.Zip-code <> #mmis.Zip-code
When #meds.Zip-ext <> #mmis.Zip-ext
When #meds.Phone <> #mmis.Phone
When #meds.Sex-code <> #mmis.Sex-code
When #meds.Race-code <> #mmis.Race-code
When #meds.Death-date <> #mmis.Death-date
When #meds.Title <> #mmis.Title
When #meds.Suffix <> #mmis.suffix
When #meds.Lock-in-code <> #mmis.Lock-in-code
```

End-decide

Check AU Member Fields subroutine

Move #both to #rpt.source

Decide for **EVERY** condition

When #meds.appl-date <> #mmis.appl-date

Move 'APPL DATE' to #rpt.Field

Move edited #meds.appl-date to #rpt.meds-value(em=9999/99/99)

Move edited #mmis.appl-date to #rpt.mmis-value(em=9999/99/99)

Perform Write-Rpt-Line

When #meds.client-last-name <> #mmis.client-last-name

Move 'CLIENT LAST NAME' to #rpt.Field

Move #meds.client-last-name to #rpt.meds-value

Move #mmis.client-last-name to #rpt.mmis-value

Perform Write-Rpt-Line

When #meds.client-first-Name <> #mmis.client-first-Name

Move 'CLIENT FIRST NAME' to #rpt.Field

Move #meds.client-first-name to #rpt.meds-value

Move #mmis.client-first-name to #rpt.mmis-value

Perform Write-Rpt-Line

When #meds.client-Mid-init <> #mmis.client-Mid-init

When #meds.family-income <> #mmis.family-income

When #meds.family-size <> #mmis.family-size

When #meds.Adult-indic <> #mmis.Adult-indic

When #meds.epsdt-code <> #mmis.epsdt-code

End-decide

Check Eligibility Records subroutine

Repeat

While #meds-rec.rec-type = #elig and #mmis-rec.rec-type = #elig

 Decide for first condition

 When #meds-rec.elig-seq-num = #mmis-rec.elig-seq-num

 Perform **Check-Eligibility-Fields**

 When #meds-rec.elig-seq-num LT #mmis-rec.elig-seq-num

 Perform **Record-Missing-From-MMIS**

 When #meds-rec.elig-seq-num GT #mmis-rec.elig-seq-num

 Perform **Record-Missing-From-MEDS**

 End Decide

End Repeat

Check Eligibility Fields subroutine

Move #both to #rpt.source

Decide for **EVERY** condition

When #meds.case-num <> #mmis.case-num

Move 'CASE NUM' to #rpt.Field

Move edited #meds.case-num to #rpt.meds-value (em=9(13))

Move edited #mmis.case-num to #rpt.mmis-value (em=9(13))

Perform Write-Rpt-Line

When #meds.category-code <> #mmis.category-code

Move 'CATEGORY' to #rpt.Field

Move edited #meds.category-code to #rpt.meds-value (em=999)

Move edited #mmis.category-code to #rpt.mmis-value (em=999)

Perform Write-Rpt-Line

When #meds.type-case-code <> #mmis.type-case-code

Move 'TYPE CASE' to #rpt.Field

Move edited #meds.type-case-code to #rpt.meds-value (em=999)

Move edited #mmis.type-case-code to #rpt.mmis-value (em=999)

Perform Write-Rpt-Line

When #meds.start-date <> #mmis.start-date

Move 'START DATE' to #rpt.Field

Move edited #meds.start-date to #rpt.meds-value(em=9999/99/99)

Move edited #mmis.start-date to #rpt.mmis-value(em=9999/99/99)

Perform Write-Rpt-Line

When #meds.close-date <> #mmis.close-date

Move 'CLOSE DATE' to #rpt.Field

Move edited #meds.close-date to #rpt.meds-value(em=9999/99/99)

Move edited #mmis.close-date to #rpt.mmis-value(em=9999/99/99)

Perform Write-Rpt-Line

/* If one date is zero and the other not, update mmis-run-num

If (#meds.close-date = 0 and #mmis.close-date > 0)

Or (#meds.close-date > 0 and #mmis.close-date = 0) then

CALLNAT 'MECxxxU' ---- update elig segment MMIS-run-num

End if

When #meds.appv-code <> #mmis.appv-code

Move 'APPROVAL CODE' to #rpt.Field

Move #meds.appv-code to #rpt.meds-value

Move #mmis.appv-code to #rpt.mmis-value

Perform Write-Rpt-Line

When #meds.closure-code <> #mmis.closure-code

Move 'CLOSURE CODE' to #rpt.Field

Move #meds.closure-code to #rpt.meds-value

Move #mmis.closure-code to #rpt.mmis-value

Perform Write-Rpt-Line

End-decide

Check Segment Records subroutine

Repeat

While #meds-rec.rec-type = #segment and #mmis-rec.rec-type = #segment

Decide for first condition

When #meds-rec.elig-seq-num = #mmis-rec.elig-seq-num

Perform **Check-Segment-Fields**

When #meds-rec.elig-seq-num LT #mmis-rec.elig-seq-num

Perform **Record-Missing-From-MMIS**

When #meds-rec.elig-seq-num GT #mmis-rec.elig-seq-num

Perform **Record-Missing-From-MEDS**

End Decide

End Repeat

Check Segment Fields subroutine

Move #both to #rpt.source

Decide for **EVERY** condition

When #meds.elig-seq-num <> #mmis.elig-seq-num

Move 'ELIG SEQ NUM' to #rpt.Field

Move edited #meds.elig-seq-num to #rpt.meds-value (em=99999)

Move edited #mmis.elig-seq-num to #rpt.mmis-value (em=99999)

Perform Write-Rpt-Line

When #meds.seg-start-date <> #mmis.seg-start-date

Move 'START DATE' to #rpt.Field

Move edited #meds.start-date to #rpt.meds-value(em=9999/99/99)

Move edited #mmis.start-date to #rpt.mmis-value(em=9999/99/99)

Perform Write-Rpt-Line

When #meds.seg-close-date <> #mmis.seg-close-date

Move 'CLOSE DATE' to #rpt.Field

Move edited #meds.close-date to #rpt.meds-value(em=9999/99/99)

Move edited #mmis.close-date to #rpt.mmis-value(em=9999/99/99)

Perform Write-Rpt-Line

When #meds.prov-num <> #mmis.prov-num

When #meds.level-of-care-code <> #mmis.level-of-care-code

When #meds.admission-code <> #mmis.admission-code

When #meds.admission-date <> #mmis.admission-date

When #meds.discharge-date <> #mmis.discharge-date

When #meds.PLI-amt <> #mmis.PLI-amt

When #meds.diagnosis-code-1 <> #mmis.diagnosis-code-1

Move 'PRIMARY DIAG. CODE' to #rpt.Field

Move #meds.diagnosis-code-1 to #rpt.meds-value

Move #mmis.diagnosis-code-1 to #rpt.mmis-value

Perform Write-Rpt-Line

When #meds.diagnosis-code-2 <> #mmis.diagnosis-code-2

Move 'SECNDRY DIAG. CODE' to #rpt.Field

Move #meds.diagnosis-code-2 to #rpt.meds-value

Move #mmis.diagnosis-code-2 to #rpt.mmis-value

Perform Write-Rpt-Line

End-decide

MEDS Eligibility Integrity Checker Process (MEIEICZ1) - Obsolete

This process compares AU members, certification periods, and CPTC records to the eligibility segments that are used by MEDS to create the MMIS Extract and MMIS Reconciliation files. The eligibility segments should be in sync with the rest of MEDS at all times if reconciliation and extract files are to be correct. This process checks the files to ensure this synchronization is in place and reports on any discrepancies.

Program Logic

Read all ME-AU-MEMBERS

For each AU Member, Find ME-AU-MEM-ELIG

Based on AU-MEM-ELIG, create first eligibility entry in array

Modify eligibility array based on sanctions from ME-AU-MEMBER

For each AU Member Elig, Find ME-ELIG-SEGMENTS

Compare each eligibility segment to eligibility array.

If not a perfect match, write error message.

At end of process, display record and error counts.

Appendix A - Sample Report - Obsolete

1 2 3 4 5 6 7 8 9 0 1 2 3
 123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012

XXXXXXXXX MEDS MMIS RECONCILIATION PROCESS XXXXXXXXX
 01/31/1999 DISCREPANCY REPORT 23:59:59

PERSON	SOURCE	RECORD	FIELD	MEDS VALUE	MMIS VALUE
1234567890123	MEDS	PERSON			PERSON MISSING FROM MMIS
1234567890124	MMIS	PERSON		PERSON MISSING FROM MEDS	
1234567890125	BOTH	PERSON	SSN	999-99-9999	999-99-9991
			SSN CLAIM NUMBER	0	555
			LAST NAME	TITLE	TITLE
			FIRST NAME	Y	NONE
			MIDDLE INITIAL	A	
			BIRTH DATE	99/99/9999	99/99/9999
			ADDRESS LINE 1	12345 S MAIN ST	999 W ROROROBIDEAUX BLVD
			ADDRESS LINE 2	APT 9999	SUITE 999
			ADDRESS LINE 3	C/O LSU INC	
			CITY	BATON ROUGE	RED STICK
			STATE	LA	AL
			ZIP CODE	99999	99999
			ZIP CODE EXT	1111	2222
			PHONE	999-999-9999	999-999-9999
			SEX CODE	M	
			RACE CODE	C	
			DEATH DATE		99-99-9999
			TITLE	MR	MS
			SUFFIX	JR	
			LOCK IN CODE	0	1
1234567890126	BOTH	AU MBR	CLIENT LAST NAME	TITLE	TIGER
			FAMILY SIZE	1	3
		ELIG	SEQUENCE NUMBER	244	244
			CLOSE DATE		01/09/1999
**	RE-SEND TO MMIS				
		SEGMENT	SEQUENCE NUMBER	243	243
			ADMISSION CODE	12	10
1234567890127	MMIS	SEGMENT	SEQUENCE NUMBER		RECV'D WITHOUT SEQUENCE NUMBER
	MEDS	SEGMENT	SEQUENCE NUMBER	104	NO MATCHING MMIS SEGMENT

Appendix B – MMIS Extract Logic prior to Feb 2004

The logic from the MMIS extract has changed considerably since the beginning of MEDS, and much of the old code remained commented out in the program. To make the current program more readable and less confusing the old comments and old code have been removed. Below is the version before this old code was removed and it's being kept here as a back-up. The standard routines, like initialize variables and write headings, have been removed.

