

**Table of Contents**

**1. End of Day Info Processes/ Scripts and Activities .....1**

**1.1 Perform back-up .....4**

**1.2 Automatic termination of clients no longer eligible .....4**

    1.2.1 EA1\_TERM\_EOD.SQL .....4

    1.2.2 EA1\_TERM\_WL\_CLIENTS.SQL .....5

**1.3 Perform automatic category changes .....5**

    1.3.1 EA1\_CAT\_EOD.SQL .....5

    1.3.2 EA1\_CAT\_SYNC\_EOD.SQL .....5

**1.4 Mark appointments as Missed or Kept .....5**

    1.4.1 EA1\_APPT\_EOD.SQL .....5

    1.4.2 EA1\_NEW\_CLASS\_EOD.SQL .....6

    1.4.3 EA1\_NEW\_APPT\_EOD.SQL .....6

**1.5 Delete information no longer used by the system .....6**

    1.5.1 EA1\_OTHER\_PURG.SQL .....6

    1.5.2 EA1\_PURGE\_OLD\_APPT.SQL .....7

    1.5.3 EA1\_PURGE\_NEW\_APPT.SQL .....7

**1.6 Set exclusively breast feeding clients to active status .....7**

    1.6.1 EA1\_IEN\_ACTIVE.SQL .....7

**1.7 Void FIs not deposited after the stale date .....7**

    1.7.1 EC1\_STATE\_DATE\_FI.SQL .....7

**1.8 Process pending transfer requests .....7**

    1.8.1 EA1\_TRANSFER.SQL .....7

**1.9 Generate notice of ineligibility and appointment letters .....7**

    1.9.1 CS\_LETTER\_OF\_TERM.FMX .....7

**1.10 Gather updated client information, new issuance information, void information, and requests for transfer .....8**

    1.10.1 EA1\_TAB\_UP.SQL .....8

    1.10.2 EA1\_EXPORT\_TAB.TXT .....10

**1.11 Transfer datafile from agency to central .....10**

    1.11.1 EA1\_FTP\_RUN.BAT/ EA1\_FTP\_TO\_CENTRAL.BAT .....10

**1.12 Delete information no longer used by the system .....10**

    1.12.1 EC1\_OTHER\_PURG.SQL .....10

**1.13 Compile direct payment and revalidated FI information and send to bank .....11**

    1.13.1 EC1\_CTRL\_CHK.SQL .....11

**1.14 Compile vendor information to send to bank .....11**

    1.14.1 EC1\_VENDOR\_FSMC.BAT .....11

    1.14.2 EC1\_VENDOR\_FSMC.SQL .....11

    1.14.3 EC1\_PGA\_FSMC.SQL .....11

    1.14.4 EC1\_OUTBOUND\_VENDOR\_FSMC.SQL .....11

    1.14.5 EC1\_OUTBOUND\_PGA\_FSMC.SQL .....12

**1.15 Consolidate client information .....12**

    1.15.1 EC2\_SP\_TRNC.SQL .....12

    1.15.2 EC2\_TRNC.SQL .....12

    1.15.3 EC2\_DELTRIG.SQL .....14

    1.15.4 EC2\_DELTRIG\_AV.SQL .....14

    1.15.5 EC2\_IN\_TAB.SQL .....14

    1.15.6 EC2\_TAB\_UP.SQL .....16

**1.16 Financial - process totals and store in F\_CASELOAD table. Totals are based on priority, language, caseload type, ethnic group and poverty level .....18**

1.16.1 EC2\_CL\_REC\_PROCESS.SQL ..... 18

1.16.2 EC2\_DEL\_CL\_FUT\_RECS.SQL ..... 18

**1.17 Update caseload assignment information from the clinics..... 18**

1.17.1 EC2\_SYNC\_CLINIC\_ASSIGNMENT.SQL ..... 18

1.17.2 EC2\_populate\_clinic.SQL ..... 18

**1.18 Consolidate and update all food instrument data to calculate FI obligation value..... 18**

1.18.1 EC2\_FOOD\_UP.SQL ..... 18

1.18.2 EC2\_FI\_UPD.SQL ..... 18

1.18.3 EC2\_FORMULAS.SQL ..... 19

**1.19 Search for potential dual enrollment clients..... 19**

1.19.1 EC2\_DUAL\_ENROLL.SQL ..... 19

**1.20 Compile new issuance and void information to send to bank..... 19**

1.20.1 EC2\_ISSUE\_FSMC.SQL ..... 19

**1.21 Poll bank, send and receive FI and Vendor information ..... 19**

1.21.1 EC4.BAT ..... 19

**1.22 Create consolidated record of all FI issuance and redemption information and create vendor payment records..... 19**

1.22.1 EC4\_BNK\_POST.SQL ..... 19

**1.23 Create date of current food funds available from all sources as well as current food obligations and expenditures to establish current cash position ..... 20**

1.23.1 EC4\_CASHFLOW\_UPD.SQL ..... 20

**1.24 Financial (end of month) - Populate F\_INCOME to POVERTY table. These values are used in populating of caseload table ..... 20**

1.24.1 EC4\_MON\_END.SQL ..... 20

**1.25 Vendor (end of month) calculate and update peer group averages..... 20**

1.25.1 EC4\_VEN\_PEER\_FI.SQL ..... 20

1.25.2 EC4\_VEN\_LOC\_AGCY\_FI ..... 20

**1.26 Vendor run and print analysis factor reports ..... 21**

1.26.1 EC4\_VEN\_PRICE.SQL ..... 21

**1.27 Generate dual enrollment report to be sent to agencies ..... 21**

1.27.1 EC4\_DUAL\_RPT.BAT ..... 21

1.27.2 EC4\_DUAL.REP ..... 21

**1.28 Gather new/updated data to be sent to the agencies..... 21**

1.28.1 EC4\_TRNC.SQL ..... 21

1.28.2 EC4\_TAB\_%.SQL ..... 21

1.28.3 EC4\_AGCY\_BEING\_PROCESSED.SQL ..... 22

1.28.4 EC4\_EXPORT\_TAB.TXT ..... 22

**1.29 Process transfer requests..... 22**

1.29.1 EC2\_STATE\_CTLI.SQL ..... 22

1.29.2 EC4\_STATE\_A\_CTLI.SQL ..... 22

**1.30 Prepare archival retrieval data..... 22**

1.30.1 EC1\_PRG\_ARCHV.SQL ..... 22

1.30.2 EC4\_RETRIEVE.SQL ..... 22

1.30.3 EC4\_TRUNC\_ARCH\_CLI.SQL ..... 23

1.30.4 EC4\_INS\_RETR\_CLI.SQL ..... 23

1.30.5 EC4\_INS\_RETR\_VEN.SQL ..... 23

**1.31 Print out status logs ..... 23**

1.31.1 EC\_PRINT.BAT ..... 23

**1.32 EOD Batch Log ..... 23**

1.32.1 EOD\_LOG.BAT ..... 23

**1.33 Transfer datafile from ftp server to agency server ..... 23**

1.33.1 EA3\_FTP\_RUN.BAT/ EA3\_FTP\_FROM\_CENTRAL.BAT ..... 23

**1.34 Consolidate information from central..... 24**

1.34.1 EA3\_TRNC.SQL ..... 24

1.34.2 EA3\_PRE.SQL..... 28

1.34.3 EA3\_DELTRIG.SQL ..... 28

1.34.4 EA3\_IN\_TAB.SQL ..... 29

1.34.5 EA3\_TAB\_UP.SQL ..... 32

**1.35 Clinic serial number replenish..... 34**

1.35.1 EC4\_CL\_SER\_REP.SQL..... 34

1.35.2 CR\_DUAL\_ENROLLMENT.REP ..... 34

**1.36 Statewide summary of participants for dual enrollment check..... 34**

1.36.1 EA3\_PART\_DUAL\_CHK.SQL ..... 34

1.36.2 CR\_DUAL\_ENROLLMENT.REP ..... 34

**1.37 Update caseload assignment information from the clinics..... 35**

1.37.1 EA3\_SYNC\_LA\_ASSIGNMENT.SQL ..... 35

**1.38 Update records at losing agencies for transferred clients and delete any irrelevant transfer information..... 35**

1.38.1 EA3\_DEL\_IRR\_TRANSFER\_INFO.SQL ..... 35

1.38.2 EA3\_TERM\_TR\_CLIENTS.SQL ..... 35

**1.39 Send redemption/rejection information to agencies ..... 35**

1.39.1 EA3\_PROC\_FI.SQL ..... 35

**1.40 Transmit archival clients and vendors and update records at agencies ..... 36**

1.40.1 EA3\_PRG\_ARCHV.SQL ..... 36

1.40.2 EA3\_RETRIEVE.SQL..... 36

1.40.3 EA3\_INS\_RETR\_CLI.SQL ..... 36

1.40.4 EA3\_INS\_RETR\_VEN.SQL ..... 36

**1.41 Vendor monthly reports ..... 36**

**1.42 Vendor quarterly reports ..... 36**

**1.43 Print out status logs of the process ..... 37**

1.43.1 EA\_PRINT.BAT ..... 37

**1.44 Update State clients for failed EOD agencies/ Sync c\_state\_clients ..... 37**

1.44.1 EC2\_NOEOD.SQL ..... 37

1.44.2 EC2\_UPD\_E\_C\_CLIENTS.SQL..... 37

1.44.3 EC4\_CSC\_UPDATE.SQL ..... 37

1.44.4 EC4\_SYNC\_STATE\_CLIENTS.SQL..... 37

**1.45 Identify Duplicate FI Range..... 37**

1.45.1 EC2\_DUPLICATE\_FI\_RANGE.SQL ..... 37

**1.46 Bring up missing issuance FI's from agencies ..... 37**

1.46.1 EC2\_FAILED\_FIS.SQL ..... 37

1.46.2 EC4\_GET\_UPDATES.SQL..... 38

**2 Appendix A..... 39**

**3 Appendix B - FSMC Banking Interface..... 80**

**3.1 FSMC Banking Interface to the AIM System ..... 80**

3.1.1 Overview ..... 80

3.1.2 Issuance Information ..... 80

3.1.3 Peer Group Average Information ..... 81

3.1.4 Vendor Information ..... 82

**4 Appendix C – Start Of Month Caseload Process ..... 84**

<b>4.1</b>	<b>Monthly population of participation information for clients.....</b>	<b>85</b>
4.1.1	Overview .....	85
4.1.2	Participation Information.....	85
<b>4.2</b>	<b>Monthly population of participation information for farmer market .....</b>	<b>88</b>
4.2.1	Overview .....	88
4.2.2	Participation Information.....	89
<b>4.3</b>	<b>Monthly population of participation information for CSFP clients.....</b>	<b>90</b>
4.3.1	Overview .....	90
4.3.2	Process .....	91

## 1. End of Day Info Processes/ Scripts and Activities

### *Overview*

This Window permits the user to initiate the end of day processing procedures required at different host computer sites. This processing synchronizes all the data at the Local Agency Servers to the data at the Central Operations Server at MicroAge. This screen allows the user to initiate the process manually, though each Local Agency's AIM database will have a pre-programmed time that the End of Day process initiates automatically.

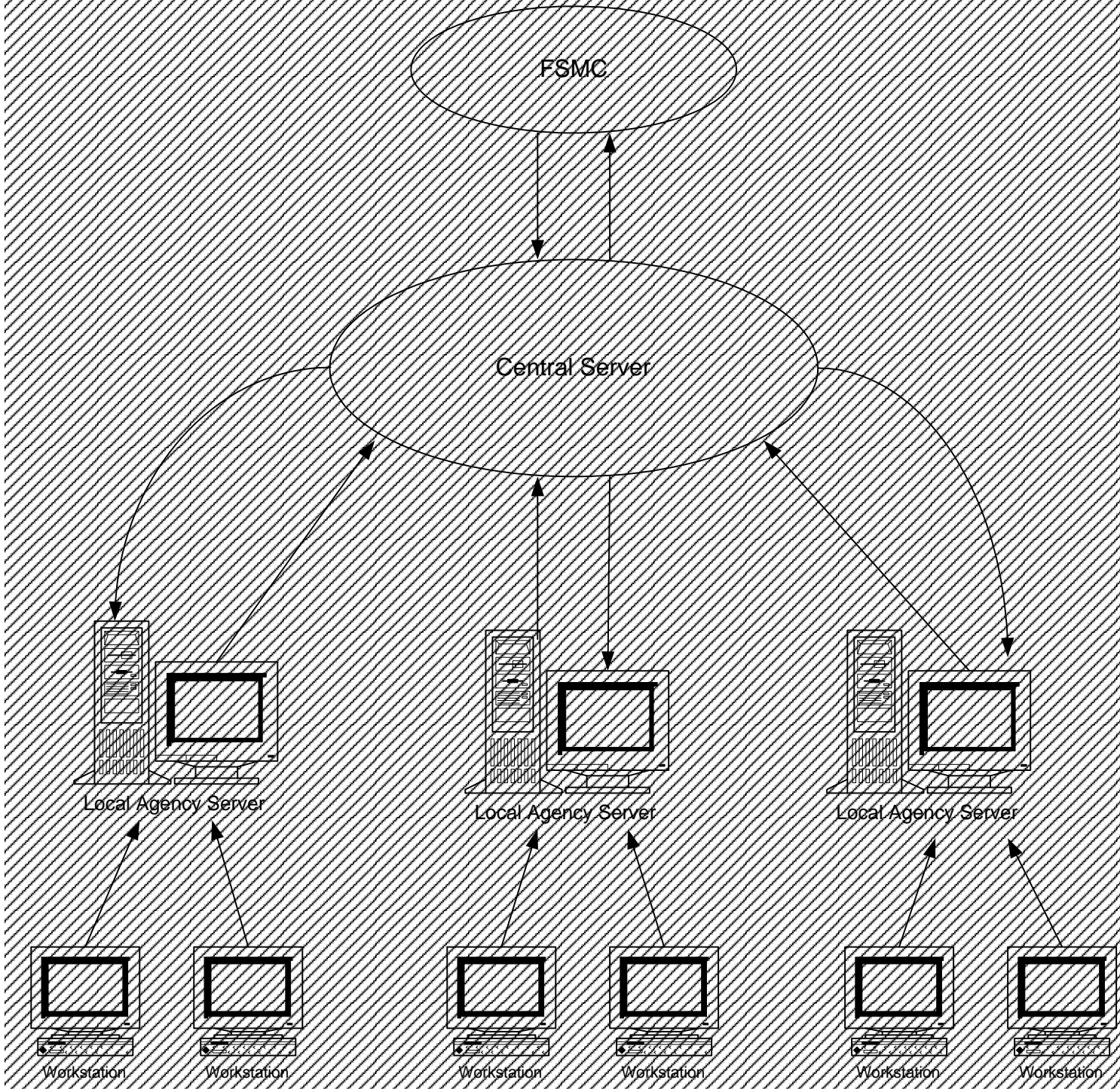
Additionally, end of day will perform the following tasks:

<b>Processes</b>	<b>LA</b>	<b>Central</b>	<b>Bank</b>
Perform backup	x	x	
Delete future appointments for participants who have transferred from one clinic to another within an agency	x		
Automatic termination of participants no longer eligible (omit if done during clinic day)	x		
Perform automatic category changes	x		
Mark appointments as Missed or Kept	x		
Delete information no longer used by the system (Includes participants with no Certification Record after 60 days on system)	x		
Set exclusively breastfeeding participants to active status	x		
Void FIs not deposited after the stale date		x	
Generate notice of ineligibility and appointment letters	x		
Gather updated participant information, new issuance information, void information, requests for transfer, and outreach org. information	x		
Send Updated Local Agency information to Central server	x	x	
Delete information no longer used by the system	x	x	
Compile direct payment and revalidated FI information and send to bank		x	x
Compile vendor information to send to bank		x	
Consolidate participant information		x	
Financial - process totals and store in F_CASELOAD table. Totals are based on priority, language, caseload type, ethnic group and poverty level		x	
Update caseload assignment information from the clinics		x	
Consolidate and update all food instrument data to calculate FI obligation value		x	
Search for potential dual enrollment participants		x	
Compile new issuance and void information to send to bank		x	x

<b>Processes</b>	<b>LA</b>	<b>Central</b>	<b>Bank</b>
Poll bank, send and receive FI and Vendor information		X	X
Create consolidated record of all FI issuance and redemption information and create vendor payment records		X	
Create date of current food funds available from all sources as well as current food obligations and expenditures to establish current cash position		X	
Financial (end of month) - Populate F_INCOME to POVERTY table. These values are used in populating of caseload table		X	
Vendor calculate and update peer group averages		X	
Vendor run and print analysis factor reports		X	
Gather new/updated data to be sent to the agencies		X	
Prepare archival retrieval data		X	
Print out status logs		X	
*EOD Batch Log		X	
Consolidate information from central	X		
Transmit System Tables	X		
Update System Tables	X		
*Clinical serial number replenishing		X	
*Statewide summary of participants for dual enrollment check	X		
Update caseload assignment information from the clinics	X		
Send redemption/rejection information to agencies	X		
Transmit archival parts. and vendors and update records at agencies	X	X	
*Vendor monthly reports		X	
*Vendor quarterly reports		X	
Print out status logs of the process	X		

New End of Day processes

# End of Day Process Flow



## 1.1 **Perform back-up**

### *Overview*

The system performs a full backup of the application at the End of Day process using script shutaimxx.cmd where xx is the agency code.