```

0005 *****
0010 * Program   : MEIMMIZ1
0015 * System    : MEDS
0020 * Title     : MMIS Interface Extract
0025 * Created  : December 30, 1998
0030 * Author   : Adnan Shah
0035 * Function : This program extracts information for MMIS
0040 * NOTE     : Any changes must also be applied to full extract MEIMMIZ7 *
0045 *
0050 * Mod Date   Mod By   Description of Changes
0055 * 02/05/1999 MESPL15 Remove Eligible Indicator (CR20)
0060 * 04/30/1999 ASHAH   Reject SLMB eligibility
0065 * 05/06/1999 BRUETH  Reject Closes (> 6 mos)
0070 * 01/27/2000 ASHAH   Parish of Res (extracted on resend) SIR 203
0075 * 01/28/2000 ASHAH   Swipe Card modified for dual elig - SIR 197
0080 * 03/06/2000 ASHAH   Swipe Card modified for dual elig - SIR 219
0085 * 03/06/2000 ASHAH   Format Address Lines module modified
0090 * 03/20/2000 MESPL1  Stamp fixed dates for close code 28 (SIR 239)
0095 * 04/05/2000 MESPL1  Decode new race codes to Other (SIR 270)
0100 * 06/05/2000 MESPL1  Don't stamp fixed date on Start date (SIR 239)
0105 * 07/05/2000 MESPL1  Closure Code 28 must start on day 01 (SIR 239)
0110 * 02/14/2001 MESPL1  Add Preferred Lang to Person (SIR 398)
0115 * 04/17/2001 MESPL13 Send SLMB, Q1, Q2, QDWI in a separate file(SIR390)
0120 * 06/01/2001 MESPL13 Change AUM Parish extract to read ALL persons elig
0125 * 10/02/2001 MESPL13 Fix AUM Elig logic to write person to file(SIR529)
0130 * 11/26/2001 MESPL13 Bypass deleted segmts not yet sent to MMIS(SIR535)
0135 * 01/04/2002 MESPL13 Send new MSIS race codes to MMIS (SIR 547)
0140 * 03/06/2002 MESPL13 Send new appv code & closure code for eligibility
0145 *          changes due to start/end date changes (SIR 544)
0150 * 05/13/2002 MESPL13 Add Hospice Segments to segment extract (SIR 593)
0155 * 07/26/2002 MESPL13 Simplify Swipe Card extract to send all with
0160 *          current run number (SIR 653)
0165 * 09/26/2002 MESPL9  Remove check for Person Rec write in Swipe Card
0170 *          to prevent not writing one at all (SIR686)
0175 * 10/30/2002 MESPL27 Backed out SIR508
0180 *
0185 *****
0190 *
1835 *****
1840 * Main Program Logic
1845 *****
1850 *
1855 PERFORM INITIALIZE-VARIABLES
1860 PERFORM HEADER-RECORD
1865 PERFORM PERSON-EXTRACT
1870 PERFORM SWIPE-CARD-EXTRACT
1875 PERFORM LOCK-IN-EXTRACT
1880 PERFORM AU-MEMBER-EXTRACT
1885 PERFORM AU-MEMBER-PARISH-OF-RESIDENCE-EXTRACT
1890 PERFORM AU-MEMBER-ELIGIBILITY-EXTRACT
1895 PERFORM SEGMENT-EXTRACT
1900 PERFORM TRAILER-RECORD
1905 PERFORM FINAL-PROCESSING
1910 *
1915 MOVE #RUN-CONTROL.RESTART-DATA TO ME-RUN-CNTL.RESTART-DATA
1920 *
1925 * Run Control Interim Update

```

```

1930 PERFORM ISSUE-PERIODIC-ET
1935 *
2880 *****
2885 DEFINE SUBROUTINE PERSON-EXTRACT /* PE
2890 *****
2895 *
2900 * Extracts records from the ME-PERSON file
2905 *
2910 ASSIGN #MMIS-RUN-PERSON-NUM.MMIS-RUN-NUM = #RUN-CONTROL.RESTART-RUN-NUM
2915 *
2920 READ-PER.
2925 READ ME-PERSON BY MMIS-RUN-PERSON-NUM FROM #MMIS-RUN-PERSON-NUM
2930 *
2935 IF #MMIS-RUN-PERSON-NUM.MMIS-RUN-NUM NE ME-PERSON.MMIS-RUN-NUM
2940 ESCAPE BOTTOM(READ-PER.)
2945 END-IF
2950 *
2955 MOVE ME-PERSON.PERSON-NUM TO #PERSON-START-DATE-9C.PERSON-NUM
2960 ASSIGN #LOGIC.#PERSON-ON-MAIN-FILE = FALSE
2965 ASSIGN #LOGIC.#PERSON-ON-ALT-FILE = FALSE
2970 *
2975 READ-AU.
2980 READ ME-AU-MEMBER BY PERSON-START-DATE-9C FROM #PERSON-START-DATE-9C
2985 /*
2990 IF #PERSON-START-DATE-9C.PERSON-NUM NE ME-AU-MEMBER.PERSON-NUM
2995 ESCAPE BOTTOM /* (READ-AU.)
3000 END-IF
3005 /*
3010 MOVE BY NAME ME-AU-MEMBER TO #ME-CERT-PERIOD
3015 PERFORM GET-CERT-PERIOD
3020 /*
3025 PERFORM PERSON-DETAIL
3030 /*
3035 IF #ME-CERT-PERIOD.TYPE-CASE = #CONSTANT.#ALTERNATE-TYPE-CASE(*)
3040 ADD 1 TO #RUN-CONTROL.#ALT-PERSON-COUNT
3045 PERFORM WRITE-ALT-WORK-FILE
3050 ASSIGN #LOGIC.#PERSON-ON-ALT-FILE = TRUE
3055 ELSE
3060 ADD 1 TO #RUN-CONTROL.#PERSON-COUNT
3065 PERFORM WRITE-WORK-FILE
3070 ASSIGN #LOGIC.#PERSON-ON-MAIN-FILE = TRUE
3075 END-IF
3080 RESET #EXTRACT-REC
3085 /*
3090 /* if we've extracted the person to both files, move on to next pers
3095 IF #LOGIC.#PERSON-ON-MAIN-FILE AND
3100 #LOGIC.#PERSON-ON-ALT-FILE
3105 ESCAPE BOTTOM /*( READ-AU.)
3110 END-IF
3115 /*
3120 END-READ /* (READ-AU.)
3125 *
3130 END-READ /* READ-PER.
3135 *
3140 END-SUBROUTINE /* PERSON-EXTRACT
3145 *
3150 *****
3155 DEFINE SUBROUTINE SWIPE-CARD-EXTRACT /* SCE
3160 *****
3165 *
3170 * Extracts records from the ME-SWIPE-CARD-REQST file
3175 *
3180 ASSIGN #MMIS-PERSON-REQUEST-DATE.MMIS-RUN-NUM
3185 = #RUN-CONTROL.RESTART-RUN-NUM
3190 *
3195 READ-SCR.
3200 READ ME-SWIPE-CARD-REQST BY MMIS-PERSON-REQUEST-DATE
3205 FROM #MMIS-PERSON-REQUEST-DATE
3210 /*
3215 IF #MMIS-PERSON-REQUEST-DATE.MMIS-RUN-NUM
3220 NE ME-SWIPE-CARD-REQST.MMIS-RUN-NUM

```

```

3225     ESCAPE BOTTOM (READ-SCR.)
3230 END-IF
3235 /*
3240 /* SIR 653 start: rely on on-line logic to correctly flag what to send
3245 PERFORM WRITE-SWIPE-CARD-TO-EXTRACT
3250 *
3255 * /* Swipe card requests are never generated for type cases 40,48,49,94.
3260 * /* Only include swipe cards for these if there is another active type
3265 * /* case. Only write 1 swipe card record per person.
3270 * ASSIGN #LOCAL.#AU-MEM-CLOSE-DATE = 0
3275 * ASSIGN #PERSON-CLOSE9C-START9C-CASE-CERT.PERSON-NUM
3280 *           = ME-SWIPE-CARD-REQST.PERSON-NUM
3285 * /* this will read open AUMs first
3290 * READ-AU-SWIPE.
3295 * READ (2) ME-AU-MEMBER BY PERSON-CLOSE9C-START9C-CASE-CERT
3300 *           STARTING FROM #PERSON-CLOSE9C-START9C-CASE-CERT
3305 * IF #PERSON-CLOSE9C-START9C-CASE-CERT.PERSON-NUM
3310 *           NE ME-AU-MEMBER.PERSON-NUM
3315 *           ESCAPE BOTTOM (READ-AU-SWIPE.)
3320 * END-IF
3325 * /*
3330 * MOVE BY NAME ME-AU-MEMBER TO #ME-CERT-PERIOD
3335 * PERFORM GET-CERT-PERIOD /* to find type case
3340 * IF #ME-CERT-PERIOD.TYPE-CASE = #CONSTANT.#ALTERNATE-TYPE-CASE(*)
3345 *           ASSIGN #LOCAL.#AU-MEM-CLOSE-DATE = ME-AU-MEMBER.CLOSE-DATE
3350 *           ESCAPE TOP /* (READ-AU-SWIPE.)
3355 * END-IF
3360 * /*
3365 * IF *COUNTER(READ-AU-SWIPE.) = 2
3370 *           /* No swipe card written (the 1st AUM must have been an alternate
3375 *           /* type case). If this AUM is active, or closed when the other
3380 *           /* AUM closed (i.e most recent elig was overlapping) extract it.
3385 *           IF ME-AU-MEMBER.CLOSE-DATE = #LOCAL.#AU-MEM-CLOSE-DATE OR
3390 *           ME-AU-MEMBER.CLOSE-DATE > *DATN
3395 *           PERFORM WRITE-SWIPE-CARD-TO-EXTRACT
3400 *           END-IF
3405 * ELSE
3410 *           PERFORM WRITE-SWIPE-CARD-TO-EXTRACT
3415 * END-IF
3420 * ESCAPE BOTTOM (READ-AU-SWIPE.) /* escape when a record is written
3425 * END-READ /*(READ-AU-SWIPE.) - SIR 653 end
3430 /*
3435 END-READ /* (READ-PER.)
3440 *
3445 END-SUBROUTINE /* SWIPE-CARD-EXTRACT
3450 *
3455 *****
3460 DEFINE SUBROUTINE WRITE-SWIPE-CARD-TO-EXTRACT
3465 *****
3470 *
3475 ASSIGN #EXTRACT-REC.#RECORD-TYPE           = #MEIMMIL2.#DETAIL
3480 ASSIGN #EXTRACT-REC.#SUB-RECORD-TYPE       = #MEIMMIL2.#SWIPE-CARD
3485 ASSIGN #EXTRACT-REC.#RECORD-SEQUENCE      = 0
3490 ASSIGN #EXTRACT-REC.#PERSON-NUMBER        = ME-SWIPE-CARD-REQST.PERSON-NUM
3495 ASSIGN #EXTRACT-REC.#PERSON-SEQUENCE      = ME-SWIPE-CARD-REQST.REQUEST-DATE
3500 *
3505 MOVE BY NAME ME-SWIPE-CARD-REQST TO #EXTRACT-REC.#SWIPE-CARD-EXTRACT
3510 ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
3515 *
3520 * SIR 653 part 2
3525 * Swipe card requests are never generated for type cases 40,48,49,94.
3530 * IF #ME-CERT-PERIOD.TYPE-CASE = #CONSTANT.#ALTERNATE-TYPE-CASE(*)
3535 *           ADD 1 TO #RUN-CONTROL.#ALT-SWIPE-CARD-COUNT
3540 *           PERFORM WRITE-ALT-WORK-FILE
3545 * ELSE
3550 ADD 1 TO #RUN-CONTROL.#SWIPE-CARD-COUNT
3555 PERFORM WRITE-WORK-FILE
3560 * END-IF /* SIR 653 end
3565 RESET #EXTRACT-REC
3570 *
3575 * Write Person Record if havent already - Changed

```