### *Processes:*

The system initiates the following NT commands on each Local Agency server to perform the backup:

The database shutdown is done using Oracle commands.

```
c:\winnt\system32\ntbackup c:\arizona d:\oracle c:\orant d:\arizona
/b /d "complete backup for Arizona oracle server" /e
/hc:on /l "d:\aim\eod\agency\logs\full_bkp.log"
/t normal
```

```
c:\winnt\system32\ntbackup backup c:\ d:\ /a /d "Incremental
backup of all drives" /hc:on /t incremental /l
d:\aim\cod\agency\logs\inc_bkp.log
```

The database startup is done using Oracle commands.

## 1.2 **Automatic termination of clients no longer eligible**

### 1.2.1 EA1\_TERM\_EOD.SQL

During the End of Day process automatic terminations\* are performed as follows:

- The WIC participant is postpartum and today's date is 180 days after the actual delivery date.
- The CSF participant is postpartum and today's date is the last day of the calendar month of the Actual Delivery date +12 calendar months.
- The WIC participant is breastfeeding and today's date is 365 days after the actual delivery date.
- The CSF participant is breastfeeding and today's date is the last day of the calendar month of the Actual Delivery date + 12 calendar months.
- The WIC participant is 5 years of age and today's date is the last day of the calendar month in which the participant turned 5 years of age.
- The CSF participant is 6 years of age and today's date is the last day of the calendar month in which the participant turned 6 years of age.
- The WIC or CSF participant's current certification period has expired and there is not a new certification period.

\*For all Participant records terminated all non-certification future appointments for the participant are deleted for the End of Day process.

## 1.2.2 EA1\_TERM\_WL\_CLIENTS.SQL

**Overview**

This script updates client records terminated if the clients are on the waiting list for more than the allotted time requirements for the stated waiting list, currently set to 180 days. (this does not apply to clients classified in check out clinic status.)

**1.3 Perform automatic category changes**

## 1.3.1 EA1\_CAT\_EOD.SQL

**Overview**

This script performs automatic category changes for clients based upon their certification category codes, current date and birth date. The following table outlines updates that take place on the client certification record (*does not apply to clients classified in check out clinic status*):

Old code	Months between current month and birth date	New code
IEN, IPN, IFF	12	C1
C1	24	C2
C2	36	C3
C3	48	C4
C4	60	C5

## 1.3.2 EA1\_CAT\_SYNC\_EOD.SQL

**Overview**

Client record's category code is kept in sync with the certification record's category code. In case of a future certification, c\_clients.cat\_category\_code maintains the current certification category until the current cert period ends. When the current cert period ends, this eod process will change the c\_clients.cat\_category\_code to equal the new c\_certifications.cat\_category\_code.

**1.4 Mark appointments as Missed or Kept**

## 1.4.1 EA1\_APPT\_EOD.SQL

**Overview**

Client appointment status is updated from pending (P) to kept (K) depending on the following conditions (*does not apply to clients classified in check out clinic status*):

- If food instruments have been issued for the appointment month
- *AND* the disposition of any food instrument is issued
- *AND/OR* medical or bloodwork records create date is greater than or equal to the appointment date

Client appointment status is updated from pending (P) to missed (M) if the above conditions are not met.

Also update all the exclusive nursing infants appointment as kept (K) when mother's appointment are marked as kept (K).

#### 1.4.2 EA1\_NEW\_CLASS\_EOD.SQL

Client's class appointment status is updated from pending (P) to kept (K) based on the following criteria (does not apply to clients classified in check out clinic status):

- If food instruments have been issued to the clients under that family for the appointment
- *OR* bloodwork data done after or equal to appointment date
- *OR* medical info entered on or after appointment date

Class appointment status is marked from pending (P) to kept (K) if above conditions are met. For 'walkin' status just mark it as 'walkin' else 'missed'

#### 1.4.3 EA1\_NEW\_APPT\_EOD.SQL

Client appointment status is marked as 'missed', 'kept', or 'walkin' for the new appointment scheduler depending on the following conditions (*does not apply to clients classified in check out clinic status*)

Pending to kept

- If FI's have been issued for the appointment month for any member of the family
- *OR* bloodwork data done after or equal to appointment date

Also update all the exclusive nursing infants appointment as kept (K) when mother's appointment are marked as kept (K).

- If one of the clients appt status is 'walkin' then mark it as 'walkin' else 'missed'

### 1.5 **Delete information no longer used by the system**

#### 1.5.1 EA1\_OTHER\_PURG.SQL

##### **Overview**

This script deletes records that are no longer used by the system. The following outlines records that are deleted and any conditions relating to the deletion of these records (*does not apply to clients classified in check out clinic status*):

- Client records will be deleted, as well as their child records, if the client has no certification or the client record has no appointment date.
- The family records will be deleted, as well as the child records, if the family records have no related client records.
- The food instrument records will be deleted if the issue date is greater than twenty four months ago.

During the End of Day process the system purges precertified participants with no further record activity after 120 days. If a participant is wait listed on this screen the record is not purged by the End of Day process after 60 days.

- Transfer requests that are more than 1 month old
- Income history records that do not have any dependent income records
- Leftover records in the temporary food instrument table
- Archived client records created more than 15 days in the past

**1.5.2 EA1\_PURGE\_OLD\_APPT.SQL**

Delete 180 days old appointments from old appointment scheduler.

**1.5.3 EA1\_PURGE\_NEW\_APPT.SQL**

Delete 90 days old appointments from new appointment scheduler.

**1.6 Set exclusively breast feeding clients to active status****1.6.1 EA1\_IEN\_ACTIVE.SQL*****Overview***

Clients which are exclusively breast feeding have their records updated to a status of active (A) from a record status of pending (P). The category code that makes a client exclusively breast feeding is category code of IEN (*does not apply to inactive infants in check out clinic status*).

**1.7 Void FIs not deposited after the stale date****1.7.1 EC1\_STATE\_DATE\_FI.SQL*****Overview***

This script updates food instrument records with a disposition code of a '1' '2' or '5' and a stale date less than the current date. The related records are updated with a disposition code of '3' and a void reason code of 'H' (*does not apply to food instruments in check out clinic status*).

**1.8 Process pending transfer requests****1.8.1 EA1\_TRANSFER.SQL*****Overview***

This script completes any pending transfer requests that could not be performed during the day.

**1.9 Generate notice of ineligibility and appointment letters****1.9.1 CS\_LETTER\_OF\_TERM.FMX*****Overview***

This form prints out the ineligibility notices for clients who have been terminated by ea1\_term\_eod and ea1\_term\_wl\_clients. This also prints appointment notices at a scheduled number of days before the client's appointment.

*Currently Arizona is not using this functionality.*

**1.10 Gather updated client information, new issuance information, void information, and requests for transfer****1.10.1 EA1\_TAB\_UP.SQL*****Overview***

This script extracts client information that has been changed or created. It compares the change or created date in each of the following tables to the last day end of day processes ran successfully.

- Staff members
- Staff phones
- Staff time studies
- Job descriptions
- Food instruments
- Food instrument types
- Food instrument foods
- Food instrument formulas
- Food packages
- Food package food instrument types
- Food package foods
- Food package prescriptions
- FI formulas
- Package distributions
- Organization units
- Organization programs
- Organization unit phones
- Outreach organizations
- Outreach comms
- Outreach organizations phones
- Outreach programs
- Outreach org las
- Family economic units
- Family phones
- Family referrals
- FEU communications
- Clients
- Clients BP Issuances
- Client communications
- Client goals
- Client notes
- Client programs
- Client referrals
- Client services
- Client more ethnic groups
- Client NE topics
- Client services NE materials
- Client previous names
- Certified peer counsels

- Counsel notes
- Client allergy foods
- Peer Rbfends
- Contacts
- Income histories
- Economic unit members
- Incomes
- Immunizations
- Infant data
- Infant child medicals
- Resolutions
- Resolved clients
- Transfer information
- Transfer histories
- Certifications
- Certification term reasons
- Cert nutr eds
- Bloodwork data
- Dietary assessments
- Diet nutrients
- Health risk factors
- C Food Package Prescriptions
- C B W healths
- C B W HH reasons bfends
- C I C healths
- C I C HH reasons bfends
- C P healths
- C P S responses
- C food box distributions
- C infant child medicals
- C woman medicals
- F case assignments
- F caseload restrictions
- F wait lists
- F wait list contacts
- I BF promotional vouchers
- I BF voucher items
- I FI inventories
- I FI ranges
- I inventories
- I inventory organizational units
- I stock inventories
- I inv org units
- V complaints
- V monthly activities
- V activity findings
- Delete statements
- Batch runs

- Printer Addresses

### 1.10.2 EA1\_EXPORT\_TAB.TXT

#### *Overview*

Tables that will be exported. The list above in section 1.10.1 are the tables that will be exported.

### **1.11 Transfer datafile from agency to central**

#### 1.11.1 EA1\_FTP\_RUN.BAT/ EA1\_FTP\_TO\_CENTRAL.BAT

This end of day batch process connects to central ftp-server from the agency site and copies the datafile to ftp-server for central processing (containing clients information).

### **1.12 Delete information no longer used by the system**

#### 1.12.1 EC1\_OTHER\_PURG.SQL

#### *Overview*

This script deletes various records that are no longer used by the system. The following records are removed if conditions are met (*does not apply to clients classified in check out clinic status*):

<b>Records deleted</b>	<b>Conditions</b>
Delete statements	Del-statements with date_created more than 2 months in the past
Income history records	Delete records that don't have corresponding records in c_incomes table
Temp FI records	Delete any records remaining in i_temp_fis because of printing problems
Archived/client records	Delete r_archived_clients records with date_created more than 15 days in the past

##### 1.12.1.1 Inputs

DEL\_STATEMENTS

##### 1.12.1.2 Selection

Determines which record are no longer used by the system. This is done in the following processing step in the where clause.

##### 1.12.1.3 Processing

Deletes unused records:

```
DELETE FROM DEL_STATEMENTS
WHERE date_created < min_date (min date is two months prior to current date).
```

```
DELETE FROM C_ECONOMIC_UNIT_MEMBERS
WHERE cih_income_history_id = inc_rec.income_history_id (inc_rec.income_history_id is any
income_history_id that does not have corresponding c_income records)
```

```
DELETE from C_INCOME_HISTORIES  
WHERE income_history_id = inc_rec.income_history_id
```

```
DELETE from I_TEMP_FIS
```

```
DELETE from R_ARCHIVED_CLIENTS  
WHERE NVL (date_modified, date_created) < (sysdate -15)
```

#### 1.12.1.4 Outputs

del\_statements

### 1.13 Compile direct payment and revalidated FI information and send to bank

#### 1.13.1 EC1\_CTRL\_CHK.SQL

##### *Overview*

This script creates bank report records for direct payment, compliance cases and replacement checks (issue method 'R'). It also updates bank report records for records that have been revalidated since the last run (revalidation code '2').

### 1.14 Compile vendor information to send to bank

#### 1.14.1 EC1\_VENDOR\_FSMC.BAT

##### *Overview*

This batch process creates a vendor text file that will be sent to the bank. If the text file already exists the current data will be appended to the existing data file. If the file doesn't exist it will be created. Both the append and the create use a temporary data set that is deleted at the end of the batch process.

#### 1.14.2 EC1\_VENDOR\_FSMC.SQL

##### *Overview*

This script creates a temporary vendor data set to be used for bank processing. The temporary data set is created based upon all vendor data in the system.

#### 1.14.3 EC1\_PGA\_FSMC.SQL

##### *Overview*

This script creates a temporary text file of vendor peer group averages data for bank processing. The file consists of all peer group average data in the system. All records are sent to the bank even if only one has been changed.

This batch process sends a vendor peer group average text file to the bank. If the last output file created has data in it, it is sent to the bank. However, if the last created output file doesn't contain data the old data file is sent to the bank.

#### 1.14.4 EC1\_OUTBOUND\_VENDOR\_FSMC.SQL

This script generalizes the state's formatted audit file to track vendor data sent to FSMC.

#### 1.14.5 EC1\_OUTBOUND\_PGA\_FSMC.SQL

This script creates a text file of peer group averages data sent to FSMC.

### **1.15 Consolidate client information**

#### 1.15.1 EC2\_SP\_TRNC.SQL

##### **Overview**

This script initializes the following tables:

- Food instruments
- Food instrument types
- Food instrument foods
- Food instrument formulas
- Food package food instrument types
- Temporary c\_state\_clients table
- Temporary caseload tables

#### 1.15.2 EC2\_TRNC.SQL

##### **Overview**

This script initializes the following tables:

- End of day controls
- Staff members
- Organizational units
- Organizational programs
- Organizational unit phones
- Outreach organizations
- Outreach organization LAS
- Outreach comms
- Outreach organization phones
- Outreach programs
- Staff phones
- Job descriptions
- Family economic units
- Family phones
- Family referrals
- FEU communications
- Smoking Histories
- Previous Families
- Clients
- Client communications
- Client goals
- Client notes
- Client programs
- Client referrals
- Client Services
- Client more ethnic groups
- Client Previous Names

- Client BF Promo Insurance
- Client Allergy Foods
- Resolved clients
- Resolutions
- Cert Nutr Eds
- Food box distribution
- Client NE topics
- Client services NE materials
- Certificated peer counsels
- Contacts
- Peer Fbfends
- Counsels notes
- Incomes
- Income histories
- Economic unit members
- Immunizations
- Infant data
- Infant child medicals
- Resolutions
- Resolved clients
- Transfers information
- Transfer histories
- Certifications
- Certification Peer Counsels
- Bloodwork data
- Certification term reasons
- Dietary assessments
- Diet nutrients
- Health risk factors
- Woman medicals
- C B W healths
- C B W HH reasons bfends
- C I C healths
- C I C HH reasons bfends
- C P healths
- Case assignments
- Caseload Restrictions
- PS Responses
- Client BP Issuances
- F Wait lists
- F Wait lists contacts
- I Food packages
- I Food packages food
- I Food package prescriptions
- I Package distributions
- I BF promotional vouchers
- I BF voucher items
- I FI inventories

- I FI ranges
- I inventories
- I inventory organizational units
- V complaints
- V monthly activities
- V activity findings
- Delete statements

#### 1.15.3 EC2\_DELTRIG.SQL

##### ***Overview***

This script deletes data from the central database that had been deleted from the local agency database since the last end of day run other than client and family base information.

#### 1.15.4 EC2\_DELTRIG\_AV.SQL

##### ***Overview***

This script deletes client and family base information from the central database.

#### 1.15.5 EC2\_IN\_TAB.SQL

##### ***Overview***

This script inserts client information that has been changed or created. It compares the change or created date in each of the following tables to the last day end of day processes ran successfully.

- Staff members
- Staff time studies
- Organizational units
- Organizational programs
- Organizational unit phones
- Outreach organizations
- Outreach\_org las
- Outreach comms
- Outreach organization phones
- Outreach programs
- Staff phones
- Job descriptions
- Family economic units
- Family phones
- Family referrals
- FEU communications
- C Food Package Prescriptions
- C PS Responses
- C\_food\_box distributions
- Clients
- Client BP Issuances
- Client More Ethnic Groups
- Client Previous Names
- Client communications
- Client goals

- Client notes
- Client programs
- Client referrals
- Client Services
- Client NE topics
- Client services NE materials
- Cert Nutr Eds
- Certificated peer counsels
- Contacts
- C Allergy foods
- Peer Fbfends
- Counsels notes
- Incomes
- Income histories
- Economic unit members
- Immunizations
- Infant data
- Infant child medicals
- Resolutions
- Resolved clients
- Transfers information
- Transfer histories
- Certifications
- Bloodwork data
- Certification term reasons
- Dietary assessments
- Diet nutrients
- Health risk factors
- Woman medicals
- C B W healths
- C B W HH reasons bfends
- C I C healths
- C I C HH reasons bfends
- C P healths
- C infant child medicals
- C woman medicals
- Case assignments
- F Wait lists
- F Wait lists contacts
- I Food packages
- I Food packages food
- I Food package prescriptions
- I Package distributions
- I BF promotional vouchers
- I BF voucher items
- I FI inventories
- I FI ranges
- I FI formulas

- I inventories
- I inventory organizational units
- I stock inventories
- Batch Runs
- Printer Addresses
- V complaints
- V monthly activities
- V activity findings
- Delete statements

#### 1.15.6 EC2\_TAB\_UP.SQL

##### *Overview*

This script updates client information that has been changed or created. It compares the change date in each of the following tables to the last day end of day processes ran successfully.

- Staff members
- Staff time studies
- Organizational units
- Organizational programs
- Organizational unit phones
- Outreach organizations
- Outreach\_org las
- Outreach comms
- Outreach organization phones
- Outreach programs
- Staff phones
- Job descriptions
- Family economic units
- Family phones
- Family referrals
- FEU communications
- C Food Package Prescriptions
- C PS Responses
- C\_food\_box distributions
- Clients
- Client BP Issuances
- Client More Ethnic Groups
- Client Previous Names
- Client communications
- Client goals
- Client notes
- Client programs
- Client referrals
- Client Services
- Client NE topics
- Client services NE materials
- Cert Nutr Eds
- Certificated peer counsels

- Contacts
- C Allergy foods
- Peer Fbfends
- Counsels notes
- Incomes
- Income histories
- Economic unit members
- Immunizations
- Infant data
- Infant child medicals
- Resolutions
- Resolved clients
- Transfers information
- Transfer histories
- Certifications
- Bloodwork data
- Certification term reasons
- Dietary assessments
- Diet nutrients
- Health risk factors
- Woman medicals
- C B W healths
- C B W HH reasons bfends
- C I C healths
- C I C HH reasons bfends
- C P healths
- C infant child medicals
- C woman medicals
- Case assignments
- F Wait lists
- F Wait lists contacts
- I Food packages
- I Food packages food
- I Food package prescriptions
- I Package distributions
- I BF promotional vouchers
- I BF voucher items
- I FI inventories
- I FI ranges
- I FI formulas
- I inventories
- I inventory organizational units
- I stock inventories
- Batch Runs
- Printer Addresses
- V complaints
- V monthly activities
- V activity findings

- Delete statements

**1.16 Financial - process totals and store in F\_CASELOAD table. Totals are based on priority, language, caseload type, ethnic group and poverty level**

1.16.1 EC2\_CL\_REC\_PROCESS.SQL

**Overview**

This end of day script is used to insert updated or inserted client/family/food instruments records from the agencies into caseload temporary table to process at the end of all the agency uploads on central database.

1.16.2 EC2\_DEL\_CL\_FUT\_RECS.SQL

**Overview**

This script is used to delete future client records with respect to the old local agency when the client is transferred. This needs to be executed at the central site after the end of day data from all local agencies is received.

**1.17 Update caseload assignment information from the clinics**

1.17.1 EC2\_SYNC\_CLINIC\_ASSIGNMENT.SQL

**Overview**

This script is used at state site for end of day processes to synchronize the caseload assignment data. The caseload assignments are re-allocated by the local agency to their clinics. After this processing the records are deleted from the eodadm.e\_f\_case\_assignments table.

1.17.2 EC2\_populate\_clinic.SQL

**Overview**

Calls database procedure caseload.daily\_fctd\_past and caseload.daily\_fctd\_current to update the caseload information from the temporary table constructed during ec2\_cl\_rec\_process.

**1.18 Consolidate and update all food instrument data to calculate FI obligation value**

1.18.1 EC2\_FOOD\_UP.SQL

**Overview**

Calls database procedure eod\_fi\_process.ec2\_fi\_upd. This script calls fi\_type\_process which takes care of copying the food instrument type tables from the temporary tables of the local agency. EOD\_FI\_PROCESS\_EC2\_UPD procedures inserts/updates the food instrument records at the central server from agency servers.

1.18.2 EC2\_FI\_UPD.SQL

**Overview**

Executes EC2\_FI\_UPDATES.SQL. This procedure processes the checks that are received from the local agency and updates/creates respective data in the Financial tables.

## 1.18.3 EC2\_FORMULAS.SQL

**Overview**

This script updates food instrument formulas information that has been changed or created. It compares the change or created date in each of the following tables to the last day end of day processes ran successfully.

*Arizona WIC doesn't use this table so this script has been commented out.*

**1.19 Search for potential dual enrollment clients**

## 1.19.1 EC2\_DUAL\_ENROLL.SQL

**Overview**

This script is used to verify clients do not have a dual enrollments.

**1.20 Compile new issuance and void information to send to bank**

## 1.20.1 EC2\_ISSUE\_FSMC.SQL

**Overview**

This script creates a text file of bank report data records with the following conditions:

Conditions
Send Code of B
Disposition code of 2 AND Issue method of B, C, O, R, and P
OR Disposition code of 3
OR Disposition code of 5 AND Revalidation code of 2

**1.21 Poll bank, send and receive FI and Vendor information**

## 1.21.1 EC4.BAT

**Overview**

This batch process sends and receives food instrument and vendor data from the bank. It checks that the above data files are contained within the bank system.

**1.22 Create consolidated record of all FI issuance and redemption information and create vendor payment records**

## 1.22.1 EC4\_BNK\_POST.SQL

**Overview**

This script creates consolidated food instrument issuance and redemption information and creates vendor payment records

**1.23 Create date of current food funds available from all sources as well as current food obligations and expenditures to establish current cash position**

## 1.23.1 EC4\_CASHFLOW\_UPD.SQL

**Overview**

This EOD script is used to update summary information in caseloads table for each distinct month/year where data has been changed in obligations and outlays.

**1.24 Financial (end of month) - Populate F\_INCOME to POVERTY table. These values are used in populating of caseload table**

## 1.24.1 EC4\_MON\_END.SQL

**Overview**

This end of day is used to perform population of income and poverty tables at the end of month. It may create a translation dataset in incomes to poverties table. Added new process for monthly caseload updates.

**1.25 Vendor (end of month) calculate and update peer group averages**

## 1.25.1 EC4\_VEN\_PEER\_FI.SQL

**Overview**

Updates or inserts into peer group averages for food instruments disposition code of '4'. The following data elements are updated:

- Total redemption amount
- Average participants
- Average redemption amount
- Maximum redemption amount
- Total redeemed FIS
- Recalculate failed flag
- Date modified
- Modified by

## 1.25.2 EC4\_VEN\_LOC\_AGCY\_FI

**Overview**

Updates or inserts into local agency averages for food instruments disposition code of '4' (redeemed). The following data elements are updated:

- Total redemption amount
- Average redemption amount
- Total redeemed FIS
- Survey flag
- Date modified
- Modified by

**1.26 Vendor run and print analysis factor reports**

## 1.26.1 EC4\_VEN\_PRICE.SQL

**Overview**

Vendors purchasing analysis and ranking reports are produced from this script. The FI cost for Vendors is calculated and stored, based on their Food Price Surveys. Vendors are ranked by average maximum price, redemption value and peer group identification.

**1.27 Generate dual enrollment report to be sent to agencies**

## 1.27.1 EC4\_DUAL\_RPT.BAT

**Overview**

This is a batch file which is called from EC4.BAT. The batch file runs the Dual Enrollment/Participation report which identifies potential dual enrollment clients.

*Arizona WIC is not using this script anymore.*

## 1.27.2 EC4\_DUAL.REP

**Overview**

This report is called by EC4\_DUAL\_REP.BAT. The report reflects dual enrollment participants as described in section 2-4.3.16 – Dual Enrollment Participation in the Enrollment and Certification Detailed Functional Design Document.

*Arizona WIC is not using this script anymore.*

**1.28 Gather new/updated data to be sent to the agencies**

## 1.28.1 EC4\_TRNC.SQL

**Overview**

This script initializes the following tables:

- End of day Control
- E\_EOD\_CONTROLS
- A\_EOD\_CONTROLS
- A\_STATE\_CLIENTS

## 1.28.2 EC4\_TAB\_%.SQL

**Overview**

The EC4\_TAB\_DOWN script runs for each Local Agency

The primary purpose of this script is to propagate down to Local Agencies changes made to selected tables at the State level. Temporary tables are utilized to facilitate data recovery efforts should they be needed.

For each temporary table, the old data in the table is truncated and new rows and values are inserted where either the date modified or the date created column of the “source” table is

after the last successful run date for End Of Day Processing. For some temporary tables, additional criteria is employed.

#### 1.28.3 EC4\_AGCY\_BEING\_PROCESSED.SQL

This script will indicate the agency for which the data is updated on state in EOD process.

#### 1.28.4 EC4\_EXPORT\_TAB.TXT

This script has a list of all tables that need to send down to agency server from central server.

### 1.29 **Process transfer requests**

#### 1.29.1 EC2\_STATE\_CTI.SQL

##### **Overview**

If access to the Central database was not available, either because the user is at a satellite clinic or the network connection is not available the system records the transfer request and the transfer is completed during the next End of Day process. All of the associated Participant information and Participant history information is transferred over the network to the new agency/clinic. At the 'losing' agency all pending appointments are removed, the Participant is terminated, and the system records the 'transfer to' clinic information. During the end of day process a transfer log is generated at both the losing and gaining Local Agency.

*This process is not used by Arizona WIC EOD.*

#### 1.29.2 EC4\_STATE\_A\_CTI.SQL

##### **Overview**

This script will not exist in the Arizona system. CTI temporary files were necessary in the Hawaii system because of architecture constraints. These files will not be needed in the AIM system due to the availability of a Wide Area Network in Arizona.

*This process is not used by Arizona WIC EOD.*

### 1.30 **Prepare archival retrieval data**

#### 1.30.1 EC1\_PRG\_ARCHV.SQL

##### **Overview**

Purge archived client records marked with an archive flag of 'Y'. The deletes have been ordered from the smallest level child up to the parent (c\_clients). Purge archived vendor records marked with an archive flag of 'Y'. The deletes have been ordered from the smallest level child up to the parent (v\_vendors).

#### 1.30.2 EC4\_RETRIEVE.SQL

##### **Overview**

Run the following scripts:

- EC4\_TRUNC\_ARCH\_CLI.SQL
- EC4\_INS\_RETR\_CLI.SQL
- EC4\_TRUNC\_ARCH\_VEN.SQL

- EC4\_INS\_RETR\_VEN.SQL

### 1.30.3 EC4\_TRUNC\_ARCH\_CLI.SQL

#### *Overview*

Delete all data from temporary tables before loading data.

### 1.30.4 EC4\_INS\_RETR\_CLI.SQL

#### *Overview*

Performs insert from Client tables into all temporary tables.

### 1.30.5 EC4\_INS\_RETR\_VEN.SQL

#### *Overview*

Performs insert from Vendor tables into all temporary tables.

## **1.31 Print out status logs**

### 1.31.1 EC\_PRINT.BAT

#### *Overview*

This batch file produces print outs by agency for the following log files:

- Truncate
- Import
- Delete
- Insert
- Update
- End of day import
- End of day insert
- End of day retrieve
- Export
- End of Day Not Run Log

## **1.32 EOD Batch Log**

### 1.32.1 EOD\_LOG.BAT

#### *Overview*

This report is produced daily at the Central Site during the End of Day process whether or not the function was used.

*Arizona WIC EOD is not using this process.*

## **1.33 Transfer datafile from ftp server to agency server**

### 1.33.1 EA3\_FTP\_RUN.BAT/ EA3\_FTP\_FROM\_CENTRAL.BAT

#### *Overview*

This batch file gets the agency database update from central.

### 1.34 Consolidate information from central

#### 1.34.1 EA3\_TRNC.SQL

##### *Overview*

This script initializes the following tables:

- A\_STATE\_CLIENTS
- C\_STATE\_CLIENTS
- F\_POVERTY\_BASES
- EOD\_CONTROLS
- A\_ACTIVITIES
- A\_APPT\_ITEMS
- AAS\_CLASS\_CATEGORIES
- AAS\_ITEMS
- A\_ATTEND\_STATUSES
- A\_OFFICE\_CLOSED
- A\_SERVICES
- I\_AGE\_RANGES
- C\_ANSWERS
- C\_ANSWER\_TYPES
- C\_BLOODWORK\_TYPES
- C\_BREAST\_PUMP\_REASONS
- C\_BREAST\_PUMP\_TYPES
- C\_CATEGORIES
- C\_CAT\_BLOOD\_FACTORS
- C\_CAT\_GOALS
- C\_CAT\_NUTR\_EDS
- C\_CAT\_REFERRALS
- C\_COMMUNICATION\_TYPES
- C\_CP\_MESSAGES
- C\_DIAGNOSES
- C\_DIETARY\_REQUIREMENTS
- C\_DIET\_NUTRIENT\_TYPES
- C\_DISABILITIES
- C\_EDUCATION\_LEVELS
- C\_ELEVATIONS
- C\_ETHNIC\_GROUPS
- C\_GOALS
- C\_HC\_PAYEES
- C\_HH\_QUESTIONS
- C\_HH\_RESPONSES
- C\_INCOME\_INTERVALS
- C\_INCOME\_LEVELS
- C\_INCOME\_SOURCES
- C\_INCOME\_VERIFICATIONS
- C\_INFANT\_STATUSES
- C\_LANGUAGES
- C\_MARITAL\_STATUSES

- C\_NCHS\_CLASSIFICATIONS
- C\_NCHS\_TYPES
- C\_NCHS\_DATA
- C\_NO\_CONTACT\_REASONS
- C\_NUTR\_ED\_MATERIALS
- C\_NUTR\_ED\_TOPICS
- C\_PICKUP\_INTERVALS
- C\_PILOT\_QUESTIONS
- C\_PILOT\_STUDIES
- C\_PRIORITIES
- C\_PROOF\_ADDRESSES
- C\_PROOF\_IDENTITIES
- C\_PS\_QUESTIONS
- C\_RACES
- C\_REASONS\_BF\_ENDED
- C\_RISK\_FACTORS
- C\_RISK\_FACTOR\_DIAGNOSES
- C\_RISK\_FACTOR\_TYPES
- C\_RF\_GOALS
- C\_RF\_NUTR\_EDS
- C\_RF\_REFERRALS
- C\_RF\_SERVICES
- C\_HH\_RESPONSE\_RFS
- C\_BMI\_DATA
- C\_BMI\_ANTHROPOMETRIC
- C\_BMI\_WEIGHT\_GAIN
- C\_SCHEDULE\_DAYS
- C\_SMOKING\_CHANGES
- C\_SMOKING\_STAGES
- C\_SOURCES\_HEALTH\_CARE
- C\_SYMPTOMS
- C\_TERM\_REASONS
- C\_TOPICS
- C\_VOTER\_REGISTRATIONS
- C\_CAT\_BLOODWORKS
- C\_DESIREABLE\_WEIGHTS
- C\_IMMUNIZATIONS\_NOT\_ASSESSED
- F\_CONTROLS
- F\_CASELOAD\_TYPES
- F\_FUND\_SOURCES
- F\_POVERTY\_LEVELS
- F\_WAIT\_LIST\_RESPONSES
- I\_BANK\_DISPOSITIONS
- I\_CATEGORY\_GROUPS
- I\_PACKAGES
- I\_PRODUCTS
- I\_REJECT\_REASONS
- I\_CONTAINERS

- I\_DISPOSITIONS
- I\_UNITS\_OF\_MEASURE
- I\_VOID\_REASONS
- I\_FOOD\_GROUPS
- I\_FOODS
- I\_FOOD\_DISTRIBUTIONS
- I\_MAXIMUM\_FOODS
- I\_FOOD\_PACKAGES
- I\_CATEGORY\_GROUP\_PKGS
- I\_FOOD\_PACKAGE\_FI\_TYPES
- I\_FOOD\_PACKAGE\_FOODS
- I\_PACKAGE\_DISTRIBUTIONS
- I\_WS\_FOOD\_GROUPS
- I\_FOOD\_INSTRUMENT\_TYPES
- I\_FOOD\_INSTRUMENT\_FOODS
- C\_FOOD\_PKG\_RISK\_FACTORS
- O\_OUTREACH\_ORG\_TYPES
- O\_OUTREACH\_COMM\_TYPES
- O\_OUTREACH\_ORGANIZATIONS
- O\_OUTREACH\_ORG\_PHONES
- O\_OUTREACH\_COMMS
- O\_PROGRAMS
- O\_PROGRAM\_DATES
- O\_TITLE\_CATEGORIES
- O\_STAFF\_TITLES
- S\_CITIES
- S\_CONTACT\_METHODS
- S\_CONTACT\_TITLES
- S\_COUNTIES
- S\_PHONE\_TYPES
- S\_STATES
- S\_ZIPS
- V\_ACTIVITY\_TYPES
- V\_APPLICATION\_MILESTONES
- V\_APPLICATION\_TYPES
- V\_BANK\_BRANCHES
- V\_CASE\_STATUSES
- V\_COLLECTION\_TYPES
- V\_COMMUNICATION\_TYPES
- V\_COMPLAINT\_SOURCE\_TYPES
- V\_COMPLAINT\_STATUSES
- V\_COMPLAINT\_SUBJECTS
- V\_COMPLIANCE\_CASE\_DESIGNATNS
- V\_COMPLIANCE\_CASE\_TYPES
- V\_DELIVERY\_TYPES
- V\_DENIAL\_REASONS
- V\_DISQUAL\_REASONS
- V\_FINDING\_CODES