```

3580 * SIR686 - Always write Person record, as it will be sorted out later;
3585 * to prevent writing Swipe Card rec only for SLMB cases (Type Case 40)
3590 * that overlap with Type cases that are Swipe Card Eligibile
3595 *
3600 FIND-PERS-SWIPE.
3605 FIND (1) ME-PERSON WITH PERSON-NUM = ME-SWIPE-CARD-REQST.PERSON-NUM
3610 /*
3615 * IF ME-PERSON.MMIS-RUN-NUM NE #RUN-CONTROL.RESTART-RUN-NUM /* SIR686
3620 PERFORM PERSON-DETAIL
3625 ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
3630 * IF #ME-CERT-PERIOD.TYPE-CASE = #ALTERNATE-TYPE-CASE(*) /* SIR 653
3635 * ADD 1 TO #RUN-CONTROL.#ALT-EXTRA-PERS-SWIPE-COUNT
3640 * PERFORM WRITE-ALT-WORK-FILE
3645 * ELSE
3650 ADD 1 TO #RUN-CONTROL.#EXTRA-PERS-SWIPE-COUNT
3655 PERFORM WRITE-WORK-FILE
3660 * END-IF /* SIR 653 end - SIR686: commented out!
3665 RESET #EXTRACT-REC
3670 * END-IF
3675 /*
3680 END-FIND /* (FIND-PERS-SWIPE.)
3685 *
3690 END-SUBROUTINE /* WRITE-SWIPE-CARD-TO-EXTRACT
3695 *
3700 *****
3705 DEFINE SUBROUTINE LOCK-IN-EXTRACT /* LIE
3710 *****
3715 *
3720 RESET #LOCAL.#SAVE-PERSON
3725 #LOCAL.#SAVE-PROV
3730 #LOCAL.#SAVE-TYPE
3735 *
3740 ASSIGN #MMIS-PERSON-PROV-START-DATE.MMIS-RUN-NUM
3745 = #RUN-CONTROL.RESTART-RUN-NUM
3750 *
3755 LOCKIN-HIST.
3760 HISTOGRAM ME-PERSON-PROV-LOCK-IN-HIST-VIEW FOR
3765 MMIS-PERSON-PROV-START-DATE FROM #MMIS-PERSON-PROV-START-DATE
3770 *
3775 IF ME-PERSON-PROV-LOCK-IN-HIST-VIEW.#MMIS-RUN-NUM
3780 NE #MMIS-PERSON-PROV-START-DATE.MMIS-RUN-NUM
3785 ESCAPE BOTTOM (LOCKIN-HIST.)
3790 END-IF
3795 *
3800 IF #LOCAL.#SAVE-PERSON NE ME-PERSON-PROV-LOCK-IN-HIST-VIEW.#PERSON-NUM
3805 ASSIGN #LOCAL.#SAVE-PERSON =
3810 ME-PERSON-PROV-LOCK-IN-HIST-VIEW.#PERSON-NUM
3815 *
3820 MOVE #LOCAL.#SAVE-PERSON TO #PERSON-START-DATE-9C.PERSON-NUM
3825 RESET #LOGIC.#PERSON-ON-MAIN-FILE
3830 #LOGIC.#PERSON-ON-ALT-FILE
3835 /*
3840 /* Determine which file based on eligibilty to write to - can
3845 /* write to both files.
3850 READ-AU-LOCK.
3855 READ ME-AU-MEMBER BY PERSON-START-DATE-9C FROM #PERSON-START-DATE-9C
3860 /*
3865 IF #PERSON-START-DATE-9C.PERSON-NUM NE ME-AU-MEMBER.PERSON-NUM
3870 ESCAPE BOTTOM(READ-AU-LOCK.)
3875 END-IF
3880 /*
3885 MOVE BY NAME ME-AU-MEMBER TO #ME-CERT-PERIOD
3890 PERFORM GET-CERT-PERIOD
3895 /*
3900 IF #ME-CERT-PERIOD.TYPE-CASE = #CONSTANT.#ALTERNATE-TYPE-CASE(*)
3905 ASSIGN #LOGIC.#PERSON-ON-ALT-FILE = TRUE
3910 ELSE
3915 ASSIGN #LOGIC.#PERSON-ON-MAIN-FILE = TRUE
3920 END-IF
3925 /*
3930 IF #LOGIC.#PERSON-ON-MAIN-FILE AND #LOGIC.#PERSON-ON-ALT-FILE

```

```

3935         ESCAPE BOTTOM(READ-AU-LOCK.)
3940         END-IF
3945     END-READ /* (READ-AU-LOCK.)
3950 *
3955     ASSIGN #PERSON-PROV-START-DATE.PERSON-NUM = #LOCAL.#SAVE-PERSON
3960     LOCKIN-READ.
3965     READ ME-PERSON-PROV-LOCK-IN BY PERSON-PROV-START-DATE
3970     FROM #PERSON-PROV-START-DATE
3975 *
3980     IF ME-PERSON-PROV-LOCK-IN.PERSON-NUM NE
3985         #PERSON-PROV-START-DATE.PERSON-NUM
3990         ESCAPE BOTTOM (LOCKIN-READ.)
3995     END-IF
4000 *
4005     IF (ME-PERSON-PROV-LOCK-IN.LOCK-IN-STATUS = 'A') AND
4010         (ME-PERSON-PROV-LOCK-IN.CLOSE-DATE = 0 OR
4015     ME-PERSON-PROV-LOCK-IN.MMIS-RUN-NUM =
4020         #RUN-CONTROL.RESTART-RUN-NUM)
4025 *
4030     ASSIGN #EXTRACT-REC.#RECORD-TYPE = #MEIMMIL2.#DETAIL
4035     ASSIGN #EXTRACT-REC.#SUB-RECORD-TYPE =
4040         #MEIMMIL2.#LOCK-IN-COMTY-CARE
4045     ASSIGN #EXTRACT-REC.#RECORD-SEQUENCE = 0
4050     ASSIGN #EXTRACT-REC.#PERSON-NUMBER =
4055         ME-PERSON-PROV-LOCK-IN.PERSON-NUM
4060     ASSIGN #EXTRACT-REC.#PERSON-SEQUENCE =
4065         ME-PERSON-PROV-LOCK-IN.START-DATE
4070 *
4075     MOVE BY NAME ME-PERSON-PROV-LOCK-IN
4080     TO #EXTRACT-REC.#LOCK-IN-EXTRACT
4085     ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
4090 *
4095     IF ME-PERSON-PROV-LOCK-IN.PROV-NUM NE #LOCAL.#SAVE-PROV
4100     ASSIGN #LOCAL.#SAVE-PROV = ME-PERSON-PROV-LOCK-IN.PROV-NUM
4105 *
4110     PROVIDER-FIND.
4115     FIND (1) ME-PROVIDER WITH PROV-NUM = #LOCAL.#SAVE-PROV
4120     MOVE ME-PROVIDER.PROV-TYPE TO #LOCAL.#SAVE-TYPE
4125     END-FIND /* (PROVIDER-FIND.)
4130     END-IF
4135 *
4140     MOVE #LOCAL.#SAVE-TYPE TO #EXTRACT-REC.PROV-TYPE
4145     /*
4150     IF #LOGIC.#PERSON-ON-ALT-FILE
4155     ADD 1 TO #RUN-CONTROL.#ALT-LOCK-IN-COUNT
4160     PERFORM WRITE-ALT-WORK-FILE
4165     END-IF
4170     IF #LOGIC.#PERSON-ON-MAIN-FILE
4175     ADD 1 TO #RUN-CONTROL.#LOCK-IN-COUNT
4180     PERFORM WRITE-WORK-FILE
4185     END-IF
4190     RESET #EXTRACT-REC
4195     END-IF /* (4005)
4200     END-READ /* (LOCKIN-READ.)
4205     /*
4210     /* Write Person Record if havent already
4215     /*
4220     FIND-PERS-LOCK.
4225     FIND (1) ME-PERSON WITH PERSON-NUM = #LOCAL.#SAVE-PERSON
4230     /*
4235     IF ME-PERSON.MMIS-RUN-NUM NE #RUN-CONTROL.RESTART-RUN-NUM
4240     PERFORM PERSON-DETAIL
4245     ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
4250     IF #LOGIC.#PERSON-ON-ALT-FILE
4255     ADD 1 TO #RUN-CONTROL.#ALT-EXTRA-PERS-LOCK-COUNT
4260     PERFORM WRITE-ALT-WORK-FILE
4265     END-IF
4270     IF #LOGIC.#PERSON-ON-MAIN-FILE
4275     ADD 1 TO #RUN-CONTROL.#EXTRA-PERS-LOCK-COUNT
4280     PERFORM WRITE-WORK-FILE
4285     END-IF

```