- V\_FSP\_REGIONAL\_OFFICES
- V\_FSP\_VIOLATION\_CODES
- V\_LEGAL\_FIRMS
- V\_LEGAL\_REPRESENTATIVES
- V\_MILESTONE\_TYPES
- V\_OWNER\_TYPES
- V\_PEER\_GROUPS
- V\_RISK\_LEVELS
- V\_SANCTION\_TYPES
- V\_STATUSES
- V\_SUSPENSION\_REASONS
- V\_VIOLATION\_ACTION
- V\_WHOLESALERS
- V\_WIC\_CODES
- V\_OWNERS
- V\_VENDORS
- O\_STAFF\_MEMBERS
- O\_ORGANIZATIONAL\_UNITS
- O\_ANNUAL\_WIC\_COST\_SUMMARIES
- F\_ANNUAL\_FACTORS
- F\_BUDGETS
- F\_CASE\_ASSIGNMENTS
- F\_CASELOADS
- F\_CASELOAD\_RESTRICTIONS
- F\_MFR\_CONTACTS
- F\_MANUFACTURERS
- I\_FI\_INVENTORIES
- I\_STOCK\_INVENTORIES
- I\_FOOD\_INSTRUMENTS
- I\_FI\_REJECT\_REASONS
- C\_RESOLUTIONS
- C\_RESOLVED\_CLIENTS
- S\_GEO\_LOCATIONS
- S\_PRINTER\_ADDRESSES
- DEL\_STATEMENTS
- R\_S\_CLIENT\_ARCHIVES
- R\_ARCHIVED\_CLIENTS
- R\_C\_FAMILY\_ECONOMIC\_UNITS
- R\_C\_FAMILY\_PHONES
- R\_C\_FAM\_REFERRALS
- R\_C\_FEU\_COMMUNICATIONS
- R\_C\_SMOKING\_HISTORIES
- R\_C\_PREV\_FAMILIES
- R\_C\_CLIENTS
- R\_C\_CERT\_PEER\_COUNSELS
- R\_C\_CONTACTS
- R\_C\_PEER\_RBFENDS
- R\_C\_P\_COUNS\_NOTES

- R\_C\_CLIENT\_COMMUNICATIONS
- R\_C\_CLIENT\_GOALS
- R\_C\_CLIENT\_NOTES
- R\_C\_CLIENT\_PROGS
- R\_C\_CLIENT\_REFERRALS
- R\_C\_CLIENT\_SERVICES
- R\_C\_CLIENT\_NE\_TOPICS
- R\_C\_CLIENT\_SVC\_NE\_MATERIALS
- R\_C\_MORE\_ETHNIC\_GROUPS
- R\_C\_PREV\_NAMES
- R\_C\_BF\_PROMO\_ISSUANCES
- R\_C\_ALLERGY\_FOODS
- R\_C\_W\_HEALTHS
- R\_C\_W\_HH\_REASONS\_BFENDS
- R\_C\_PS\_RESPONSES
- R\_C\_CLIENT\_BP\_ISSUANCES
- R\_C\_INCOME\_HISTORIES
- R\_C\_ECONOMIC\_UNIT\_MEMBERS
- R\_C\_INCOMES
- R\_C\_INFANT\_DATA
- R\_C\_RESOLUTIONS
- R\_C\_RESOLVED\_CLIENTS
- R\_C\_TRANSFER\_HISTORIES
- R\_C\_CERTIFICATIONS
- R\_C\_CERT\_NUTR\_EDS
- R\_C\_FOOD\_BOX\_DISTS
- R\_C\_BLOODWORK\_DATA
- R\_C\_CERT\_TERM\_REASONS
- R\_C\_DIETARY\_ASSESSMENTS
- R\_C\_DIET\_NUTRIENTS
- R\_C\_HEALTH\_RISK\_FACTORS
- R\_C\_INFANT\_CHILD\_MEDICALS
- R\_C\_WOMAN\_MEDICALS
- R\_C\_I\_C\_HEALTHS
- R\_C\_I\_C\_HH\_REASONS\_BFENDS
- R\_C\_FOOD\_PACKAGE\_PRESCRIPTIONS
- R\_I\_BF\_PROMO\_VCHRS

#### 1.34.2 EA3\_PRE.SQL

##### **Overview**

This script updates organizational units staff member id to null if it's not in the current agency.

*Arizona WIC does not use this script.*

#### 1.34.3 EA3\_DELTRIG.SQL

##### **Overview**

This script deletes data from the local agency database that had been deleted from the central database since the last end of day run.

## 1.34.4 EA3\_IN\_TAB.SQL

**Overview**

This script inserts records into AIM\* tables from end of day EODADM\* tables. The following tables:

- A\_ACTIVITIES
- AAS\_CLASS\_CATEGORIES
- AAS\_ITEMS
- A\_APPT\_ITEMS
- A\_ATTEND\_STATUSES
- A\_OFFICE\_CLOSED
- A\_SERVICES
- I\_AGE\_RANGES
- C\_ANSWERS
- C\_ANSWER\_TYPES
- C\_BLOODWORK\_TYPES
- C\_BREAST\_PUMP\_REASONS
- C\_BREAST\_PUMP\_TYPES
- C\_CATEGORIES
- C\_CAT\_BLOOD\_FACTORS
- C\_CAT\_GOALS
- C\_CAT\_NUTR\_EDS
- C\_CAT\_REFERRALS
- C\_COMMUNICATION\_TYPES
- C\_CP\_MESSAGES
- C\_DIAGNOSES
- C\_DIETARY\_REQUIREMENTS
- C\_DIET\_NUTRIENT\_TYPES
- C\_DISABILITIES
- C\_EDUCATION\_LEVELS
- C\_ELEVATIONS
- C\_ETHNIC\_GROUPS
- C\_GOALS
- C\_HC\_PAYEES
- C\_HH\_QUESTIONS
- C\_HH\_RESPONSES
- C\_INCOME\_INTERVALS
- C\_INCOME\_LEVELS
- F\_POVERTY\_BASES
- C\_INCOME\_SOURCES
- C\_INCOME\_VERIFICATIONS
- C\_INFANT\_STATUSES
- C\_LANGUAGES
- C\_MARITAL\_STATUSES
- C\_NCHS\_CLASSIFICATIONS
- C\_NCHS\_TYPES
- C\_NCHS\_DATA
- C\_NO\_CONTACT\_REASONS
- C\_NUTR\_ED\_MATERIALS

- C\_NUTR\_ED\_TOPICS
- C\_PICKUP\_INTERVALS
- C\_PILOT\_QUESTIONS
- C\_PILOT\_STUDIES
- C\_PRIORITIES
- C\_PROOF\_ADDRESSES
- C\_PROOF\_IDENTITIES
- C\_PS\_QUESTIONS
- C\_RACES
- C\_REASONS\_BF\_ENDED
- C\_RISK\_FACTORS
- C\_RISK\_FACTOR\_DIAGNOSES
- C\_RISK\_FACTOR\_TYPES
- C\_RF\_GOALS
- C\_RF\_NUTR\_EDS
- C\_RF\_REFERRALS
- C\_RF\_SERVICES
- C\_HH\_RESPONSE\_RFS
- C\_BMI\_DATA
- C\_BMI\_ANTHROPOMETRIC
- C\_BMI\_WEIGHT\_GAIN
- C\_SCHEDULE\_DAYS
- C\_SMOKING\_CHANGES
- C\_SMOKING\_STAGES
- C\_SOURCES\_HEALTH\_CARE
- C\_SYMPTOMS
- C\_TERM\_REASONS
- C\_TOPICS
- C\_VOTER\_REGISTRATIONS
- C\_CAT\_BLOODWORKS
- C\_DESIREABLE\_WEIGHTS
- C\_IMMUNIZATIONS\_NOT\_ASSESSED
- F\_CONTROLS
- F\_CASELOAD\_TYPES
- F\_FUND\_SOURCES
- F\_POVERTY\_LEVELS
- F\_WAIT\_LIST\_RESPONSES
- I\_BANK\_DISPOSITIONS
- I\_CATEGORY\_GROUPS
- I\_PACKAGES
- I\_PRODUCTS
- I\_REJECT\_REASONS
- I\_CONTAINERS
- I\_DISPOSITIONS
- I\_UNITS\_OF\_MEASURE
- I\_VOID\_REASONS
- I\_FOOD\_GROUPS
- I\_FOODS

- I\_FOOD\_DISTRIBUTIONS
- I\_MAXIMUM\_FOODS
- I\_FOOD\_PACKAGES
- I\_CATEGORY\_GROUP\_PKGS
- I\_FOOD\_PACKAGE\_FOODS
- I\_PACKAGE\_DISTRIBUTIONS
- I\_WS\_FOOD\_GROUPS
- I\_FOOD\_INSTRUMENT\_TYPES
- I\_FOOD\_PACKAGE\_FI\_TYPES
- I\_FOOD\_INSTRUMENT\_FOODS
- C\_FOOD\_PKG\_RISK\_FACTORS
- O\_OUTREACH\_ORG\_TYPES
- O\_OUTREACH\_COMM\_TYPES
- O\_PROGRAMS
- O\_PROGRAM\_DATES
- O\_TITLE\_CATEGORIES
- O\_STAFF\_TITLES
- S\_CITIES
- S\_CONTACT\_METHODS
- S\_CONTACT\_TITLES
- S\_COUNTIES
- S\_PHONE\_TYPES
- S\_STATES
- S\_ZIPS
- S\_GEO\_LOCATIONS
- V\_ACTIVITY\_TYPES
- V\_APPLICATION\_MILESTONES
- V\_APPLICATION\_TYPES
- V\_BANK\_BRANCHES
- V\_CASE\_STATUSES
- V\_COLLECTION\_TYPES
- V\_COMMUNICATION\_TYPES
- V\_COMPLAINT\_SOURCE\_TYPES
- V\_COMPLAINT\_STATUSES
- V\_COMPLAINT\_SUBJECTS
- V\_COMPLIANCE\_CASE\_DESIGNATIONS
- V\_COMPLIANCE\_CASE\_TYPES
- V\_DELIVERY\_TYPES
- V\_DENIAL\_REASONS
- V\_DISQUAL\_REASONS
- V\_FINDING\_CODES
- V\_FSP\_REGIONAL\_OFFICES
- V\_FSP\_VIOLATION\_CODES
- V\_LEGAL\_FIRMS
- V\_LEGAL\_REPRESENTATIVES
- V\_MILESTONE\_TYPES
- V\_OWNER\_TYPES
- V\_PEER\_GROUPS

- V\_RISK\_LEVELS
- V\_SANCTION\_TYPES
- V\_STATUSES
- V\_SUSPENSION\_REASONS
- V\_VIOLATION\_ACTION
- V\_WHOLESALERS
- V\_WIC\_CODES
- V\_OWNERS
- V\_VENDORS
- O\_STAFF\_MEMBERS
- O\_ORGANIZATIONAL\_UNITS
- O\_ANNUAL\_WIC\_COST\_SUMMARIES
- F\_ANNUAL\_FACTORS
- F\_BUDGETS
- F\_CASE\_ASSIGNMENTS
- F\_CASELOADS
- F\_CASELOAD\_RESTRICTIONS
- F\_MFR\_CONTACTS
- F\_MANUFACTURERS
- I\_FI\_INVENTORIES
- I\_STOCK\_INVENTORIES
- C\_RESOLVED\_CLIENTS
- C\_RESOLUTIONS
- I\_FI\_REJECT\_REASONS
- S\_PRINTER\_ADDRESSES
- DEL\_STATEMENTS

#### 1.34.5 EA3\_TAB\_UP.SQL

##### *Overview*

This script performs updates to the following tables:

- EOD\_CONTROLS
- O\_STAFF\_MEMBERS
- O\_STAFF\_TIME\_STUDIES
- I\_FOOD\_INSTRUMENT\_TYPES
- O\_ORGANIZATIONAL\_UNITS
- I\_FOOD\_INSTRUMENT\_FOODS
- O\_ORG\_PROGRAMS
- I\_FOOD\_INSTRUMENTS
- O\_ORG\_UNIT\_PHONES
- I\_FI\_FORMULAS
- O\_OUTREACH\_ORGANIZATIONS
- O\_OUT\_ORG\_LAS
- I\_FOOD\_PACKAGE\_FI\_TYPES
- O\_OUTREACH\_COMMS
- O\_OUTREACH\_ORG\_PHONES
- O\_OUTREACH\_PROGRAMS
- O\_STAFF\_PHONES
- O\_JOB\_DESCRIPTIONS

- C\_FAMILY\_ECONOMIC\_UNITS
- C\_FAMILY\_PHONES
- C\_FAM\_REFERRALS
- C\_FEU\_COMMUNICATIONS
- C\_SMOKING\_HISTORIES
- C\_PREV\_FAMILIES
- C\_CLIENTS
- C\_CERT\_PEER\_COUNSELS
- C\_CONTACTS
- C\_PEER\_RBFENDS
- C\_P\_COUNS\_NOTES
- C\_CLIENT\_COMMUNICATIONS
- C\_CLIENT\_GOALS
- C\_CLIENT\_NOTES
- C\_CLIENT\_PROGS
- C\_CLIENT\_REFERRALS
- C\_CLIENT\_SERVICES
- C\_CLIENT\_NE\_TOPICS
- C\_CLIENT\_SVC\_NE\_MATERIALS
- C\_MORE\_ETHNIC\_GROUPS
- C\_PREV\_NAMES
- C\_BF\_PROMO\_ISSUANCES
- C\_ALLERGY\_FOODS
- C\_W\_HEALTHS
- C\_W\_HH\_REASONS\_BFENDS
- C\_PS\_RESPONSES
- C\_CLIENT\_BP\_ISSUANCES
- C\_INCOME\_HISTORIES
- C\_ECONOMIC\_UNIT\_MEMBERS
- C\_INCOMES
- C\_INFANT\_DATA
- C\_RESOLUTIONS
- C\_RESOLVED\_CLIENTS
- C\_TRANSFERS\_INFO
- C\_TRANSFER\_HISTORIES
- C\_CERTIFICATIONS
- C\_CERT\_NUTR\_EDS
- C\_FOOD\_BOX\_DISTS
- C\_BLOODWORK\_DATA
- C\_CERT\_TERM\_REASONS
- C\_DIETARY\_ASSESSMENTS
- C\_DIET\_NUTRIENTS
- C\_HEALTH\_RISK\_FACTORS
- C\_INFANT\_CHILD\_MEDICALS
- C\_WOMAN\_MEDICALS
- C\_I\_C\_HEALTHS
- C\_I\_C\_HH\_REASONS\_BFENDS
- F\_CASE\_ASSIGNMENTS

- F\_WAIT\_LISTS
- F\_WAIT\_LIST\_CONTACTS
- F\_CASELOAD\_RESTRICTIONS
- I\_FOOD\_PACKAGES
- I\_FOOD\_PACKAGE\_FOODS
- I\_PACKAGE\_DISTRIBUTIONS
- C\_FOOD\_PACKAGE\_PRESCRIPTIONS
- I\_BF\_PROMO\_VCHRS
- I\_BF\_VCHR\_ITEMS
- I\_FI\_INVENTORIES
- I\_INVENTORIES
- I\_STOCK\_INVENTORIES
- I\_INV\_ORG\_UNITS
- V\_COMPLAINTS
- V\_MON\_ACTIVITIES
- E\_V\_ACTIVITY\_FINDINGS
- I\_BATCH\_RUNS
- S\_PRINTER\_ADDRESSES
- DEL\_STATEMENTS

### **1.35 Clinic serial number replenish**

#### 1.35.1 EC4\_CL\_SER\_REP.SQL

##### **Overview**

During the end of day process the system automatically replenishes serial numbers for the clinics with the number of serial numbers at 10% or less of the number defined by the static factors, “small clinic FI’s”, “med clinic FI’s”, and “large clinic FI’s”.

#### 1.35.2 CR\_DUAL\_ENROLLMENT.REP

##### **Overview**

This report file prints the dual enrollment report based upon updated c\_state\_clients data coming down from central.

*Arizona WIC is not using this script.*

### **1.36 Statewide summary of participants for dual enrollment check**

#### 1.36.1 EA3\_PART\_DUAL\_CHK.SQL

##### **Overview**

The system also generates a Notice to Reapply for CSF Participants six weeks prior to the end of their certification period if they remain potentially categorically eligible. A notice of Ineligibility is generated six weeks prior to the end of the certification period of Participants that will no longer be categorically eligible for CSF.

*Arizona WIC is not using this script.*

#### 1.36.2 CR\_DUAL\_ENROLLMENT.REP

##### **Overview**

This report file prints the dual enrollment report based upon update c\_state\_clients data coming down from central.

*Arizona WIC is not using this script.*

### **1.37 Update caseload assignment information from the clinics**

#### 1.37.1 EA3\_SYNC\_LA\_ASSIGNMENT.SQL

##### **Overview**

This script is used at local agencies for end of day processing to synchronize the caseload assignment data that has been re-allocated by the state. This script needs to be executed before the end of day administrative tables are deleted, as it uses the end of day administrative f\_case\_assignment table to insert and or update data into f\_case\_assignments at the local agency.

### **1.38 Update records at losing agencies for transferred clients and delete any irrelevant transfer information**

#### 1.38.1 EA3\_DEL\_IRR\_TRANSFER\_INFO.SQL

\*Part of the Hawaii Transfer System, will not be used in the Arizona program since transfers are immediate.

##### **Overview**

This script is used at local agencies for end of day processing and purges irrelevant data which came from state. It also deletes client information which belongs to some other LA from CTI tables. This script needs to be executed after end of day data from the central site is loaded into the actual tables from end of day administration tables.

*Arizona WIC doesn't use this script.*

#### 1.38.2 EA3\_TERM\_TR\_CLIENTS.SQL

\*Part of the Hawaii Transfer System, will not be used in the Arizona program since transfers are immediate.

##### **Overview**

After end of day data is received from state and before it is loaded from end of day administration tables to the swic tables, this script is used to terminate clients who have been transferred to some other local agency (*doesn't apply to clients from checked out clinic*).

*Arizona WIC doesn't use this script.*

### **1.39 Send redemption/rejection information to agencies**

#### 1.39.1 EA3\_PROC\_FI.SQL

##### **Overview**

Process the Food instrument's received from Central. Update the food instrument data of the local agency with the food instrument's received from the local agency.

## **1.40 Transmit archival clients and vendors and update records at agencies**

### **1.40.1 EA3\_PRG\_ARCHV.SQL**

#### **Overview**

Purge archived client records marked with an archive flag of 'Y'. The deletes have been ordered from the smallest level child up to the parent (c\_clients). Archived vendors will be purged via the delete triggers.

### **1.40.2 EA3\_RETRIEVE.SQL**

#### **Overview**

Run scripts EA3\_INS\_RETR\_CLI.SQL and EA3\_INS\_RETR\_VEN.SQL to retrieve clients who were archived and then requested to be put back on the system.

### **1.40.3 EA3\_INS\_RETR\_CLI.SQL**

#### **Overview**

This script performs inserts from the temporary tables into all client tables.

### **1.40.4 EA3\_INS\_RETR\_VEN.SQL**

#### **Overview**

This script performs inserts from the temporary tables in to all vendor tables.

## **1.41 Vendor monthly reports**

#### **Overview**

This produces Vendor reports that run monthly, listed by DFDD section number. The monthly reports are:

- Section 7 – 3.1.5: Compliance Cases and Sanctions Summary
- Section 7 – 3.1.6: Monitoring and Sanctions History
- Section 7 – 3.4.11: Vendor Alpha within Agency (AZW168)
- Section 7 – 3.4.12: Vendor Application Data
- Section 7 – 3.4.13: Vendor Authorization
- Section 7 – 3.4.14: Vendor ID within State (AZW165)
- Section 7 – 3.5.1: FI Type File Listing (AZW981)
- Section 7 – 3.5.2: FI Type Price Update (AZW305)
- Section 7 – 3.5.3: FI's Paid/Voided Index (AZW370)
- Section 7 – 3.5.4: FI's Paid/Voided by Type Index (AZW370x)
- Section 7 – 3.5.5: Flagged FI's (AZW320)
- Section 7 – 3.5.6: Redemption Error (AZW702)
- Section 7 – 3.5.7: Vendor Food Cost (AZW310)
- Section 7 – 3.5.8: Vendor Food Cost – Issuing Agency Analysis (AZW315)
- Section 7 – 3.5.10: Vendor Reject (AZW340)
- Section 7 – 3.5.11: Vendor Report Card (AZW330)

*Arizona WIC doesn't run these reports in EOD.*

## **1.42 Vendor quarterly reports**

#### **Overview**

This produces Vendor reports that run quarterly, listed by DFDD section number. The quarterly reports are:

Section 7 – 3.1.9: Vendor Quarterly Profiles

Section 7 – 3.4.4: High Risk Vendor (AZW350, AZW355)

Section 7 – 3.5.9: Vendor Formula Redemption (AZW388)

*Arizona WIC doesn't run these reports in EOD.*

### **1.43 Print out status logs of the process**

#### **1.43.1 EA\_PRINT.BAT**

##### ***Overview***

This batch file produces print outs by agency for the following log files:

- Insert
- Export
- Truncate
- Import
- Delete
- Update

### **1.44 Update State clients for failed EOD agencies/ Sync c\_state\_clients**

#### **1.44.1 EC2\_NOEOD.SQL**

##### ***Overview***

This script will insert record for any missing eod run agency. So that it can take care of c\_state\_clients record for that agency on next successful EOD run.

#### **1.44.2 EC2\_UPD\_E\_C\_CLIENTS.SQL**

This script will update last\_first\_bdate column in e\_c\_clients table.

#### **1.44.3 EC4\_CSC\_UPDATE.SQL**

This script will save all the c\_state\_clients records for missing EOD agency.

#### **1.44.4 EC4\_SYNC\_STATE\_CLIENTS.SQL**

This script will keep all the c\_state\_clients records in sync.

### **1.45 Identify Duplicate FI Range**

#### **1.45.1 EC2\_DUPLICATE\_FI\_RANGE.SQL**

This script checks for any FI ranges that have been duplicated over two agencies.

### **1.46 Bring up missing issuance FI's from agencies**

#### **1.46.1 EC2\_FAILED\_FIS.SQL**

This script populates the e\_failed\_fis table with any new failed fi's from the agencies.

#### 1.46.2 EC4\_GET\_UPDATES.SQL

This script generates an update script for each agency. Update statements are based on any failed FI records existing in the e\_failed\_fis table for the current agency. The script that results from this process is later copied by each LA during their EA3 process and is executed at that agency during EA3.

## 2 Appendix A

### Appendix A - End of Day Report Definitions and Layouts

The system generates several reports and logs during and after the End of Day Process. Some are used for participant communications, others are used to track the successful completion of the End of Day process. The following is a list of the logs and outputs that are generated at the Central and Local Agencies from the End of Day process:

#### Central

**CTRL\_SQL.LOG** - A list that details by date the participants at each agency whose records were added, deleted or updated. This includes information regarding the processing of food instruments. This log shows information for each of these EOD scripts:

*EC1\_OTHER\_PURG.sql*, an example of the content of this script's file is as follows:

```
RUNNING EC1_OTHER_PURG.SQL
PL/SQL procedure successfully completed.
FINISHED EC1_OTHER_PURG.SQL
```

*EC1\_PRG\_ARCHV.sql*, an example of the content of this script's file is as follows:

```
RUNNING EC1_PRG_ARCHV.SQL
PL/SQL procedure successfully completed.
FINISHED EC1_PRG_ARCHV.SQL
```

*EC1\_CTRL\_CHK.sql*, an example of the content of this script's file is as follows:

```
RUNNING EC1_CTRL_CHK.SQL
Input truncated to 1 characters
PL/SQL procedure successfully completed.
FINISHED EC1_CTRL_CHK.SQL
```

```
EC1_STALE_DATE_FI.SQL
RUNNING EC1_STALE_DATE_FI.SQL
PL/SQL procedure successfully completed.
FINISHED EC1_STALE_DATE_FI.SQL
```

*EC2\_CL\_REC\_PROCESS.sql*, an example of the content of this script's file is as follows:

Input truncated to 1 characters  
 Start Time :24-Oct-2003 : 07:40:42 PM  
 Running caseload\_population.new\_upd\_cert\_fctd for :1/2004  
 start date 01-OCT-03 end date 31-OCT-03  
 Total caseload count :28  
 Time :24-Oct-2003 : 07:44:04 PM  
 Running caseload\_population.void\_checks\_fctd for :11/2003  
 start date 01-AUG-03 end date 31-AUG-03  
 Total Count :97 Last\_upd\_date :23-OCT-2003:19:32:22  
 Time :24-Oct-2003 : 07:44:04 PM  
 Running caseload\_population.void\_checks\_fctd for :12/2003  
 start date 01-SEP-03 end date 30-SEP-03  
 Total Count :97 Last\_upd\_date :23-OCT-2003:19:32:22  
 Time :24-Oct-2003 : 07:44:04 PM  
 Running caseload\_population.void\_checks\_fctd for :1/2004  
 start date 01-OCT-03 end date 31-OCT-03  
 Total Count :97 Last\_upd\_date :23-OCT-2003:19:32:22  
 Time :24-Oct-2003 : 07:44:04 PM  
 Running caseload\_population.late\_pickup\_checks\_fctd for :12/2003  
 start date 01-SEP-03 end date 30-SEP-03  
 Total Count :97 Last\_upd\_date :23-OCT-2003:19:32:22  
 Time :24-Oct-2003 : 07:44:04 PM  
 Running caseload\_population.late\_pickup\_checks\_fctd for :1/2004  
 start date 01-OCT-03 end date 31-OCT-03  
 Total Count :97 Last\_upd\_date :23-OCT-2003:19:32:22  
 End Time :24-Oct-2003 : 07:44:04 PM  
 FINISHED ec2\_cl\_rec\_process.sql

*EC2\_DEL\_CL\_FUT\_RECS.sql*, an example of the content of this script's file is as follows:

```

START_TIME -----
----- 09/16/1999 08:00 PM
RUNNING EC2_DEL_CL_FUT_RECS.SQL
Input truncated to 1 characters
FINISHED EC2_DEL_CL_FUT_RECS.SQL
  
```

*EC2\_DUAL\_ENROLL.sql*, an example of the content of this script's file is as follows:

```

RUNNING EC2_DUAL_ENROLL.SQL
FINISHED EC2_DUAL_ENROLL.SQL
  
```

*EC2\_SYNC\_CLINIC\_ASSIGNMENT.sql*, an example of the content of this script's file is as follows:

```

EC2_SYNC_CLINIC_ASSIGNMENT Processing
UPDATING EOD_BY
1 row updated.
Commit complete.
UPDATE FINISHED
RUNNING EC2_SYNC_CLINIC_ASSIGNMENT.SQL
  
```

```

START_TIME
-----
09/16/1999 08:15 PM
Block processing started...
executed successfully...
FINISHED EC2_SYNC_CLINIC_ASSIGNMENT.SQL
ec2_sync_clinic_assignment
PL/SQL procedure successfully completed.

```

*EC2\_DUPLICATE\_FI\_RANGE.sql*, an example of the content of this script's file is as follows

```

RUNNING EC2_DUPLICATE_FI_RANGE.SQL
START_TIME
-----
10/24/2003 08:48 PM

PL/SQL procedure successfully completed.
FINISHED EC2_DUPLICATE_FI_RANGE.SQL

```

*EC2\_FOOD\_UP.sql*, an example of the content of this script's file is as follows:

```

EC2_FOOD_UP Processing
RUNNING EC2_FOOD_UP.SQL

```

```

START_TIME
-----
09/16/1999 08:15 PM
PL/SQL procedure successfully completed.
Table altered.

PL/SQL procedure successfully completed.
Table altered.

FINISHED EC2_FOOD_UP.SQL

```

*EC2\_FI\_UPD.sql*, an example of the content of this script's file is as follows:

```

EC2_FI_UPD Processing
RUNNING EC2_FI_UPD.SQL

```

```

START_TIME
-----
09/16/1999 08:19 PM
TOTAL NO. OF PROCESSED FOOD INSTRUMENTS : 16299

```

```

PL/SQL procedure successfully completed.

```

```

FINISHED EC2_FI_UPD.SQL

```

*EC2\_FAILED\_FIS.sql*, an example of the content of this script's file is as follows:

```

EC2_FAILED_FIS Processing
RUNNING EC2_FAILED_FIS.SQL

```

```

START_TIME

```

-----  
10/24/2003 09:15 PM

Following is information regarding FIs that failed to insert at Central usually because of missing client food package records (cfpp). Here is an outline of what is displayed below:

- 1) FIs that failed on a previous day, but were recovered today.
- 2) Recovered FIs that have been deleted from the e\_failed\_fis table.
- 3) FIs that failed today and were inserted into the e\_failed\_fis table.
- 4) Count of all failed FIs by agency.
- 5) List of all failed FIs ordered by agency.

\*\*\*\*\*

Recovered Client FIs

FIS\_RECOVERED

-----

0

Recovered Test-Print and Voided FIs

FIS\_RECOVERED

-----

0

Recovered by Agency Count

no rows selected

\*\*\*\*\*

\*\*\*\*\*

Deleting Client FI's from Failed FI List

0 rows deleted.

Deleting Test Prints and Voids from Failed FI List

0 rows deleted.

Commit complete.

\*\*\*\*\*

\*\*\*\*\*

Inserting failed FI information into Failed FI Table.

0 rows created.

Updating revisited FIs to Failed FIs list

0 rows updated.

Commit complete.

\*\*\*\*\*

\*\*\*\*\*

Final List of Failed FI count

no rows selected

\*\*\*\*\*

\*\*\*\*\*

List of all Failed FIs

no rows selected

FINISHED EC2\_FAILED\_FIS.SQL

*EC2\_ISSUE\_FSMC.sql*, an example of the content of this script's file is as follows:

*RUNNING EC2\_ISSUE\_FSMC.SQL*

*START\_TIME*

-----  
*10/24/2003 09:15 PM*

*Terminate spooling to EC2\_ISSUE\_FSMC.LOG because EC2\_ISSUE\_FSMC.SQL will spool to issuean.dat*

*UPDATING EOD\_BY*

*1 row updated.*

*Commit complete.*

*EC4\_BNK\_POST.sql*, an example of the content of this script's file is as follows:

*START\_TIME* -----  
----- 09/16/1999 08:34 PM  
*RUNNING EC4\_BNK\_POST.SQL*  
  
*FINISHED EC4\_BNK\_POST.SQL*

*EC4\_CASHFLOW\_UPD.sql*, an example of the content of this script's file is as follows:

```

START_TIME -----
----- 09/16/1999 08:39 PM
RUNNING EC4_CASHFLOW_UPD.SQL
Input truncated to 1 characters
Processing started...
Updated calculated values ...
Updated calculated values ...
Processing outlays...
PL/SQL procedure successfully completed.

Processing obligations...
Updated calculated values ...
Processing outlays...done...

```

*EC4\_MON\_END.sql*, an example of the content of this script's file is as follows:

```

START_TIME -----
----- 09/16/1999 08:42 PM
RUNNING EC4_MON_END.SQL
begin.
PL/SQL procedure successfully completed.

ipl_date opened.

FINISHED EC4_MON_END.SQL

```

*EC4\_VEN\_RISK\_HISTORY.sql*, an example of the content of this script's file is as follows:

```

RUNNING EC4_VEN_RISK_HISTORY.SQL

PL/SQL procedure successfully completed.

FINISHED EC4_VEN_RISK_HISTORY.SQL

```

*EC4\_VEN\_PEER\_FI.sql*, an example of the content of this script's file is as follows:

```

RUNNING ec4_ven_peer_fi.sql
No Month-End processing due at this time.
Peer Group Average update will not occur.

PL/SQL procedure successfully completed.

FINISHED ec4_ven_peer_fi.sql

```

*EC4\_VEN\_LOC\_AGCY\_FI.sql*, an example of the content of this script's file is as follows:

```

RUNNING ec4_ven_loc_agcy_fi.sql
Starting ec4_ven_loc_agcy_fi.sql, LA FI Averages update

PL/SQL procedure successfully completed.

FINISHED ec4_ven_loc_agcy_fi.sql

```

*EC4\_VEN\_PRICE.sql*, an example of the content of this script's file is as follows:

RUNNING ec4\_ven\_price.sql

Table truncated.

Table truncated.

Starting Vendor Ranking EOD

Inserting/Updating V\_Fi\_Prices information for the following Vendor ID / FI Type Combinations:

Vendor Ranking Successfully Completed

PL/SQL procedure successfully completed.

FINISHED ec4\_ven\_price.sql

*EC4\_CL\_SER\_REP.sql*, an example of the content of this script's file is as follows:

RUNNING ec4\_cl\_ser\_rep.sql

Input truncated to 1 characters

Inserted new fi inventory record for MEDIUM Clinic seq\_id 77

Inserted new fi inventory record for SMALL Clinic seq\_id 88

Inserted new fi inventory record for SMALL Clinic seq\_id 144

Inserted new fi inventory record for MEDIUM Clinic seq\_id 153

PL/SQL procedure successfully completed.

FINISHED ec4\_cl\_ser\_rep.sql

*EC4\_SYNC\_STATE\_CLIENTS.sql*, an example of the content of this script's file is as follows:

RUNNING ec4\_sync\_state\_clients.sql

Input truncated to 1 characters

PL/SQL procedure successfully completed.

FINISHED ec4\_sync\_state\_clients.sql

**SQL.LOG** - A historic list that details by date the participants at each agency whose records were added, deleted or updated. This log follows the same format as **Ctrl\_sql.log** but displays a history instead of only the current day.