```

4290         RESET #EXTRACT-REC
4295         END-IF
4300         END-FIND /* (FIND-PERS-LOCK.)
4305     END-IF /* (3800) change in person
4310 END-HISTOGRAM /* (LOCKIN-HIST.)
4315 *
4320 END-SUBROUTINE /* LOCK-IN-EXTRACT
4325 *
4330 *****
4335 DEFINE SUBROUTINE AU-MEMBER-EXTRACT /* AME
4340 *****
4345 *
4350 RESET #LOCAL.#APPL-NUM
4355     #LOCAL.#CLIENT-NUM
4360     #LOCAL.#SAVE-CASE
4365     #LOCAL.#SAVE-CERT
4370 *
4375 * Extracts records from the ME-AU-MEMBER file
4380 *
4385 ASSIGN #MMIS-CASE-CERT-AU-MBR.MMIS-RUN-NUM
4390     = #RUN-CONTROL.RESTART-RUN-NUM
4395 *
4400 AUMBR-READ.
4405 READ ME-AU-MEMBER BY MMIS-CASE-CERT-AU-MBR #MMIS-CASE-CERT-AU-MBR
4410 *
4415     IF #MMIS-CASE-CERT-AU-MBR.MMIS-RUN-NUM NE ME-AU-MEMBER.MMIS-RUN-NUM
4420         ESCAPE BOTTOM(AUMBR-READ.)
4425     END-IF
4430 *
4435 * Read Log file and write a Parish(5) record for newly added AU Member
4440 *
4445     ASSIGN #LOG-BY-ENTRY.CASE-NUM         = ME-AU-MEMBER.CASE-NUM
4450     ASSIGN #LOG-BY-ENTRY.CERT-PERIOD-NUM = ME-AU-MEMBER.CERT-PERIOD-NUM
4455     ASSIGN #LOG-BY-ENTRY.AU-MEMBER-NUM  = ME-AU-MEMBER.AU-MEMBER-NUM
4460 *
4465     READ-AUM-LOG.
4470     READ (1) ME-AU-MEMBER-LOG BY LOG-BY-ENTRY
4475         STARTING FROM #LOG-BY-ENTRY
4480 *
4485     IF ME-AU-MEMBER-LOG.CASE-NUM NE #LOG-BY-ENTRY.CASE-NUM
4490     OR ME-AU-MEMBER-LOG.CERT-PERIOD-NUM NE #LOG-BY-ENTRY.CERT-PERIOD-NUM
4495     OR ME-AU-MEMBER-LOG.AU-MEMBER-NUM NE #LOG-BY-ENTRY.AU-MEMBER-NUM
4500         ESCAPE BOTTOM /* (READ-AUM-LOG.)
4505     END-IF
4510 *
4515     IF ME-AU-MEMBER-LOG.MMIS-RUN-NUM = #RUN-CONTROL.RESTART-RUN-NUM
4520         PERFORM WRITE-PARISH-RECORD
4525     END-IF
4530 *
4535 END-READ /* (READ-AUM-LOG.)
4540 *
4545     ASSIGN #EXTRACT-REC.#RECORD-TYPE     = #MEIMMIL2.#DETAIL
4550     ASSIGN #EXTRACT-REC.#SUB-RECORD-TYPE = #MEIMMIL2.#AU-MEMBER
4555     ASSIGN #EXTRACT-REC.#RECORD-SEQUENCE = 0
4560     ASSIGN #EXTRACT-REC.#PERSON-NUMBER  = ME-AU-MEMBER.PERSON-NUM
4565     ASSIGN #EXTRACT-REC.#PERSON-SEQUENCE = 1
4570 *
4575     ASSIGN #CASE-CERT-PERIOD-NUM.CASE-NUM = ME-AU-MEMBER.CASE-NUM
4580     ASSIGN #CASE-CERT-PERIOD-NUM.CERT-PERIOD-NUM
4585         = ME-AU-MEMBER.CERT-PERIOD-NUM
4590 *
4595     CERT-FIND.
4600     FIND (1) ME-CERT-PERIOD WITH CASE-CERT-PERIOD-NUM
4605         = #CASE-CERT-PERIOD-NUM
4610         ASSIGN #LOCAL.#APPL-NUM     = ME-CERT-PERIOD.APPL-NUM
4615         ASSIGN #LOCAL.#CLIENT-NUM  = ME-CERT-PERIOD.CLIENT-NUM
4620     END-FIND /* (CERT-FIND.)
4625 *
4630     ASSIGN #CASE-CERT-CPTC-NUM.CASE-NUM = ME-AU-MEMBER.CASE-NUM
4635     ASSIGN #CASE-CERT-CPTC-NUM.CERT-PERIOD-NUM
4640         = ME-AU-MEMBER.CERT-PERIOD-NUM

```

```

4645  ASSIGN #CASE-CERT-CPTC-NUM.CPTC-NUM = 1
4650 *
4655  TCAS-FIND1.
4660  FIND (1) ME-CPTC WITH CASE-CERT-CPTC-NUM = #CASE-CERT-CPTC-NUM
4665  ASSIGN #LOCAL.#CAT-TYPE-CASE = ME-CPTC.CAT-TYPE-CASE
4670  END-FIND /* (TCAS-FIND1.)
4675 *
4680  CLIENT-FIND.
4685  FIND (1) ME-PERSON WITH PERSON-NUM = #LOCAL.#CLIENT-NUM
4690  ASSIGN #EXTRACT-REC.#CLIENT-FIRST-NAME = ME-PERSON.FIRST-NAME
4695  ASSIGN #EXTRACT-REC.#CLIENT-LAST-NAME = ME-PERSON.LAST-NAME
4700  ASSIGN #EXTRACT-REC.#CLIENT-MIDDLE-INIT= ME-PERSON.MIDDLE-INIT
4705  END-FIND /* (CLIENT-FIND.)
4710 *
4715  ASSIGN #CASE-NUM-APPL-NUM.CASE-NUM = ME-CERT-PERIOD.CASE-NUM
4720  ASSIGN #CASE-NUM-APPL-NUM.APPL-NUM = ME-CERT-PERIOD.APPL-NUM
4725 *
4730  APPL-FIND.
4735  FIND (1) ME-APPLICATION WITH CASE-NUM-APPL-NUM = #CASE-NUM-APPL-NUM
4740  MOVE ME-APPLICATION.APPL-DATE TO #EXTRACT-REC.APPL-DATE
4745 *
4750  CALLNAT 'MECAPNN1'          /** Derive Application Date
4755  ME-APPLICATION.CASE-NUM
4760  #CASE-CERT-CPTC-NUM.CERT-PERIOD-NUM
4765  #EXTRACT-REC.APPL-DATE
4770 *
4775  END-FIND /* (APPL-FIND.)
4780 *
4785  MOVE BY NAME ME-AU-MEMBER TO #EXTRACT-REC.#AU-MEMBER-EXTRACT
4790 *
4795  IF ME-AU-MEMBER.CASE-NUM      NE #LOCAL.#SAVE-CASE OR
4800  ME-AU-MEMBER.CERT-PERIOD-NUM NE #LOCAL.#SAVE-CERT
4805  ASSIGN #LOCAL.#SAVE-CASE = ME-AU-MEMBER.CASE-NUM
4810  ASSIGN #LOCAL.#SAVE-CERT = ME-AU-MEMBER.CERT-PERIOD-NUM
4815 *
4820 * Calculate Family Size and Gross Family Income
4825 *
4830  MOVE #LOCAL.#SAVE-CASE TO #MEIMPN3-IN.CASE-NUM
4835  MOVE #LOCAL.#SAVE-CERT TO #MEIMPN3-IN.CERT-PERIOD-NUM
4840 *
4845  CALLNAT 'MEIMPN3'          /* Calc Family Size and Gross Family Income
4850  #MEIMPN3-IN /* MEIMPP3
4855  #MEIMPN3-OUT
4860  #CXXGNLPL-OUT /* Standard Error Handler
4865  END-IF
4870 *
4875  MOVE BY NAME #MEIMPN3-OUT TO #EXTRACT-REC.#AU-MEMBER-EXTRACT
4880  ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
4885 *
4890  IF #LOCAL.#TYPE-CASE = #CONSTANT.#ALTERNATE-TYPE-CASE(*)
4895  ADD 1 TO #RUN-CONTROL.#ALT-AU-MEMBER-COUNT
4900  PERFORM WRITE-ALT-WORK-FILE
4905  ELSE
4910  ADD 1 TO #RUN-CONTROL.#AU-MEMBER-COUNT
4915  PERFORM WRITE-WORK-FILE
4920  END-IF
4925  RESET #EXTRACT-REC
4930  /*
4935  /* Write Person Record if havent already
4940  /*
4945  FIND-PERSON.
4950  FIND (1) ME-PERSON WITH PERSON-NUM = ME-AU-MEMBER.PERSON-NUM
4955  IF ME-PERSON.MMIS-RUN-NUM NE #RUN-CONTROL.RESTART-RUN-NUM
4960  PERFORM PERSON-DETAIL
4965  ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
4970  IF #LOCAL.#TYPE-CASE = #CONSTANT.#ALTERNATE-TYPE-CASE(*)
4975  ADD 1 TO #RUN-CONTROL.#ALT-EXTRA-PERS-AU-COUNT
4980  PERFORM WRITE-ALT-WORK-FILE
4985  ELSE
4990  ADD 1 TO #RUN-CONTROL.#EXTRA-PERS-AU-COUNT
4995  PERFORM WRITE-WORK-FILE

```

```

5000     END-IF
5005     RESET #EXTRACT-REC
5010     END-IF
5015     END-FIND /* (FIND-PERSON.)
5020     /*
5025     RESET #LOCAL.#CAT-TYPE-CASE
5030     *
5035     END-READ /* (AUMBR-READ.)
5040     *
5045     END-SUBROUTINE /* AU-MEMBER-EXTRACT
5050     *
5055     *****
5060     DEFINE SUBROUTINE AU-MEMBER-PARISH-OF-RESIDENCE-EXTRACT /* AMPORE
5065     *****
5070     *
5075     * Extracts records from the ME-AU-MEMBER file
5080     *
5085     ASSIGN #MMIS-CASE.MMIS-RUN-NUM = #RUN-CONTROL.RESTART-RUN-NUM
5090     *
5095     LOOP1.
5100     READ ME-CASE BY MMIS-CASE FROM #MMIS-CASE
5105     *
5110     IF #MMIS-CASE.MMIS-RUN-NUM NE ME-CASE.MMIS-RUN-NUM
5115         ESCAPE BOTTOM(LOOP1.)
5120     END-IF
5125     *
5130     ASSIGN #CASE-NUM-PERSON-NUM.CASE-NUM = ME-CASE.CASE-NUM
5135     *
5140     LOOP2.
5145     READ ME-CASE-MEMBER BY CASE-NUM-PERSON-NUM
5150         FROM #CASE-NUM-PERSON-NUM
5155     *
5160     IF #CASE-NUM-PERSON-NUM.CASE-NUM NE ME-CASE-MEMBER.CASE-NUM
5165         ESCAPE BOTTOM(LOOP2.)
5170     END-IF
5175     *
5180     ASSIGN #PERSON-START-DATE-9C.PERSON-NUM = ME-CASE-MEMBER.PERSON-NUM
5185     ASSIGN #LOGIC.#PARISH-ON-MAIN-FILE = FALSE
5190     *
5195     LOOP3.
5200     READ ME-AU-MEMBER BY PERSON-START-DATE-9C FROM #PERSON-START-DATE-9C
5205     *
5210     IF #PERSON-START-DATE-9C.PERSON-NUM NE ME-AU-MEMBER.PERSON-NUM
5215         ESCAPE BOTTOM(LOOP3.)
5220     END-IF
5225     *
5230     IF ME-AU-MEMBER.CASE-NUM NE ME-CASE.CASE-NUM
5235         ESCAPE TOP /* (LOOP3.)
5240     END-IF
5245     *
5250     * We only extract "active" eligibility, retroactive eligibility
5255     * parish of residence records will be created in the AU Member
5260     * extract code.
5265     *
5270     IF (ME-AU-MEMBER.CLOSE-DATE > 0 AND
5275         ME-AU-MEMBER.CLOSE-DATE < *DATN) OR
5280         (ME-AU-MEMBER.CLAIMS-SUS-DATE > 0 AND
5285         ME-AU-MEMBER.CLAIMS-SUS-DATE < *DATN)
5290     /*
5295     /* Parish Record will be Extracted on a Resend even if the
5300     /* Cert is closed in the Past. Person Record must be stamped for
5305     /* Resend in the same run as well. 12/29/99 MESPL23
5310     /*
5315     FIND (1) ME-PERSON WITH PERSON-NUM = ME-AU-MEMBER.PERSON-NUM
5320     END-FIND
5325     IF ME-PERSON.MMIS-RUN-NUM = #RUN-CONTROL.RESTART-RUN-NUM AND
5330         ME-CASE.MMIS-RUN-NUM = #RUN-CONTROL.RESTART-RUN-NUM
5335         IGNORE
5340     ELSE
5345         ESCAPE TOP /* (LOOP3.) read ALL elig for the person
5350     END-IF

```

```

5355     END-IF
5360 *
5365     ASSIGN #EXTRACT-REC.#RECORD-TYPE      = #MEIMMIL2.#DETAIL
5370     ASSIGN #EXTRACT-REC.#SUB-RECORD-TYPE = #MEIMMIL2.#AU-MEMBER-PARISH
5375     ASSIGN #EXTRACT-REC.#RECORD-SEQUENCE = 0
5380     ASSIGN #EXTRACT-REC.#PERSON-NUMBER   = ME-CASE-MEMBER.PERSON-NUM
5385 *
5390     ASSIGN #CASE-CERT-CPTC-NUM.CASE-NUM = ME-AU-MEMBER.CASE-NUM
5395     ASSIGN #CASE-CERT-CPTC-NUM.CERT-PERIOD-NUM
5400     = ME-AU-MEMBER.CERT-PERIOD-NUM
5405     ASSIGN #CASE-CERT-CPTC-NUM.CPTC-NUM = 1
5410 *
5415     TCAS-FIND2.
5420     FIND (1) ME-CPTC WITH CASE-CERT-CPTC-NUM = #CASE-CERT-CPTC-NUM
5425     ASSIGN #LOCAL.#CAT-TYPE-CASE = ME-CPTC.CAT-TYPE-CASE
5430     END-FIND /* (TCAS-FIND2.)
5435 *
5440     FOR #LOCAL.#I = 1 TO C*PARISH-RES-HISTORY
5445     MOVE ME-CASE.PARISH-START-DATE(#I) TO
5450     #EXTRACT-REC.PARISH-START-DATE
5455     #EXTRACT-REC.#PERSON-SEQUENCE
5460     MOVE ME-CASE.PARISH-LOC-ID(#I)      TO
5465     #EXTRACT-REC.PARISH-LOC-ID
5470     ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
5475 *
5480     IF #LOCAL.#TYPE-CASE = #CONSTANT.#ALTERNATE-TYPE-CASE(*)
5485     ADD 1 TO #RUN-CONTROL.#ALT-AU-MEMBER-PARISH-COUNT
5490     PERFORM WRITE-ALT-WORK-FILE
5495     ELSE
5500     ADD 1 TO #RUN-CONTROL.#AU-MEMBER-PARISH-COUNT
5505     PERFORM WRITE-WORK-FILE
5510     ASSIGN #LOGIC.#PARISH-ON-MAIN-FILE = TRUE
5515     END-IF
5520 *
5525     END-FOR
5530     /*
5535     /* Write Person Record if havent already
5540     /*
5545     FIND-PERS-PARISH.
5550     FIND (1) ME-PERSON WITH PERSON-NUM = ME-AU-MEMBER.PERSON-NUM
5555     /*
5560     IF ME-PERSON.MMIS-RUN-NUM NE #RUN-CONTROL.RESTART-RUN-NUM
5565     RESET #EXTRACT-REC
5570     PERFORM PERSON-DETAIL
5575     ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
5580     IF #LOCAL.#TYPE-CASE = #CONSTANT.#ALTERNATE-TYPE-CASE(*)
5585     ADD 1 TO #RUN-CONTROL.#ALT-EXTRA-PERS-PARISH-COUNT
5590     PERFORM WRITE-ALT-WORK-FILE
5595     ELSE
5600     ADD 1 TO #RUN-CONTROL.#EXTRA-PERS-PARISH-COUNT
5605     PERFORM WRITE-WORK-FILE
5610     END-IF
5615     RESET #EXTRACT-REC
5620     END-IF
5625     /*
5630     END-FIND /* (FIND-PERS-PARISH.)
5635     /*
5640     RESET #EXTRACT-REC
5645     #LOCAL.#CAT-TYPE-CASE
5650     /* keep reading if we haven't yet found elig for the main file
5655     IF #LOGIC.#PARISH-ON-MAIN-FILE
5660     ESCAPE BOTTOM /* (LOOP3.)
5665     END-IF
5670 *
5675     END-READ /* (LOOP3.)
5680     END-READ /* (LOOP2.)
5685     END-READ /* (LOOP1.)
5690 *
5695     END-SUBROUTINE /* AU-MEMBER-PARISH-OF-RESIDENCE-EXTRACT
5700 *
5705     *****

```

```

5710 DEFINE SUBROUTINE AU-MEMBER-ELIGIBILITY-EXTRACT /* AMEE
5715 *****
5720 *
5725 RESET #LOCAL.#SAVE-CCC
5730 #LOCAL.#SAVE-PERSON
5735 *
5740 * Extracts records from the ME-ELIG-SEGMENT file
5745 *
5750 ASSIGN #MMIS-PERSON-ELIG.MMIS-RUN-NUM = #RUN-CONTROL.RESTART-RUN-NUM
5755 *
5760 READ-MES.
5765 READ ME-ELIG-SEGMENT BY MMIS-PERSON-ELIG FROM #MMIS-PERSON-ELIG
5770 *
5775 IF #MMIS-PERSON-ELIG.MMIS-RUN-NUM NE ME-ELIG-SEGMENT.MMIS-RUN-NUM
5780 ESCAPE BOTTOM (READ-MES.)
5785 END-IF
5790 *
5795 IF ME-ELIG-SEGMENT.DELETE-INDIC = #CONSTANT.#YES
5800 /* if the current MMIS run # is the same as the run # when added
5805 /* segment has never been sent to MMIS.
5810 IF ME-ELIG-SEGMENT.MMIS-RUN-NUM = ME-ELIG-SEGMENT.MMIS-ADD-RUN-NUM
5815 ESCAPE TOP /* READ-MES. /* ignore recs added then deleted
5820 END-IF
5825 END-IF
5830 *
5835 ASSIGN #CASE-CERT-AU-MBR.CASE-NUM = ME-ELIG-SEGMENT.CASE-NUM
5840 ASSIGN #CASE-CERT-AU-MBR.CERT-PERIOD-NUM
5845 = ME-ELIG-SEGMENT.CERT-PERIOD-NUM
5850 ASSIGN #CASE-CERT-AU-MBR.AU-MEMBER-NUM = ME-ELIG-SEGMENT.AU-MEMBER-NUM
5855 *
5860 AU-FIND.
5865 FIND (1) ME-AU-MEMBER WITH CASE-CERT-AU-MBR = #CASE-CERT-AU-MBR
5870 END-FIND /* (Au-find.)
5875 *
5880 * This logic was used to send person only if after conversion or if
5885 * there was a gap in elig prior to this change - removed 10/2/01.
5890 * RESET #LOGIC.#CREATE-PERSON-REC
5895 * IF ME-ELIG-SEGMENT.PERSON-NUM NE #LOCAL.#SAVE-PERSON
5900 * IF #LOGIC.#SEND-PERSON-AFTER-CONVERSION
5905 * MOVE TRUE TO #LOGIC.#CREATE-PERSON-REC
5910 * ELSE
5915 * send if gap in elig & we havent already sent in this run
5920 * ASSIGN #CASE-CERT-AU-MBR.CASE-NUM = ME-ELIG-SEGMENT.CASE-NUM
5925 * ASSIGN #CASE-CERT-AU-MBR.CERT-PERIOD-NUM
5930 * = ME-ELIG-SEGMENT.CERT-PERIOD-NUM
5935 * ASSIGN #CASE-CERT-AU-MBR.AU-MEMBER-NUM
5940 * = ME-ELIG-SEGMENT.AU-MEMBER-NUM
5945 * READ ME-AU-MEM-ELIG BY CASE-CERT-AU-MBR FROM #CASE-CERT-AU-MBR
5950 * IF ME-AU-MEM-ELIG.CASE-NUM NE #CASE-CERT-AU-MBR.CASE-NUM OR
5955 * ME-AU-MEM-ELIG.CERT-PERIOD-NUM NE
5960 * #CASE-CERT-AU-MBR.CERT-PERIOD-NUM OR
5965 * ME-AU-MEM-ELIG.AU-MEMBER-NUM NE
5970 * #CASE-CERT-AU-MBR.AU-MEMBER-NUM
5975 * ESCAPE BOTTOM
5980 * END-IF
5985 * IF *COUNTER = 1
5990 * MOVE ME-AU-MEM-ELIG.CLOSE-DATE TO #LOCAL.#CURR-CLOSE-DATE-N
5995 * ELSE
6000 * MOVE ME-AU-MEM-ELIG.START-DATE TO #LOCAL.#START-DATE-N
6005 * MOVE #LOCAL.#CURR-CLOSE-DATE-N TO #LOCAL.#PREV-CLOSE-DATE-N
6010 * MOVE ME-AU-MEM-ELIG.CLOSE-DATE TO #LOCAL.#CURR-CLOSE-DATE-N
6015 * END-IF
6020 * END-READ
6025 * IF #LOCAL.#CURR-CLOSE-DATE-N = 0
6030 * MOVE EDITED #LOCAL.#START-DATE-A TO
6035 * #LOCAL.#START-DATE-D (EM=YYYYMMDD)
6040 * MOVE EDITED #LOCAL.#PREV-CLOSE-DATE-A TO
6045 * #LOCAL.#PREV-CLOSE-DATE-D (EM=YYYYMMDD)
6050 * SUBTRACT #LOCAL.#PREV-CLOSE-DATE-D FROM #LOCAL.#START-DATE-D
6055 * GIVING #LOCAL.#DAYS
6060 * IF #LOCAL.#DAYS > 1

```