**EC.LOG** - A list that details by date and time that all Central Agency scripts were initiated and finished. An example of the content of this file is as follows:

10-24-2003 19:30:12 - EC1 STARTED

10-24-2003 19:30:14 - EC1.BAT EOD ALREADY RUN CHECK STARTED  
10-24-2003 19:30:14 - EC1.BAT EOD ALREADY RUN CHECK FINISHED  
10-24-2003 19:30:19 - EC1.BAT AIM APPLICATION FILE UPDATE STARTED  
10-24-2003 19:30:19 - EC1.BAT AIM APPLICATION FILE UPDATE FINISHED  
10-24-2003 19:30:39 - EC1.BAT SQL STARTED  
10-24-2003 19:37:35 - BUILDING ANPGA.DAT  
10-24-2003 19:37:45 - BUILDING ANVENDOR.DAT  
10-24-2003 19:38:09 - EC1.BAT SQL FINISHED  
10-24-2003 19:38:19 - STARTED EC2.BAT  
10-24-2003 19:38:24 - EC2.BAT CHECKING AGENCY 01  
10-24-2003 19:38:24 - EC2.BAT AGENCY 01 FOUND  
10-24-2003 19:38:40 - EC2.BAT AGENCY 01 IMPORTED  
10-24-2003 19:42:33 - EC2.BAT AGENCY 01 LOADED  
10-24-2003 19:42:43 - EC2.BAT CHECKING AGENCY 02  
10-24-2003 19:42:43 - EC2.BAT AGENCY 02 FOUND  
10-24-2003 19:42:58 - EC2.BAT AGENCY 02 IMPORTED  
10-24-2003 19:44:15 - EC2.BAT AGENCY 02 LOADED  
10-24-2003 19:44:25 - EC2.BAT CHECKING AGENCY 03  
10-24-2003 19:44:25 - EC2.BAT AGENCY 03 FOUND  
10-24-2003 19:44:42 - EC2.BAT AGENCY 03 IMPORTED  
10-24-2003 19:45:45 - EC2.BAT AGENCY 03 LOADED  
10-24-2003 19:45:55 - EC2.BAT CHECKING AGENCY 04  
10-24-2003 19:45:55 - EC2.BAT AGENCY 04 FOUND  
10-24-2003 19:46:11 - EC2.BAT AGENCY 04 IMPORTED  
10-24-2003 19:47:09 - EC2.BAT AGENCY 04 LOADED  
10-24-2003 19:47:19 - EC2.BAT CHECKING AGENCY 05  
10-24-2003 19:47:19 - EC2.BAT AGENCY 05 FOUND  
10-24-2003 19:47:34 - EC2.BAT AGENCY 05 IMPORTED  
10-24-2003 19:48:37 - EC2.BAT AGENCY 05 LOADED  
10-24-2003 19:48:47 - EC2.BAT CHECKING AGENCY 06  
10-24-2003 19:48:47 - EC2.BAT AGENCY 06 FOUND  
10-24-2003 19:49:07 - EC2.BAT AGENCY 06 IMPORTED  
10-24-2003 19:49:55 - EC2.BAT AGENCY 06 LOADED  
10-24-2003 19:50:05 - EC2.BAT CHECKING AGENCY 07  
10-24-2003 19:50:05 - EC2.BAT AGENCY 07 FOUND  
10-24-2003 19:50:34 - EC2.BAT AGENCY 07 IMPORTED  
10-24-2003 20:10:54 - EC2.BAT AGENCY 07 LOADED  
10-24-2003 20:11:05 - EC2.BAT CHECKING AGENCY 08  
10-24-2003 20:11:05 - EC2.BAT AGENCY 08 FOUND  
10-24-2003 20:11:21 - EC2.BAT AGENCY 08 IMPORTED  
10-24-2003 20:12:41 - EC2.BAT AGENCY 08 LOADED  
10-24-2003 20:12:51 - EC2.BAT CHECKING AGENCY 09  
10-24-2003 20:12:51 - EC2.BAT AGENCY 09 FOUND  
10-24-2003 20:13:13 - EC2.BAT AGENCY 09 IMPORTED  
10-24-2003 20:14:13 - EC2.BAT AGENCY 09 LOADED  
10-24-2003 20:14:23 - EC2.BAT CHECKING AGENCY 10  
10-24-2003 20:14:23 - EC2.BAT AGENCY 10 FOUND  
10-24-2003 20:14:45 - EC2.BAT AGENCY 10 IMPORTED  
10-24-2003 20:21:08 - EC2.BAT AGENCY 10 LOADED  
10-24-2003 20:21:19 - EC2.BAT CHECKING AGENCY 11  
10-24-2003 20:21:19 - EC2.BAT AGENCY 11 FOUND

10-24-2003 20:21:43 - EC2.BAT AGENCY 11 IMPORTED  
10-24-2003 20:25:09 - EC2.BAT AGENCY 11 LOADED  
10-24-2003 20:25:19 - EC2.BAT CHECKING AGENCY 13  
10-24-2003 20:25:19 - EC2.BAT AGENCY 13 FOUND  
10-24-2003 20:25:38 - EC2.BAT AGENCY 13 IMPORTED  
10-24-2003 20:27:41 - EC2.BAT AGENCY 13 LOADED  
10-24-2003 20:27:51 - EC2.BAT CHECKING AGENCY 14  
10-24-2003 20:27:51 - EC2.BAT AGENCY 14 FOUND  
10-24-2003 20:28:10 - EC2.BAT AGENCY 14 IMPORTED  
10-24-2003 20:31:39 - EC2.BAT AGENCY 14 LOADED  
10-24-2003 20:31:49 - EC2.BAT CHECKING AGENCY 15  
10-24-2003 20:31:49 - EC2.BAT AGENCY 15 FOUND  
10-24-2003 20:32:12 - EC2.BAT AGENCY 15 IMPORTED  
10-24-2003 20:33:04 - EC2.BAT AGENCY 15 LOADED  
10-24-2003 20:33:14 - EC2.BAT CHECKING AGENCY 17  
10-24-2003 20:33:14 - EC2.BAT AGENCY 17 FOUND  
10-24-2003 20:33:32 - EC2.BAT AGENCY 17 IMPORTED  
10-24-2003 20:34:42 - EC2.BAT AGENCY 17 LOADED  
10-24-2003 20:34:52 - EC2.BAT CHECKING AGENCY 25  
10-24-2003 20:34:52 - EC2.BAT AGENCY 25 FOUND  
10-24-2003 20:35:16 - EC2.BAT AGENCY 25 IMPORTED  
10-24-2003 20:36:05 - EC2.BAT AGENCY 25 LOADED  
10-24-2003 20:36:15 - EC2.BAT CHECKING AGENCY 27  
10-24-2003 20:36:15 - EC2.BAT AGENCY 27 FOUND  
10-24-2003 20:36:36 - EC2.BAT AGENCY 27 IMPORTED  
10-24-2003 20:38:06 - EC2.BAT AGENCY 27 LOADED  
10-24-2003 20:38:16 - EC2.BAT CHECKING AGENCY 28  
10-24-2003 20:38:16 - EC2.BAT AGENCY 28 FOUND  
10-24-2003 20:38:35 - EC2.BAT AGENCY 28 IMPORTED  
10-24-2003 20:41:16 - EC2.BAT AGENCY 28 LOADED  
10-24-2003 20:41:26 - EC2.BAT CHECKING AGENCY 29  
10-24-2003 20:41:26 - EC2.BAT AGENCY 29 FOUND  
10-24-2003 20:41:45 - EC2.BAT AGENCY 29 IMPORTED  
10-24-2003 20:43:23 - EC2.BAT AGENCY 29 LOADED  
10-24-2003 20:43:33 - EC2.BAT CHECKING AGENCY 30  
10-24-2003 20:43:33 - EC2.BAT AGENCY 30 FOUND  
10-24-2003 20:43:53 - EC2.BAT AGENCY 30 IMPORTED  
10-24-2003 20:45:45 - EC2.BAT AGENCY 30 LOADED  
10-24-2003 20:45:55 - EC2.BAT CHECKING AGENCY 32  
10-24-2003 20:45:55 - EC2.BAT AGENCY 32 FOUND  
10-24-2003 20:46:13 - EC2.BAT AGENCY 32 IMPORTED  
10-24-2003 20:47:08 - EC2.BAT AGENCY 32 LOADED  
10-24-2003 20:47:18 - EC2.BAT EC2.SQL STARTED  
10-24-2003 21:13:45 - EC2.BAT EC2.SQL FINISHED  
10-24-2003 21:13:55 - EC2.BAT FINISHED  
10-24-2003 21:13:55 - EC4.BAT STARTED  
10-24-2003 21:14:57 - EC4.BAT FTPTRANS.EXE STARTED  
10-24-2003 21:17:36 - EC4.BAT FTPTRANS.EXE FINISHED  
10-24-2003 21:21:46 - EC4.BAT SQLOAD OF FSMC STARTED  
10-24-2003 21:22:13 - EC4.BAT SQLOAD OF FSMC FINISHED  
10-24-2003 21:22:23 - EC4.BAT BANKSQL STARTED

10-24-2003 21:33:40 - EC4.BAT BANKSQL FINISHED  
10-24-2003 21:33:50 - EC4.BAT EC4.SQL STARTED  
10-24-2003 21:54:20 - EC4.BAT EC4.SQL FINISHED  
10-24-2003 21:54:30 - EC4.BAT EC4\_CSC.SQL STARTED  
10-24-2003 21:54:35 - EC4.BAT EC4\_CSC.SQL FINISHED  
10-24-2003 21:56:14 - EC4.BAT AGENCY 01 IMPORTED  
10-24-2003 21:56:24 - EC4.BAT AGENCY 01 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 21:56:24 - EC4.BAT AGENCY 01 STARTED  
10-24-2003 21:58:17 - EC4.BAT AGENCY 01 STATE CLIENTS INSERT STARTED  
10-24-2003 21:58:28 - EC4.BAT AGENCY 01 INSERTED  
10-24-2003 21:58:28 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 21:58:38 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 21:59:09 - EC4.BAT AGENCY 01 EXPORTED  
10-24-2003 21:59:19 - EC4.BAT AGENCY 01 DONE  
10-24-2003 21:59:19 - EC4.BAT AGENCY 01 EC4\_GET\_UPDATES STARTED  
10-24-2003 21:59:20 - EC4.BAT AGENCY 01 EC4\_GET\_UPDATES FINISHED  
10-24-2003 21:59:20 - EC4.BAT AGENCY 01 ZIP STARTED  
10-24-2003 21:59:22 - EC4.BAT AGENCY 01 ZIP FINISHED  
10-24-2003 21:59:24 - EC4.BAT AGENCY 02 IMPORTED  
10-24-2003 21:59:24 - EC4.BAT AGENCY 02 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 21:59:24 - EC4.BAT AGENCY 02 STARTED  
10-24-2003 22:01:07 - EC4.BAT AGENCY 02 STATE CLIENTS INSERT STARTED  
10-24-2003 22:01:17 - EC4.BAT AGENCY 02 INSERTED  
10-24-2003 22:01:18 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:01:27 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:01:46 - EC4.BAT AGENCY 02 EXPORTED  
10-24-2003 22:01:56 - EC4.BAT AGENCY 02 DONE  
10-24-2003 22:01:56 - EC4.BAT AGENCY 02 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:01:57 - EC4.BAT AGENCY 02 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:01:57 - EC4.BAT AGENCY 02 ZIP STARTED  
10-24-2003 22:01:59 - EC4.BAT AGENCY 02 ZIP FINISHED  
10-24-2003 22:02:01 - EC4.BAT AGENCY 03 IMPORTED  
10-24-2003 22:02:01 - EC4.BAT AGENCY 03 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:02:02 - EC4.BAT AGENCY 03 STARTED  
10-24-2003 22:03:46 - EC4.BAT AGENCY 03 STATE CLIENTS INSERT STARTED  
10-24-2003 22:03:57 - EC4.BAT AGENCY 03 INSERTED  
10-24-2003 22:03:57 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:04:06 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:04:35 - EC4.BAT AGENCY 03 EXPORTED  
10-24-2003 22:04:46 - EC4.BAT AGENCY 03 DONE  
10-24-2003 22:04:46 - EC4.BAT AGENCY 03 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:04:46 - EC4.BAT AGENCY 03 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:04:46 - EC4.BAT AGENCY 03 ZIP STARTED  
10-24-2003 22:04:49 - EC4.BAT AGENCY 03 ZIP FINISHED  
10-24-2003 22:04:50 - EC4.BAT AGENCY 04 IMPORTED  
10-24-2003 22:04:50 - EC4.BAT AGENCY 04 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:04:51 - EC4.BAT AGENCY 04 STARTED

10-24-2003 22:06:36 - EC4.BAT AGENCY 04 STATE CLIENTS INSERT STARTED  
10-24-2003 22:06:47 - EC4.BAT AGENCY 04 INSERTED  
10-24-2003 22:06:47 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:06:55 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:07:14 - EC4.BAT AGENCY 04 EXPORTED  
10-24-2003 22:07:24 - EC4.BAT AGENCY 04 DONE  
10-24-2003 22:07:24 - EC4.BAT AGENCY 04 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:07:24 - EC4.BAT AGENCY 04 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:07:25 - EC4.BAT AGENCY 04 ZIP STARTED  
10-24-2003 22:07:27 - EC4.BAT AGENCY 04 ZIP FINISHED  
10-24-2003 22:07:28 - EC4.BAT AGENCY 05 IMPORTED  
10-24-2003 22:07:28 - EC4.BAT AGENCY 05 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:07:29 - EC4.BAT AGENCY 05 STARTED  
10-24-2003 22:09:15 - EC4.BAT AGENCY 05 STATE CLIENTS INSERT STARTED  
10-24-2003 22:09:25 - EC4.BAT AGENCY 05 INSERTED  
10-24-2003 22:09:25 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:09:34 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:09:53 - EC4.BAT AGENCY 05 EXPORTED  
10-24-2003 22:10:03 - EC4.BAT AGENCY 05 DONE  
10-24-2003 22:10:03 - EC4.BAT AGENCY 05 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:10:04 - EC4.BAT AGENCY 05 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:10:04 - EC4.BAT AGENCY 05 ZIP STARTED  
10-24-2003 22:10:06 - EC4.BAT AGENCY 05 ZIP FINISHED  
10-24-2003 22:10:08 - EC4.BAT AGENCY 06 IMPORTED  
10-24-2003 22:10:08 - EC4.BAT AGENCY 06 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:10:09 - EC4.BAT AGENCY 06 STARTED  
10-24-2003 22:11:52 - EC4.BAT AGENCY 06 STATE CLIENTS INSERT STARTED  
10-24-2003 22:12:03 - EC4.BAT AGENCY 06 INSERTED  
10-24-2003 22:12:03 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:12:12 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:12:30 - EC4.BAT AGENCY 06 EXPORTED  
10-24-2003 22:12:40 - EC4.BAT AGENCY 06 DONE  
10-24-2003 22:12:40 - EC4.BAT AGENCY 06 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:12:41 - EC4.BAT AGENCY 06 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:12:41 - EC4.BAT AGENCY 06 ZIP STARTED  
10-24-2003 22:12:43 - EC4.BAT AGENCY 06 ZIP FINISHED  
10-24-2003 22:12:45 - EC4.BAT AGENCY 07 IMPORTED  
10-24-2003 22:12:45 - EC4.BAT AGENCY 07 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:12:45 - EC4.BAT AGENCY 07 STARTED  
10-24-2003 22:14:36 - EC4.BAT AGENCY 07 STATE CLIENTS INSERT STARTED  
10-24-2003 22:14:46 - EC4.BAT AGENCY 07 INSERTED  
10-24-2003 22:14:46 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:14:55 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:15:41 - EC4.BAT AGENCY 07 EXPORTED  
10-24-2003 22:15:51 - EC4.BAT AGENCY 07 DONE  
10-24-2003 22:15:51 - EC4.BAT AGENCY 07 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:15:52 - EC4.BAT AGENCY 07 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:15:52 - EC4.BAT AGENCY 07 ZIP STARTED