```

6065 *           MOVE TRUE TO #LOGIC.#CREATE-PERSON-REC
6070 *           END-IF
6075 *           END-IF
6080 *           END-IF
6085 * END-IF
6090 *
6095 * Now create the Eligibility Records
6100 * Get approval code & type case from CPTC, if diff from last ELIG-SEG
6105 *
6110 IF ME-ELIG-SEGMENT.CERT-PERIOD-NUM NE #LOCAL.#SAVE-CASE-NUM OR
6115 ME-ELIG-SEGMENT.CPTC-NUM           NE #LOCAL.#SAVE-CPTC-NUM
6120 ASSIGN #LOCAL.#SAVE-CASE-NUM = ME-ELIG-SEGMENT.CASE-NUM
6125 ASSIGN #LOCAL.#SAVE-CERT-NUM = ME-ELIG-SEGMENT.CERT-PERIOD-NUM
6130 ASSIGN #LOCAL.#SAVE-CPTC-NUM = ME-ELIG-SEGMENT.CPTC-NUM
6135 ASSIGN #CASE-CERT-CPTC-NUM = #LOCAL.#SAVE-CCC
6140 *
6145 CPTC-FIND.
6150 FIND (1) ME-CPTC WITH CASE-CERT-CPTC-NUM = #CASE-CERT-CPTC-NUM
6155 MOVE ME-CPTC.APPV-CODE TO #EXTRACT-REC.APPV-CODE
6160 MOVE ME-CPTC.CAT-TYPE-CASE TO #EXTRACT-REC.CAT-TYPE-CASE
6165 #LOCAL.#CAT-TYPE-CASE
6170 END-FIND /* (CPTC-FIND.)
6175 END-IF
6180 *
6185 * Move ME-ELIG-SEGMENT data to record
6190 *
6195 ASSIGN #EXTRACT-REC.#RECORD-TYPE      = #MEIMMIL2.#DETAIL
6200 ASSIGN #EXTRACT-REC.#SUB-RECORD-TYPE  = #MEIMMIL2.#AU-MEMBER-ELG
6205 ASSIGN #EXTRACT-REC.#RECORD-SEQUENCE = 0
6210 ASSIGN #EXTRACT-REC.#PERSON-NUMBER   = ME-ELIG-SEGMENT.PERSON-NUM
6215 ASSIGN #EXTRACT-REC.#PERSON-SEQUENCE =
6220 ME-ELIG-SEGMENT.ELIG-SEQUENCE-NUM
6225 *
6230 MOVE BY NAME ME-ELIG-SEGMENT TO #EXTRACT-REC.#AU-MEMBER-ELIG-EXTRACT
6235 IF ME-ELIG-SEGMENT.CLOSURE-CODE = #METCLRL.#QMB-FULL-REMOVAL
6240 MOVE ME-ELIG-SEGMENT.START-DATE TO #LOCAL.#WORK-DATE
6245 MOVE 01 TO #LOCAL.#WORK-DATE-DD
6250 MOVE #LOCAL.#WORK-DATE TO #EXTRACT-REC.#AME-START-DATE
6255 MOVE #METCLRL.#QMB-FULL-CLOSE-DATE TO #EXTRACT-REC.#AME-CLOSE-DATE
6260 ELSE
6265 MOVE ME-ELIG-SEGMENT.START-DATE TO #EXTRACT-REC.#AME-START-DATE
6270 MOVE ME-ELIG-SEGMENT.CLOSE-DATE TO #EXTRACT-REC.#AME-CLOSE-DATE
6275 END-IF
6280 MOVE ME-CPTC.APPV-CODE TO #EXTRACT-REC.APPV-CODE
6285 MOVE ME-CPTC.CAT-TYPE-CASE TO #EXTRACT-REC.CAT-TYPE-CASE
6290 *
6295 * Closure code 65 is for segments created when AU Member start date is
6300 * moved back or end date moved forward. MMIS wants these to look like
6305 * open/closes which have approval code 67 & closure code 66 (SIR 544)
6310 IF ME-ELIG-SEGMENT.CLOSURE-CODE = #CONSTANT.#MMIS-CLOSE-CODE /* 65
6315 MOVE #CONSTANT.#OPEN-CLOSE-APPV-CODE TO #EXTRACT-REC.APPV-CODE /* 67
6320 MOVE #CONSTANT.#SYS-CLOSURE-CODE TO #EXTRACT-REC.CLOSURE-CODE /* 66
6325 END-IF
6330 *
6335 ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
6340 *
6345 IF #LOCAL.#TYPE-CASE = #CONSTANT.#ALTERNATE-TYPE-CASE(*)
6350 ADD 1 TO #RUN-CONTROL.#ALT-AU-MEMBER-ELIG-COUNT
6355 PERFORM WRITE-ALT-WORK-FILE
6360 ELSE
6365 ADD 1 TO #RUN-CONTROL.#AU-MEMBER-ELIG-COUNT
6370 PERFORM WRITE-WORK-FILE
6375 END-IF
6380 RESET #EXTRACT-REC
6385 *
6390 *
6395 * Write a Person Record if we haven't already.
6400 *
6405 * IF #LOGIC.#CREATE-PERSON-REC
6410 * ASSIGN #LOCAL.#SAVE-PERSON = ME-ELIG-SEGMENT.PERSON-NUM
6415 *

```

```

6420 PERSON-FIND.
6425 FIND (1) ME-PERSON WITH PERSON-NUM = ME-ELIG-SEGMENT.PERSON-NUM
6430 *
6435 * Check to see if we already created a PERSON-REC in this run
6440 IF ME-PERSON.MMIS-RUN-NUM NE #RUN-CONTROL.RESTART-RUN-NUM
6445 PERFORM PERSON-DETAIL
6450 ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
6455 IF #LOCAL.#TYPE-CASE = #CONSTANT.#ALTERNATE-TYPE-CASE(*)
6460 ADD 1 TO #RUN-CONTROL.#ALT-EXTRA-PERSON-COUNT
6465 PERFORM WRITE-ALT-WORK-FILE
6470 ELSE
6475 ADD 1 TO #RUN-CONTROL.#EXTRA-PERSON-COUNT
6480 PERFORM WRITE-WORK-FILE
6485 END-IF
6490 RESET #EXTRACT-REC
6495 *
6500 END-IF
6505 END-FIND /* (PERSON-FIND.)
6510 * END-IF
6515 *
6520 RESET #LOCAL.#CAT-TYPE-CASE
6525 *
6530 END-READ /* (LOOP1.)
6535 *
6540 END-SUBROUTINE /* AU-MEMBER-ELIGIBILITY-EXTRACT
6545 *
6550 *****
6555 DEFINE SUBROUTINE SEGMENT-EXTRACT /* SE
6560 *****
6565 *
6570 * Extracts records from the ME-SEGMENT file
6575 *
6580 ASSIGN #MMIS-CASE-CERT-CPTC-SEG.MMIS-RUN-NUM
6585 = #RUN-CONTROL.RESTART-RUN-NUM
6590 *
6595 ME-SEG.
6600 READ ME-SEGMENT BY MMIS-CASE-CERT-CPTC-SEG FROM #MMIS-CASE-CERT-CPTC-SEG
6605 *
6610 IF ME-SEGMENT.MMIS-RUN-NUM NE #MMIS-CASE-CERT-CPTC-SEG.MMIS-RUN-NUM
6615 ESCAPE BOTTOM(ME-SEG.)
6620 END-IF
6625 *
6630 ASSIGN #CASE-CERT-CPTC-AU.CASE-NUM = ME-SEGMENT.CASE-NUM
6635 ASSIGN #CASE-CERT-CPTC-AU.CERT-PERIOD-NUM = ME-SEGMENT.CERT-PERIOD-NUM
6640 ASSIGN #CASE-CERT-CPTC-AU.CPTC-NUM = ME-SEGMENT.CPTC-NUM
6645 *
6650 ME-ELIG.
6655 READ ME-ELIG-SEGMENT BY CASE-CERT-CPTC-AU FROM #CASE-CERT-CPTC-AU
6660 *
6665 IF ME-ELIG-SEGMENT.CASE-NUM NE #CASE-CERT-CPTC-AU.CASE-NUM OR
6670 ME-ELIG-SEGMENT.CERT-PERIOD-NUM NE
6675 #CASE-CERT-CPTC-AU.CERT-PERIOD-NUM OR
6680 ME-ELIG-SEGMENT.CPTC-NUM NE #CASE-CERT-CPTC-AU.CPTC-NUM
6685 ESCAPE BOTTOM(ME-ELIG.)
6690 END-IF
6695 *
6700 MOVE ME-ELIG-SEGMENT.ELIG-SEQUENCE-NUM
6705 TO #EXTRACT-REC.#ELIG-SEQUENCE-NUM
6710 ESCAPE BOTTOM /* (ME-ELIG.) We only need the first
6715 *
6720 END-READ /* (ME-ELIG.)
6725 *
6730 ASSIGN #EXTRACT-REC.#RECORD-TYPE = #MEIMMIL2.#DETAIL
6735 ASSIGN #EXTRACT-REC.#SUB-RECORD-TYPE = #MEIMMIL2.#AU-MEMBER-SEGMENT
6740 ASSIGN #EXTRACT-REC.#RECORD-SEQUENCE = 0
6745 ASSIGN #EXTRACT-REC.#PERSON-NUMBER = ME-ELIG-SEGMENT.PERSON-NUM
6750 ASSIGN #EXTRACT-REC.#PERSON-SEQUENCE = ME-ELIG-SEGMENT.START-DATE
6755 *
6760 MOVE BY NAME ME-SEGMENT TO #EXTRACT-REC.#AU-MEMBER-SEG-EXTRACT
6765 MOVE ME-SEGMENT.PROV-NUM TO #EXTRACT-REC.#PROV-NUM
6770 ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER

```