10-24-2003 22:15:57 - EC4.BAT AGENCY 07 ZIP FINISHED  
10-24-2003 22:16:00 - EC4.BAT AGENCY 08 IMPORTED  
10-24-2003 22:16:00 - EC4.BAT AGENCY 08 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:16:00 - EC4.BAT AGENCY 08 STARTED  
10-24-2003 22:17:49 - EC4.BAT AGENCY 08 STATE CLIENTS INSERT STARTED  
10-24-2003 22:18:00 - EC4.BAT AGENCY 08 INSERTED  
10-24-2003 22:18:00 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:18:09 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:18:29 - EC4.BAT AGENCY 08 EXPORTED  
10-24-2003 22:18:39 - EC4.BAT AGENCY 08 DONE  
10-24-2003 22:18:39 - EC4.BAT AGENCY 08 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:18:40 - EC4.BAT AGENCY 08 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:18:40 - EC4.BAT AGENCY 08 ZIP STARTED  
10-24-2003 22:18:42 - EC4.BAT AGENCY 08 ZIP FINISHED  
10-24-2003 22:18:44 - EC4.BAT AGENCY 09 IMPORTED  
10-24-2003 22:18:44 - EC4.BAT AGENCY 09 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:18:45 - EC4.BAT AGENCY 09 STARTED  
10-24-2003 22:20:28 - EC4.BAT AGENCY 09 STATE CLIENTS INSERT STARTED  
10-24-2003 22:20:39 - EC4.BAT AGENCY 09 INSERTED  
10-24-2003 22:20:39 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:20:48 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:21:06 - EC4.BAT AGENCY 09 EXPORTED  
10-24-2003 22:21:16 - EC4.BAT AGENCY 09 DONE  
10-24-2003 22:21:16 - EC4.BAT AGENCY 09 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:21:17 - EC4.BAT AGENCY 09 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:21:17 - EC4.BAT AGENCY 09 ZIP STARTED  
10-24-2003 22:21:19 - EC4.BAT AGENCY 09 ZIP FINISHED  
10-24-2003 22:21:21 - EC4.BAT AGENCY 10 IMPORTED  
10-24-2003 22:21:21 - EC4.BAT AGENCY 10 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:21:21 - EC4.BAT AGENCY 10 STARTED  
10-24-2003 22:23:04 - EC4.BAT AGENCY 10 STATE CLIENTS INSERT STARTED  
10-24-2003 22:23:15 - EC4.BAT AGENCY 10 INSERTED  
10-24-2003 22:23:15 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:23:25 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:24:03 - EC4.BAT AGENCY 10 EXPORTED  
10-24-2003 22:24:14 - EC4.BAT AGENCY 10 DONE  
10-24-2003 22:24:14 - EC4.BAT AGENCY 10 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:24:16 - EC4.BAT AGENCY 10 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:24:16 - EC4.BAT AGENCY 10 ZIP STARTED  
10-24-2003 22:24:19 - EC4.BAT AGENCY 10 ZIP FINISHED  
10-24-2003 22:24:21 - EC4.BAT AGENCY 11 IMPORTED  
10-24-2003 22:24:21 - EC4.BAT AGENCY 11 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:24:21 - EC4.BAT AGENCY 11 STARTED  
10-24-2003 22:26:04 - EC4.BAT AGENCY 11 STATE CLIENTS INSERT STARTED  
10-24-2003 22:26:15 - EC4.BAT AGENCY 11 INSERTED  
10-24-2003 22:26:15 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:26:23 - EC4.BAT EC4\_RETRIEVE FINISHED

10-24-2003 22:26:45 - EC4.BAT AGENCY 11 EXPORTED  
10-24-2003 22:26:56 - EC4.BAT AGENCY 11 DONE  
10-24-2003 22:26:56 - EC4.BAT AGENCY 11 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:26:56 - EC4.BAT AGENCY 11 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:26:56 - EC4.BAT AGENCY 11 ZIP STARTED  
10-24-2003 22:26:59 - EC4.BAT AGENCY 11 ZIP FINISHED  
10-24-2003 22:27:01 - EC4.BAT AGENCY 13 IMPORTED  
10-24-2003 22:27:01 - EC4.BAT AGENCY 13 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:27:02 - EC4.BAT AGENCY 13 STARTED  
10-24-2003 22:28:45 - EC4.BAT AGENCY 13 STATE CLIENTS INSERT STARTED  
10-24-2003 22:28:57 - EC4.BAT AGENCY 13 INSERTED  
10-24-2003 22:28:57 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:29:05 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:29:32 - EC4.BAT AGENCY 13 EXPORTED  
10-24-2003 22:29:42 - EC4.BAT AGENCY 13 DONE  
10-24-2003 22:29:42 - EC4.BAT AGENCY 13 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:29:43 - EC4.BAT AGENCY 13 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:29:43 - EC4.BAT AGENCY 13 ZIP STARTED  
10-24-2003 22:29:45 - EC4.BAT AGENCY 13 ZIP FINISHED  
10-24-2003 22:29:47 - EC4.BAT AGENCY 14 IMPORTED  
10-24-2003 22:29:47 - EC4.BAT AGENCY 14 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:29:48 - EC4.BAT AGENCY 14 STARTED  
10-24-2003 22:31:31 - EC4.BAT AGENCY 14 STATE CLIENTS INSERT STARTED  
10-24-2003 22:31:41 - EC4.BAT AGENCY 14 INSERTED  
10-24-2003 22:31:41 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:31:50 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:32:11 - EC4.BAT AGENCY 14 EXPORTED  
10-24-2003 22:32:21 - EC4.BAT AGENCY 14 DONE  
10-24-2003 22:32:21 - EC4.BAT AGENCY 14 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:32:22 - EC4.BAT AGENCY 14 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:32:22 - EC4.BAT AGENCY 14 ZIP STARTED  
10-24-2003 22:32:24 - EC4.BAT AGENCY 14 ZIP FINISHED  
10-24-2003 22:32:26 - EC4.BAT AGENCY 15 IMPORTED  
10-24-2003 22:32:26 - EC4.BAT AGENCY 15 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:32:26 - EC4.BAT AGENCY 15 STARTED  
10-24-2003 22:34:09 - EC4.BAT AGENCY 15 STATE CLIENTS INSERT STARTED  
10-24-2003 22:34:20 - EC4.BAT AGENCY 15 INSERTED  
10-24-2003 22:34:20 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:34:28 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:34:47 - EC4.BAT AGENCY 15 EXPORTED  
10-24-2003 22:34:57 - EC4.BAT AGENCY 15 DONE  
10-24-2003 22:34:57 - EC4.BAT AGENCY 15 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:34:57 - EC4.BAT AGENCY 15 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:34:57 - EC4.BAT AGENCY 15 ZIP STARTED  
10-24-2003 22:34:59 - EC4.BAT AGENCY 15 ZIP FINISHED  
10-24-2003 22:35:01 - EC4.BAT AGENCY 17 IMPORTED  
10-24-2003 22:35:01 - EC4.BAT AGENCY 17 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL

10-24-2003 22:35:02 - EC4.BAT AGENCY 17 STARTED  
10-24-2003 22:36:46 - EC4.BAT AGENCY 17 STATE CLIENTS INSERT STARTED  
10-24-2003 22:36:57 - EC4.BAT AGENCY 17 INSERTED  
10-24-2003 22:36:57 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:37:05 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:37:24 - EC4.BAT AGENCY 17 EXPORTED  
10-24-2003 22:37:34 - EC4.BAT AGENCY 17 DONE  
10-24-2003 22:37:34 - EC4.BAT AGENCY 17 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:37:34 - EC4.BAT AGENCY 17 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:37:35 - EC4.BAT AGENCY 17 ZIP STARTED  
10-24-2003 22:37:37 - EC4.BAT AGENCY 17 ZIP FINISHED  
10-24-2003 22:37:38 - EC4.BAT AGENCY 25 IMPORTED  
10-24-2003 22:37:38 - EC4.BAT AGENCY 25 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:37:39 - EC4.BAT AGENCY 25 STARTED  
10-24-2003 22:39:20 - EC4.BAT AGENCY 25 STATE CLIENTS INSERT STARTED  
10-24-2003 22:39:30 - EC4.BAT AGENCY 25 INSERTED  
10-24-2003 22:39:30 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:39:39 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:39:58 - EC4.BAT AGENCY 25 EXPORTED  
10-24-2003 22:40:08 - EC4.BAT AGENCY 25 DONE  
10-24-2003 22:40:08 - EC4.BAT AGENCY 25 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:40:09 - EC4.BAT AGENCY 25 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:40:09 - EC4.BAT AGENCY 25 ZIP STARTED  
10-24-2003 22:40:11 - EC4.BAT AGENCY 25 ZIP FINISHED  
10-24-2003 22:40:13 - EC4.BAT AGENCY 27 IMPORTED  
10-24-2003 22:40:13 - EC4.BAT AGENCY 27 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:40:13 - EC4.BAT AGENCY 27 STARTED  
10-24-2003 22:41:54 - EC4.BAT AGENCY 27 STATE CLIENTS INSERT STARTED  
10-24-2003 22:42:04 - EC4.BAT AGENCY 27 INSERTED  
10-24-2003 22:42:04 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:42:14 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:42:43 - EC4.BAT AGENCY 27 EXPORTED  
10-24-2003 22:42:53 - EC4.BAT AGENCY 27 DONE  
10-24-2003 22:42:53 - EC4.BAT AGENCY 27 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:42:54 - EC4.BAT AGENCY 27 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:42:54 - EC4.BAT AGENCY 27 ZIP STARTED  
10-24-2003 22:42:56 - EC4.BAT AGENCY 27 ZIP FINISHED  
10-24-2003 22:42:58 - EC4.BAT AGENCY 28 IMPORTED  
10-24-2003 22:42:58 - EC4.BAT AGENCY 28 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:42:58 - EC4.BAT AGENCY 28 STARTED  
10-24-2003 22:44:41 - EC4.BAT AGENCY 28 STATE CLIENTS INSERT STARTED  
10-24-2003 22:44:52 - EC4.BAT AGENCY 28 INSERTED  
10-24-2003 22:44:52 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:45:01 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:45:21 - EC4.BAT AGENCY 28 EXPORTED  
10-24-2003 22:45:31 - EC4.BAT AGENCY 28 DONE  
10-24-2003 22:45:31 - EC4.BAT AGENCY 28 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:45:32 - EC4.BAT AGENCY 28 EC4\_GET\_UPDATES FINISHED

10-24-2003 22:45:32 - EC4.BAT AGENCY 28 ZIP STARTED  
10-24-2003 22:45:34 - EC4.BAT AGENCY 28 ZIP FINISHED  
10-24-2003 22:45:36 - EC4.BAT AGENCY 29 IMPORTED  
10-24-2003 22:45:36 - EC4.BAT AGENCY 29 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:45:37 - EC4.BAT AGENCY 29 STARTED  
10-24-2003 22:47:19 - EC4.BAT AGENCY 29 STATE CLIENTS INSERT STARTED  
10-24-2003 22:47:30 - EC4.BAT AGENCY 29 INSERTED  
10-24-2003 22:47:30 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:47:38 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:47:57 - EC4.BAT AGENCY 29 EXPORTED  
10-24-2003 22:48:07 - EC4.BAT AGENCY 29 DONE  
10-24-2003 22:48:07 - EC4.BAT AGENCY 29 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:48:07 - EC4.BAT AGENCY 29 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:48:07 - EC4.BAT AGENCY 29 ZIP STARTED  
10-24-2003 22:48:10 - EC4.BAT AGENCY 29 ZIP FINISHED  
10-24-2003 22:48:11 - EC4.BAT AGENCY 30 IMPORTED  
10-24-2003 22:48:11 - EC4.BAT AGENCY 30 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:48:12 - EC4.BAT AGENCY 30 STARTED  
10-24-2003 22:49:54 - EC4.BAT AGENCY 30 STATE CLIENTS INSERT STARTED  
10-24-2003 22:50:05 - EC4.BAT AGENCY 30 INSERTED  
10-24-2003 22:50:05 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:50:13 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:50:32 - EC4.BAT AGENCY 30 EXPORTED  
10-24-2003 22:50:42 - EC4.BAT AGENCY 30 DONE  
10-24-2003 22:50:42 - EC4.BAT AGENCY 30 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:50:43 - EC4.BAT AGENCY 30 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:50:43 - EC4.BAT AGENCY 30 ZIP STARTED  
10-24-2003 22:50:45 - EC4.BAT AGENCY 30 ZIP FINISHED  
10-24-2003 22:50:47 - EC4.BAT AGENCY 32 IMPORTED  
10-24-2003 22:50:47 - EC4.BAT AGENCY 32 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:50:47 - EC4.BAT AGENCY 32 STARTED  
10-24-2003 22:52:29 - EC4.BAT AGENCY 32 STATE CLIENTS INSERT STARTED  
10-24-2003 22:52:40 - EC4.BAT AGENCY 32 INSERTED  
10-24-2003 22:52:40 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:52:48 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:53:06 - EC4.BAT AGENCY 32 EXPORTED  
10-24-2003 22:53:16 - EC4.BAT AGENCY 32 DONE  
10-24-2003 22:53:16 - EC4.BAT AGENCY 32 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:53:17 - EC4.BAT AGENCY 32 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:53:17 - EC4.BAT AGENCY 32 ZIP STARTED  
10-24-2003 22:53:19 - EC4.BAT AGENCY 32 ZIP FINISHED  
10-24-2003 22:53:19 - EC4.BAT IS DONE  
10-24-2003 22:53:20 - EC5.BAT STARTED  
10-24-2003 22:53:21 - EC\_PRINT.BAT STARTED  
10-24-2003 22:53:48 - EC\_PRINT.BAT FI REPORTS STARTED  
10-24-2003 22:53:58 - EC\_PRINT.BAT FI REPORTS FINISHED  
10-24-2003 22:54:08 - EC\_PRINT.BAT VENDOR REPORTS STARTED  
10-24-2003 22:54:18 - EC\_PRINT.BAT VENDOR REPORTS FINISHED

10-24-2003 22:54:28 - EC\_PRINT.BAT EC5.SQL STARTED  
 10-24-2003 22:54:39 - EC\_PRINT.BAT EC5.SQL FINISHED  
 10-24-2003 22:54:40 - EC\_PRINT.BAT IS DONE  
 10-24-2003 22:54:40 - EC5.BAT IS DONE  
 10-24-2003 22:54:40 - EC1.BAT STARTING ARCHIVE OF EOD LOGS  
 10-24-2003 22:54:41 - EC1.BAT FINISHED ARCHIVE OF EOD LOGS  
 10-24-2003 22:54:41 - EC1.BAT FINISHED EOD PROCESSING IS COMPLETE  
 Column 1: The date the EOD script was executed (10) characters.  
 Column 2: The time (HH:MM:SS) that the EOD script executed the particular function of record.  
 Column 3: Explanation of what is being executed within the EOD script.

**EC\_HIS.LOG** - A historic list that details by date and time that all Central Agency scripts initiated and finished. An example of the content of this file is as follows:

10-24-2003 19:30:12 - EC1 STARTED  
 10-24-2003 19:30:14 - EC1.BAT EOD ALREADY RUN CHECK STARTED  
 10-24-2003 19:30:14 - EC1.BAT EOD ALREADY RUN CHECK FINISHED  
 10-24-2003 19:30:19 - EC1.BAT AIM APPLICATION FILE UPDATE STARTED  
 10-24-2003 19:30:19 - EC1.BAT AIM APPLICATION FILE UPDATE FINISHED  
 10-24-2003 19:30:39 - EC1.BAT SQL STARTED  
 10-24-2003 19:37:35 - BUILDING ANPGA.DAT  
 10-24-2003 19:37:45 - BUILDING ANVENDOR.DAT  
 10-24-2003 19:38:09 - EC1.BAT SQL FINISHED  
 10-24-2003 19:38:19 - STARTED EC2.BAT  
 10-24-2003 19:38:24 - EC2.BAT CHECKING AGENCY 01  
 10-24-2003 19:38:24 - EC2.BAT AGENCY 01 FOUND  
 10-24-2003 19:38:40 - EC2.BAT AGENCY 01 IMPORTED  
 10-24-2003 19:42:33 - EC2.BAT AGENCY 01 LOADED  
 10-24-2003 19:42:43 - EC2.BAT CHECKING AGENCY 02  
 10-24-2003 19:42:43 - EC2.BAT AGENCY 02 FOUND  
 10-24-2003 19:42:58 - EC2.BAT AGENCY 02 IMPORTED  
 10-24-2003 19:44:15 - EC2.BAT AGENCY 02 LOADED  
 10-24-2003 19:44:25 - EC2.BAT CHECKING AGENCY 03  
 10-24-2003 19:44:25 - EC2.BAT AGENCY 03 FOUND  
 10-24-2003 19:44:42 - EC2.BAT AGENCY 03 IMPORTED  
 10-24-2003 19:45:45 - EC2.BAT AGENCY 03 LOADED  
 10-24-2003 19:45:55 - EC2.BAT CHECKING AGENCY 04  
 10-24-2003 19:45:55 - EC2.BAT AGENCY 04 FOUND  
 10-24-2003 19:46:11 - EC2.BAT AGENCY 04 IMPORTED  
 10-24-2003 19:47:09 - EC2.BAT AGENCY 04 LOADED  
 10-24-2003 19:47:19 - EC2.BAT CHECKING AGENCY 05  
 10-24-2003 19:47:19 - EC2.BAT AGENCY 05 FOUND  
 10-24-2003 19:47:34 - EC2.BAT AGENCY 05 IMPORTED  
 10-24-2003 19:48:37 - EC2.BAT AGENCY 05 LOADED  
 10-24-2003 19:48:47 - EC2.BAT CHECKING AGENCY 06  
 10-24-2003 19:48:47 - EC2.BAT AGENCY 06 FOUND  
 10-24-2003 19:49:07 - EC2.BAT AGENCY 06 IMPORTED  
 10-24-2003 19:49:55 - EC2.BAT AGENCY 06 LOADED