```

6775 *
6780 ASSIGN #CASE-CERT-CPTC-NUM.CASE-NUM = ME-SEGMENT.CASE-NUM
6785 ASSIGN #CASE-CERT-CPTC-NUM.CERT-PERIOD-NUM =
6790 ME-SEGMENT.CERT-PERIOD-NUM
6795 ASSIGN #CASE-CERT-CPTC-NUM.CPTC-NUM = 1
6800 *
6805 TCAS-FIND3.
6810 FIND (1) ME-CPTC WITH CASE-CERT-CPTC-NUM = #CASE-CERT-CPTC-NUM
6815 ASSIGN #LOCAL.#CAT-TYPE-CASE = ME-CPTC.CAT-TYPE-CASE
6820 END-FIND /* (TCAS-FIND3.)
6825 *
6830 IF #LOCAL.#TYPE-CASE = #CONSTANT.#ALTERNATE-TYPE-CASE(*)
6835 ADD 1 TO #RUN-CONTROL.#ALT-AU-MEMBER-SEG-COUNT
6840 PERFORM WRITE-ALT-WORK-FILE
6845 ELSE
6850 ADD 1 TO #RUN-CONTROL.#AU-MEMBER-SEG-COUNT
6855 PERFORM WRITE-WORK-FILE
6860 END-IF
6865 RESET #EXTRACT-REC
6870 /*
6875 /* Write Person Record if havent already
6880 /*
6885 FIND-PERS-SGMT.
6890 FIND (1) ME-PERSON WITH PERSON-NUM = ME-ELIG-SEGMENT.PERSON-NUM
6895 /*
6900 IF ME-PERSON.MMIS-RUN-NUM NE #RUN-CONTROL.RESTART-RUN-NUM
6905 PERFORM PERSON-DETAIL
6910 ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
6915 IF #LOCAL.#TYPE-CASE = #CONSTANT.#ALTERNATE-TYPE-CASE(*)
6920 ADD 1 TO #RUN-CONTROL.#ALT-EXTRA-PERS-SGMT-COUNT
6925 PERFORM WRITE-ALT-WORK-FILE
6930 ELSE
6935 ADD 1 TO #RUN-CONTROL.#EXTRA-PERS-SGMT-COUNT
6940 PERFORM WRITE-WORK-FILE
6945 END-IF
6950 RESET #EXTRACT-REC
6955 END-IF
6960 /*
6965 END-FIND /* (FIND-PERS-SGMT.)
6970 /*
6975 RESET #LOCAL.#CAT-TYPE-CASE
6980 *
6985 END-READ /*(ME-SEG.)
6990 *
6995 * Extracts records from the ME-HOSPICE-SGMT file
7000 *
7005 ASSIGN #MMIS-CASE-CERT-CPTC-AU-SEG.MMIS-RUN-NUM
7010 = #RUN-CONTROL.RESTART-RUN-NUM
7015 *
7020 ME-HSPC.
7025 READ ME-HOSPICE-SGMT BY MMIS-CASE-CERT-CPTC-AU-SEG
7030 FROM #MMIS-CASE-CERT-CPTC-AU-SEG
7035 *
7040 IF ME-HOSPICE-SGMT.MMIS-RUN-NUM NE
7045 #MMIS-CASE-CERT-CPTC-AU-SEG.MMIS-RUN-NUM
7050 ESCAPE BOTTOM(ME-HSPC.) /* End of current MMIS extract.
7055 END-IF
7060 *
7065 ASSIGN #CASE-CERT-CPTC-AU.CASE-NUM = ME-HOSPICE-SGMT.CASE-NUM
7070 ASSIGN #CASE-CERT-CPTC-AU.CERT-PERIOD-NUM
7075 = ME-HOSPICE-SGMT.CERT-PERIOD-NUM
7080 ASSIGN #CASE-CERT-CPTC-AU.CPTC-NUM = ME-HOSPICE-SGMT.CPTC-NUM
7085 ASSIGN #CASE-CERT-CPTC-AU.AU-MEMBER-NUM
7090 = ME-HOSPICE-SGMT.AU-MEMBER-NUM
7095 *
7100 ME-ELIG2.
7105 READ ME-ELIG-SEGMENT BY CASE-CERT-CPTC-AU FROM #CASE-CERT-CPTC-AU
7110 *
7115 IF ME-ELIG-SEGMENT.CASE-NUM NE #CASE-CERT-CPTC-AU.CASE-NUM OR
7120 ME-ELIG-SEGMENT.CERT-PERIOD-NUM NE
7125 #CASE-CERT-CPTC-AU.CERT-PERIOD-NUM OR

```



```

7130     ME-ELIG-SEGMENT.CPTC-NUM NE #CASE-CERT-CPTC-AU.CPTC-NUM OR
7135     ME-ELIG-SEGMENT.AU-MEMBER-NUM NE #CASE-CERT-CPTC-AU.AU-MEMBER-NUM
7140     ESCAPE BOTTOM(ME-ELIG2.)
7145     END-IF
7150 *
7155     MOVE ME-ELIG-SEGMENT.ELIG-SEQUENCE-NUM
7160     TO #EXTRACT-REC.#ELIG-SEQUENCE-NUM
7165     ESCAPE BOTTOM /* (ME-ELIG2.) We only need the first
7170 *
7175     END-READ /* (ME-ELIG2.)
7180 *
7185     ASSIGN #EXTRACT-REC.#RECORD-TYPE      = #MEIMMIL2.#DETAIL
7190     ASSIGN #EXTRACT-REC.#SUB-RECORD-TYPE = #MEIMMIL2.#AU-MEMBER-SEGMENT
7195     ASSIGN #EXTRACT-REC.#RECORD-SEQUENCE = 0
7200     ASSIGN #EXTRACT-REC.#PERSON-NUMBER   = ME-ELIG-SEGMENT.PERSON-NUM
7205     ASSIGN #EXTRACT-REC.#PERSON-SEQUENCE = ME-ELIG-SEGMENT.START-DATE
7210 *
7215     MOVE BY NAME ME-HOSPICE-SGMT TO #EXTRACT-REC.#AU-MEMBER-SEG-EXTRACT
7220     MOVE ME-HOSPICE-SGMT.HSPC-START-DATE TO #EXTRACT-REC.SEG-START-DATE
7225     MOVE ME-HOSPICE-SGMT.HSPC-CLOSE-DATE TO #EXTRACT-REC.SEG-CLOSE-DATE
7230     MOVE ME-HOSPICE-SGMT.PROV-NUM TO #EXTRACT-REC.#PROV-NUM
7235     RESET #EXTRACT-REC.PLI-AMT
7240     ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
7245 *
7250 * Assign the Superdescriptor for TCAS-FIND5
7255     ASSIGN #CASE-CERT-CPTC-NUM.CASE-NUM = ME-HOSPICE-SGMT.CASE-NUM
7260     ASSIGN #CASE-CERT-CPTC-NUM.CERT-PERIOD-NUM
7265     = ME-HOSPICE-SGMT.CERT-PERIOD-NUM
7270     ASSIGN #CASE-CERT-CPTC-NUM.CPTC-NUM = 1
7275 *
7280     TCAS-FIND5.
7285     FIND (1) ME-CPTC WITH CASE-CERT-CPTC-NUM = #CASE-CERT-CPTC-NUM
7290     ASSIGN #LOCAL.#CAT-TYPE-CASE = ME-CPTC.CAT-TYPE-CASE
7295     END-FIND /* (TCAS-FIND5.)
7300 *
7305     IF #LOCAL.#TYPE-CASE = #CONSTANT.#ALTERNATE-TYPE-CASE(*)
7310     ADD 1 TO #RUN-CONTROL.#ALT-AU-MEMBER-SEG-COUNT
7315     PERFORM WRITE-ALT-WORK-FILE
7320     ELSE
7325     ADD 1 TO #RUN-CONTROL.#AU-MEMBER-SEG-COUNT
7330     PERFORM WRITE-WORK-FILE
7335     END-IF
7340     RESET #EXTRACT-REC
7345     /*
7350     /* Write Person Record if havent already
7355     /*
7360     FIND-PERS-HSPC.
7365     FIND (1) ME-PERSON WITH PERSON-NUM = ME-ELIG-SEGMENT.PERSON-NUM
7370     /*
7375     IF ME-PERSON.MMIS-RUN-NUM NE #RUN-CONTROL.RESTART-RUN-NUM
7380     PERFORM PERSON-DETAIL
7385     ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
7390     IF #LOCAL.#TYPE-CASE = #CONSTANT.#ALTERNATE-TYPE-CASE(*)
7395     ADD 1 TO #RUN-CONTROL.#ALT-EXTRA-PERS-SGMT-COUNT
7400     PERFORM WRITE-ALT-WORK-FILE
7405     ELSE
7410     ADD 1 TO #RUN-CONTROL.#EXTRA-PERS-SGMT-COUNT
7415     PERFORM WRITE-WORK-FILE
7420     END-IF
7425     RESET #EXTRACT-REC
7430     END-IF
7435     /*
7440     END-FIND /* (FIND-PERS-HSPC.)
7445     /*
7450     RESET #LOCAL.#CAT-TYPE-CASE
7455 *
7460     END-READ /* (ME-HSPC.)
7465 *
7470     END-SUBROUTINE /* SEGMENT-EXTRACT
7475 *
7580     *****

```

```

7585 DEFINE SUBROUTINE WRITE-WORK-FILE /* WWF
7590 *****
7595 *
7600 DECIDE ON FIRST #EXTRACT-REC.#RECORD-TYPE
7605 VALUE #MEIMMIL2.#HEADER
7610 WRITE WORK FILE 1 VARIABLE #EXTRACT-REC.#RECORD-INFO
7615 #EXTRACT-REC.#HEADER-INFO
7620 #EXTRACT-REC.#END-MARKER
7625 VALUE #MEIMMIL2.#DETAIL
7630 DECIDE ON FIRST #EXTRACT-REC.#SUB-RECORD-TYPE
7635 VALUE #MEIMMIL2.#PERSON-RECORD
7640 WRITE WORK FILE 1 VARIABLE #EXTRACT-REC.#RECORD-INFO
7645 #EXTRACT-REC.#PERSON-EXTRACT
7650 #EXTRACT-REC.#END-MARKER
7655 VALUE #MEIMMIL2.#SWIPE-CARD
7660 WRITE WORK FILE 1 VARIABLE #EXTRACT-REC.#RECORD-INFO
7665 #EXTRACT-REC.#SWIPE-CARD-EXTRACT
7670 #EXTRACT-REC.#END-MARKER
7675 VALUE #MEIMMIL2.#LOCK-IN-COMTY-CARE
7680 WRITE WORK FILE 1 VARIABLE #EXTRACT-REC.#RECORD-INFO
7685 #EXTRACT-REC.#LOCK-IN-EXTRACT
7690 #EXTRACT-REC.#END-MARKER
7695 VALUE #MEIMMIL2.#AU-MEMBER
7700 WRITE WORK FILE 1 VARIABLE #EXTRACT-REC.#RECORD-INFO
7705 #EXTRACT-REC.#AU-MEMBER-EXTRACT
7710 #EXTRACT-REC.#END-MARKER
7715 VALUE #MEIMMIL2.#AU-MEMBER-PARISH
7720 WRITE WORK FILE 1 VARIABLE #EXTRACT-REC.#RECORD-INFO
7725 #EXTRACT-REC.#AU-MEMBER-PARISH-EXTRACT
7730 #EXTRACT-REC.#END-MARKER
7735 VALUE #MEIMMIL2.#AU-MEMBER-ELG
7740 WRITE WORK FILE 1 VARIABLE #EXTRACT-REC.#RECORD-INFO
7745 #EXTRACT-REC.#AU-MEMBER-ELIG-EXTRACT
7750 #EXTRACT-REC.#END-MARKER
7755 VALUE #MEIMMIL2.#AU-MEMBER-SEGMENT
7760 WRITE WORK FILE 1 VARIABLE #EXTRACT-REC.#RECORD-INFO
7765 #EXTRACT-REC.#AU-MEMBER-SEG-EXTRACT
7770 #EXTRACT-REC.#END-MARKER
7775 NONE
7780 IGNORE
7785 END-DECIDE
7790 VALUE #MEIMMIL2.#TRAILER
7795 WRITE WORK FILE 1 VARIABLE #EXTRACT-REC.#RECORD-INFO
7800 #EXTRACT-REC.#END-MARKER
7805 NONE
7810 IGNORE
7815 END-DECIDE
7820 ADD 1 TO #RUN-CONTROL.#EXTRACT-COUNT
7825 *
7830 END-SUBROUTINE /* WRITE-WORK-FILE
7835 *
7840 *****
7845 DEFINE SUBROUTINE WRITE-ALT-WORK-FILE /* WAWF
7850 *****
7855 *
7860 * alternate file for SLMB, Q1, Q2, QDWI
7865 DECIDE ON FIRST #EXTRACT-REC.#RECORD-TYPE
7870 VALUE #MEIMMIL2.#HEADER
7875 WRITE WORK FILE 2 VARIABLE #EXTRACT-REC.#RECORD-INFO
7880 #EXTRACT-REC.#HEADER-INFO
7885 #EXTRACT-REC.#END-MARKER
7890 VALUE #MEIMMIL2.#DETAIL
7895 DECIDE ON FIRST #EXTRACT-REC.#SUB-RECORD-TYPE
7900 VALUE #MEIMMIL2.#PERSON-RECORD
7905 WRITE WORK FILE 2 VARIABLE #EXTRACT-REC.#RECORD-INFO
7910 #EXTRACT-REC.#PERSON-EXTRACT
7915 #EXTRACT-REC.#END-MARKER
7920 VALUE #MEIMMIL2.#SWIPE-CARD /* currently s/b none of these
7925 WRITE WORK FILE 2 VARIABLE #EXTRACT-REC.#RECORD-INFO
7930 #EXTRACT-REC.#SWIPE-CARD-EXTRACT
7935 #EXTRACT-REC.#END-MARKER

```

```

7940     VALUE #MEIMMIL2.#LOCK-IN-COMTY-CARE
7945     WRITE WORK FILE 2 VARIABLE #EXTRACT-REC.#RECORD-INFO
7950     #EXTRACT-REC.#LOCK-IN-EXTRACT
7955     #EXTRACT-REC.#END-MARKER
7960     VALUE #MEIMMIL2.#AU-MEMBER
7965     WRITE WORK FILE 2 VARIABLE #EXTRACT-REC.#RECORD-INFO
7970     #EXTRACT-REC.#AU-MEMBER-EXTRACT
7975     #EXTRACT-REC.#END-MARKER
7980     VALUE #MEIMMIL2.#AU-MEMBER-PARISH
7985     WRITE WORK FILE 2 VARIABLE #EXTRACT-REC.#RECORD-INFO
7990     #EXTRACT-REC.#AU-MEMBER-PARISH-EXTRACT
7995     #EXTRACT-REC.#END-MARKER
8000     VALUE #MEIMMIL2.#AU-MEMBER-ELG
8005     WRITE WORK FILE 2 VARIABLE #EXTRACT-REC.#RECORD-INFO
8010     #EXTRACT-REC.#AU-MEMBER-ELIG-EXTRACT
8015     #EXTRACT-REC.#END-MARKER
8020     VALUE #MEIMMIL2.#AU-MEMBER-SEGMENT
8025     WRITE WORK FILE 2 VARIABLE #EXTRACT-REC.#RECORD-INFO
8030     #EXTRACT-REC.#AU-MEMBER-SEG-EXTRACT
8035     #EXTRACT-REC.#END-MARKER
8040     NONE
8045     IGNORE
8050     END-DECIDE
8055     VALUE #MEIMMIL2.#TRAILER
8060     WRITE WORK FILE 2 VARIABLE #EXTRACT-REC.#RECORD-INFO
8065     #EXTRACT-REC.#END-MARKER
8070     NONE
8075     IGNORE
8080     END-DECIDE
8085     ADD 1 TO #RUN-CONTROL.#ALT-EXTRACT-COUNT
8090     *
8095     END-SUBROUTINE /* WRITE-ALT-WORK-FILE
8100     *
8105     *****
8110     DEFINE SUBROUTINE PERSON-DETAIL /* PD
8115     *****
8120     *
8125     ASSIGN #EXTRACT-REC.#RECORD-TYPE      = #MEIMMIL2.#DETAIL
8130     ASSIGN #EXTRACT-REC.#SUB-RECORD-TYPE  = #MEIMMIL2.#PERSON-RECORD
8135     ASSIGN #EXTRACT-REC.#RECORD-SEQUENCE = 0
8140     ASSIGN #EXTRACT-REC.#PERSON-NUMBER   = ME-PERSON.PERSON-NUM
8145     ASSIGN #EXTRACT-REC.#PERSON-SEQUENCE = 1
8150     *
8155     MOVE BY NAME ME-PERSON TO #EXTRACT-REC.#PERSON-EXTRACT
8160     *
8165     * Format Address Lines
8170     *
8175     MOVE ME-PERSON.ADDR(1) TO #MEIMMPN2-I/O.#ADDR-1
8180     MOVE ME-PERSON.ADDR(2) TO #MEIMMPN2-I/O.#ADDR-2
8185     MOVE ME-PERSON.ADDR(3) TO #MEIMMPN2-I/O.#ADDR-3
8190     *
8195     CALLNAT 'MEIMMPN2' /* Format Address Lines
8200     #MEIMMPN2-I/O
8205     #CXXGNLP-OUT
8210     *
8215     MOVE #MEIMMPN2-I/O.#ADDR-1 TO #EXTRACT-REC.#ADDR-1
8220     MOVE #MEIMMPN2-I/O.#ADDR-2 TO #EXTRACT-REC.#ADDR-2
8225     MOVE #MEIMMPN2-I/O.#ADDR-3 TO #EXTRACT-REC.#ADDR-3
8230     *
8235     * New Person Indicator
8240     *
8245     MOVE ME-PERSON.PERSON-NUM TO #MEIMMPN1-IN.PERSON-NUM
8250     MOVE #MMIS-RUN-PERSON-NUM.MMIS-RUN-NUM
8255     TO #MEIMMPN1-IN.#CURR-MMIS-RUN-NUM
8260     *
8265     CALLNAT 'MEIMMPN1' /* Determine New Person Indicator
8270     #MEIMMPN1-IN
8275     #MEIMMPN1-OUT
8280     #CXXGNLP-OUT
8285     *
8290     IF #MEIMMPN1-OUT.#NEW-PERSON-INDIC = 'Y'

```

```

8295  ASSIGN #EXTRACT-REC.#NEW-PERSON-INDIC = '1'
8300  ELSE
8305  ASSIGN #EXTRACT-REC.#NEW-PERSON-INDIC = '0'
8310  END-IF
8315  *
8320  *   Default Preferred Language Indicator
8325  *
8330  IF #EXTRACT-REC.PREF-LANG-CODE = 0
8335  ASSIGN #EXTRACT-REC.PREF-LANG-CODE = #METLNGL.#ENGLISH
8340  END-IF
8345  *
8350  ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
8355  *
8360  END-SUBROUTINE /* PERSON-DETAIL
8365  *
8555  *****
8560  DEFINE SUBROUTINE WRITE-PARISH-RECORD /* WPR
8565  *****
8570  *
8575  * This subroutine writes out a Parish record (5) for all newly added
8580  * AU Members
8585  *
8590  FIND-CASE.
8595  FIND (1) ME-CASE WITH CASE-NUM = ME-AU-MEMBER-LOG.CASE-NUM
8600  *
8605  ASSIGN #EXTRACT-REC.#RECORD-TYPE      = #MEIMMIL2.#DETAIL
8610  ASSIGN #EXTRACT-REC.#SUB-RECORD-TYPE  = #MEIMMIL2.#AU-MEMBER-PARISH
8615  ASSIGN #EXTRACT-REC.#RECORD-SEQUENCE  = 0
8620  ASSIGN #EXTRACT-REC.#PERSON-NUMBER    = ME-AU-MEMBER-LOG.PERSON-NUM
8625  *
8630  * Assign the superdescriptor for TCAS-FIND4
8635  ASSIGN #CASE-CERT-CPTC-NUM.CASE-NUM
8640  = ME-AU-MEMBER-LOG.CASE-NUM
8645  ASSIGN #CASE-CERT-CPTC-NUM.CERT-PERIOD-NUM
8650  = ME-AU-MEMBER-LOG.CERT-PERIOD-NUM
8655  ASSIGN #CASE-CERT-CPTC-NUM.CPTC-NUM = 1
8660  *
8665  TCAS-FIND4.
8670  FIND (1) ME-CPTC WITH CASE-CERT-CPTC-NUM
8675  = #CASE-CERT-CPTC-NUM
8680  ASSIGN #LOCAL.#CAT-TYPE-CASE = ME-CPTC.CAT-TYPE-CASE
8685  END-FIND /* (TCAS-FIND4.)
8690  *
8695  FOR #LOCAL.#I = 1 TO C*PARISH-RES-HISTORY
8700  MOVE ME-CASE.PARISH-START-DATE(#I) TO
8705  #EXTRACT-REC.PARISH-START-DATE
8710  #EXTRACT-REC.#PERSON-SEQUENCE
8715  MOVE ME-CASE.PARISH-LOC-ID(#I) TO
8720  #EXTRACT-REC.PARISH-LOC-ID
8725  ASSIGN #EXTRACT-REC.#END-MARKER = #CONSTANT.#END-MARKER
8730  IF #LOCAL.#TYPE-CASE = #CONSTANT.#ALTERNATE-TYPE-CASE(*)
8735  ADD 1 TO #RUN-CONTROL.#ALT-NEW-AU-MEM-PARISH-COUNT
8740  PERFORM WRITE-ALT-WORK-FILE
8745  ELSE
8750  ADD 1 TO #RUN-CONTROL.#NEW-AU-MEM-PARISH-COUNT
8755  PERFORM WRITE-WORK-FILE
8760  END-IF
8765  END-FOR
8770  /*
8775  RESET #EXTRACT-REC
8780  #LOCAL.#CAT-TYPE-CASE
8785  END-FIND /* (FIND-CASE.)
8790  *
8795  END-SUBROUTINE /* WRITE-PARISH-RECORD
9270  *
9275  END
□

```