10-24-2003 19:50:05 - EC2.BAT CHECKING AGENCY 07  
10-24-2003 19:50:05 - EC2.BAT AGENCY 07 FOUND  
10-24-2003 19:50:34 - EC2.BAT AGENCY 07 IMPORTED  
10-24-2003 20:10:54 - EC2.BAT AGENCY 07 LOADED  
10-24-2003 20:11:05 - EC2.BAT CHECKING AGENCY 08  
10-24-2003 20:11:05 - EC2.BAT AGENCY 08 FOUND  
10-24-2003 20:11:21 - EC2.BAT AGENCY 08 IMPORTED  
10-24-2003 20:12:41 - EC2.BAT AGENCY 08 LOADED  
10-24-2003 20:12:51 - EC2.BAT CHECKING AGENCY 09  
10-24-2003 20:12:51 - EC2.BAT AGENCY 09 FOUND  
10-24-2003 20:13:13 - EC2.BAT AGENCY 09 IMPORTED  
10-24-2003 20:14:13 - EC2.BAT AGENCY 09 LOADED  
10-24-2003 20:14:23 - EC2.BAT CHECKING AGENCY 10  
10-24-2003 20:14:23 - EC2.BAT AGENCY 10 FOUND  
10-24-2003 20:14:45 - EC2.BAT AGENCY 10 IMPORTED  
10-24-2003 20:21:08 - EC2.BAT AGENCY 10 LOADED  
10-24-2003 20:21:19 - EC2.BAT CHECKING AGENCY 11  
10-24-2003 20:21:19 - EC2.BAT AGENCY 11 FOUND  
10-24-2003 20:21:43 - EC2.BAT AGENCY 11 IMPORTED  
10-24-2003 20:25:09 - EC2.BAT AGENCY 11 LOADED  
10-24-2003 20:25:19 - EC2.BAT CHECKING AGENCY 13  
10-24-2003 20:25:19 - EC2.BAT AGENCY 13 FOUND  
10-24-2003 20:25:38 - EC2.BAT AGENCY 13 IMPORTED  
10-24-2003 20:27:41 - EC2.BAT AGENCY 13 LOADED  
10-24-2003 20:27:51 - EC2.BAT CHECKING AGENCY 14  
10-24-2003 20:27:51 - EC2.BAT AGENCY 14 FOUND  
10-24-2003 20:28:10 - EC2.BAT AGENCY 14 IMPORTED  
10-24-2003 20:31:39 - EC2.BAT AGENCY 14 LOADED  
10-24-2003 20:31:49 - EC2.BAT CHECKING AGENCY 15  
10-24-2003 20:31:49 - EC2.BAT AGENCY 15 FOUND  
10-24-2003 20:32:12 - EC2.BAT AGENCY 15 IMPORTED  
10-24-2003 20:33:04 - EC2.BAT AGENCY 15 LOADED  
10-24-2003 20:33:14 - EC2.BAT CHECKING AGENCY 17  
10-24-2003 20:33:14 - EC2.BAT AGENCY 17 FOUND  
10-24-2003 20:33:32 - EC2.BAT AGENCY 17 IMPORTED  
10-24-2003 20:34:42 - EC2.BAT AGENCY 17 LOADED  
10-24-2003 20:34:52 - EC2.BAT CHECKING AGENCY 25  
10-24-2003 20:34:52 - EC2.BAT AGENCY 25 FOUND  
10-24-2003 20:35:16 - EC2.BAT AGENCY 25 IMPORTED  
10-24-2003 20:36:05 - EC2.BAT AGENCY 25 LOADED  
10-24-2003 20:36:15 - EC2.BAT CHECKING AGENCY 27  
10-24-2003 20:36:15 - EC2.BAT AGENCY 27 FOUND  
10-24-2003 20:36:36 - EC2.BAT AGENCY 27 IMPORTED  
10-24-2003 20:38:06 - EC2.BAT AGENCY 27 LOADED  
10-24-2003 20:38:16 - EC2.BAT CHECKING AGENCY 28  
10-24-2003 20:38:16 - EC2.BAT AGENCY 28 FOUND  
10-24-2003 20:38:35 - EC2.BAT AGENCY 28 IMPORTED  
10-24-2003 20:41:16 - EC2.BAT AGENCY 28 LOADED  
10-24-2003 20:41:26 - EC2.BAT CHECKING AGENCY 29  
10-24-2003 20:41:26 - EC2.BAT AGENCY 29 FOUND  
10-24-2003 20:41:45 - EC2.BAT AGENCY 29 IMPORTED

10-24-2003 20:43:23 - EC2.BAT AGENCY 29 LOADED  
10-24-2003 20:43:33 - EC2.BAT CHECKING AGENCY 30  
10-24-2003 20:43:33 - EC2.BAT AGENCY 30 FOUND  
10-24-2003 20:43:53 - EC2.BAT AGENCY 30 IMPORTED  
10-24-2003 20:45:45 - EC2.BAT AGENCY 30 LOADED  
10-24-2003 20:45:55 - EC2.BAT CHECKING AGENCY 32  
10-24-2003 20:45:55 - EC2.BAT AGENCY 32 FOUND  
10-24-2003 20:46:13 - EC2.BAT AGENCY 32 IMPORTED  
10-24-2003 20:47:08 - EC2.BAT AGENCY 32 LOADED  
10-24-2003 20:47:18 - EC2.BAT EC2.SQL STARTED  
10-24-2003 21:13:45 - EC2.BAT EC2.SQL FINISHED  
10-24-2003 21:13:55 - EC2.BAT FINISHED  
10-24-2003 21:13:55 - EC4.BAT STARTED  
10-24-2003 21:14:57 - EC4.BAT FTPTRANS.EXE STARTED  
10-24-2003 21:17:36 - EC4.BAT FTPTRANS.EXE FINISHED  
10-24-2003 21:21:46 - EC4.BAT SQLOAD OF FSMC STARTED  
10-24-2003 21:22:13 - EC4.BAT SQLOAD OF FSMC FINISHED  
10-24-2003 21:22:23 - EC4.BAT BANKSQL STARTED  
10-24-2003 21:33:40 - EC4.BAT BANKSQL FINISHED  
10-24-2003 21:33:50 - EC4.BAT EC4.SQL STARTED  
10-24-2003 21:54:20 - EC4.BAT EC4.SQL FINISHED  
10-24-2003 21:54:30 - EC4.BAT EC4\_CSC.SQL STARTED  
10-24-2003 21:54:35 - EC4.BAT EC4\_CSC.SQL FINISHED  
10-24-2003 21:56:14 - EC4.BAT AGENCY 01 IMPORTED  
10-24-2003 21:56:24 - EC4.BAT AGENCY 01 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 21:56:24 - EC4.BAT AGENCY 01 STARTED  
10-24-2003 21:58:17 - EC4.BAT AGENCY 01 STATE CLIENTS INSERT STARTED  
10-24-2003 21:58:28 - EC4.BAT AGENCY 01 INSERTED  
10-24-2003 21:58:28 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 21:58:38 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 21:59:09 - EC4.BAT AGENCY 01 EXPORTED  
10-24-2003 21:59:19 - EC4.BAT AGENCY 01 DONE  
10-24-2003 21:59:19 - EC4.BAT AGENCY 01 EC4\_GET\_UPDATES STARTED  
10-24-2003 21:59:20 - EC4.BAT AGENCY 01 EC4\_GET\_UPDATES FINISHED  
10-24-2003 21:59:20 - EC4.BAT AGENCY 01 ZIP STARTED  
10-24-2003 21:59:22 - EC4.BAT AGENCY 01 ZIP FINISHED  
10-24-2003 21:59:24 - EC4.BAT AGENCY 02 IMPORTED  
10-24-2003 21:59:24 - EC4.BAT AGENCY 02 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 21:59:24 - EC4.BAT AGENCY 02 STARTED  
10-24-2003 22:01:07 - EC4.BAT AGENCY 02 STATE CLIENTS INSERT STARTED  
10-24-2003 22:01:17 - EC4.BAT AGENCY 02 INSERTED  
10-24-2003 22:01:18 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:01:27 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:01:46 - EC4.BAT AGENCY 02 EXPORTED  
10-24-2003 22:01:56 - EC4.BAT AGENCY 02 DONE  
10-24-2003 22:01:56 - EC4.BAT AGENCY 02 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:01:57 - EC4.BAT AGENCY 02 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:01:57 - EC4.BAT AGENCY 02 ZIP STARTED  
10-24-2003 22:01:59 - EC4.BAT AGENCY 02 ZIP FINISHED

10-24-2003 22:02:01 - EC4.BAT AGENCY 03 IMPORTED  
10-24-2003 22:02:01 - EC4.BAT AGENCY 03 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:02:02 - EC4.BAT AGENCY 03 STARTED  
10-24-2003 22:03:46 - EC4.BAT AGENCY 03 STATE CLIENTS INSERT STARTED  
10-24-2003 22:03:57 - EC4.BAT AGENCY 03 INSERTED  
10-24-2003 22:03:57 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:04:06 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:04:35 - EC4.BAT AGENCY 03 EXPORTED  
10-24-2003 22:04:46 - EC4.BAT AGENCY 03 DONE  
10-24-2003 22:04:46 - EC4.BAT AGENCY 03 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:04:46 - EC4.BAT AGENCY 03 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:04:46 - EC4.BAT AGENCY 03 ZIP STARTED  
10-24-2003 22:04:49 - EC4.BAT AGENCY 03 ZIP FINISHED  
10-24-2003 22:04:50 - EC4.BAT AGENCY 04 IMPORTED  
10-24-2003 22:04:50 - EC4.BAT AGENCY 04 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:04:51 - EC4.BAT AGENCY 04 STARTED  
10-24-2003 22:06:36 - EC4.BAT AGENCY 04 STATE CLIENTS INSERT STARTED  
10-24-2003 22:06:47 - EC4.BAT AGENCY 04 INSERTED  
10-24-2003 22:06:47 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:06:55 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:07:14 - EC4.BAT AGENCY 04 EXPORTED  
10-24-2003 22:07:24 - EC4.BAT AGENCY 04 DONE  
10-24-2003 22:07:24 - EC4.BAT AGENCY 04 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:07:24 - EC4.BAT AGENCY 04 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:07:25 - EC4.BAT AGENCY 04 ZIP STARTED  
10-24-2003 22:07:27 - EC4.BAT AGENCY 04 ZIP FINISHED  
10-24-2003 22:07:28 - EC4.BAT AGENCY 05 IMPORTED  
10-24-2003 22:07:28 - EC4.BAT AGENCY 05 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:07:29 - EC4.BAT AGENCY 05 STARTED  
10-24-2003 22:09:15 - EC4.BAT AGENCY 05 STATE CLIENTS INSERT STARTED  
10-24-2003 22:09:25 - EC4.BAT AGENCY 05 INSERTED  
10-24-2003 22:09:25 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:09:34 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:09:53 - EC4.BAT AGENCY 05 EXPORTED  
10-24-2003 22:10:03 - EC4.BAT AGENCY 05 DONE  
10-24-2003 22:10:03 - EC4.BAT AGENCY 05 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:10:04 - EC4.BAT AGENCY 05 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:10:04 - EC4.BAT AGENCY 05 ZIP STARTED  
10-24-2003 22:10:06 - EC4.BAT AGENCY 05 ZIP FINISHED  
10-24-2003 22:10:08 - EC4.BAT AGENCY 06 IMPORTED  
10-24-2003 22:10:08 - EC4.BAT AGENCY 06 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:10:09 - EC4.BAT AGENCY 06 STARTED  
10-24-2003 22:11:52 - EC4.BAT AGENCY 06 STATE CLIENTS INSERT STARTED  
10-24-2003 22:12:03 - EC4.BAT AGENCY 06 INSERTED  
10-24-2003 22:12:03 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:12:12 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:12:30 - EC4.BAT AGENCY 06 EXPORTED

10-24-2003 22:12:40 - EC4.BAT AGENCY 06 DONE  
10-24-2003 22:12:40 - EC4.BAT AGENCY 06 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:12:41 - EC4.BAT AGENCY 06 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:12:41 - EC4.BAT AGENCY 06 ZIP STARTED  
10-24-2003 22:12:43 - EC4.BAT AGENCY 06 ZIP FINISHED  
10-24-2003 22:12:45 - EC4.BAT AGENCY 07 IMPORTED  
10-24-2003 22:12:45 - EC4.BAT AGENCY 07 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:12:45 - EC4.BAT AGENCY 07 STARTED  
10-24-2003 22:14:36 - EC4.BAT AGENCY 07 STATE CLIENTS INSERT STARTED  
10-24-2003 22:14:46 - EC4.BAT AGENCY 07 INSERTED  
10-24-2003 22:14:46 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:14:55 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:15:41 - EC4.BAT AGENCY 07 EXPORTED  
10-24-2003 22:15:51 - EC4.BAT AGENCY 07 DONE  
10-24-2003 22:15:51 - EC4.BAT AGENCY 07 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:15:52 - EC4.BAT AGENCY 07 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:15:52 - EC4.BAT AGENCY 07 ZIP STARTED  
10-24-2003 22:15:57 - EC4.BAT AGENCY 07 ZIP FINISHED  
10-24-2003 22:16:00 - EC4.BAT AGENCY 08 IMPORTED  
10-24-2003 22:16:00 - EC4.BAT AGENCY 08 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:16:00 - EC4.BAT AGENCY 08 STARTED  
10-24-2003 22:17:49 - EC4.BAT AGENCY 08 STATE CLIENTS INSERT STARTED  
10-24-2003 22:18:00 - EC4.BAT AGENCY 08 INSERTED  
10-24-2003 22:18:00 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:18:09 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:18:29 - EC4.BAT AGENCY 08 EXPORTED  
10-24-2003 22:18:39 - EC4.BAT AGENCY 08 DONE  
10-24-2003 22:18:39 - EC4.BAT AGENCY 08 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:18:40 - EC4.BAT AGENCY 08 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:18:40 - EC4.BAT AGENCY 08 ZIP STARTED  
10-24-2003 22:18:42 - EC4.BAT AGENCY 08 ZIP FINISHED  
10-24-2003 22:18:44 - EC4.BAT AGENCY 09 IMPORTED  
10-24-2003 22:18:44 - EC4.BAT AGENCY 09 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:18:45 - EC4.BAT AGENCY 09 STARTED  
10-24-2003 22:20:28 - EC4.BAT AGENCY 09 STATE CLIENTS INSERT STARTED  
10-24-2003 22:20:39 - EC4.BAT AGENCY 09 INSERTED  
10-24-2003 22:20:39 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:20:48 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:21:06 - EC4.BAT AGENCY 09 EXPORTED  
10-24-2003 22:21:16 - EC4.BAT AGENCY 09 DONE  
10-24-2003 22:21:16 - EC4.BAT AGENCY 09 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:21:17 - EC4.BAT AGENCY 09 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:21:17 - EC4.BAT AGENCY 09 ZIP STARTED  
10-24-2003 22:21:19 - EC4.BAT AGENCY 09 ZIP FINISHED  
10-24-2003 22:21:21 - EC4.BAT AGENCY 10 IMPORTED  
10-24-2003 22:21:21 - EC4.BAT AGENCY 10 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:21:21 - EC4.BAT AGENCY 10 STARTED

10-24-2003 22:23:04 - EC4.BAT AGENCY 10 STATE CLIENTS INSERT STARTED  
10-24-2003 22:23:15 - EC4.BAT AGENCY 10 INSERTED  
10-24-2003 22:23:15 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:23:25 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:24:03 - EC4.BAT AGENCY 10 EXPORTED  
10-24-2003 22:24:14 - EC4.BAT AGENCY 10 DONE  
10-24-2003 22:24:14 - EC4.BAT AGENCY 10 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:24:16 - EC4.BAT AGENCY 10 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:24:16 - EC4.BAT AGENCY 10 ZIP STARTED  
10-24-2003 22:24:19 - EC4.BAT AGENCY 10 ZIP FINISHED  
10-24-2003 22:24:21 - EC4.BAT AGENCY 11 IMPORTED  
10-24-2003 22:24:21 - EC4.BAT AGENCY 11 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:24:21 - EC4.BAT AGENCY 11 STARTED  
10-24-2003 22:26:04 - EC4.BAT AGENCY 11 STATE CLIENTS INSERT STARTED  
10-24-2003 22:26:15 - EC4.BAT AGENCY 11 INSERTED  
10-24-2003 22:26:15 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:26:23 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:26:45 - EC4.BAT AGENCY 11 EXPORTED  
10-24-2003 22:26:56 - EC4.BAT AGENCY 11 DONE  
10-24-2003 22:26:56 - EC4.BAT AGENCY 11 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:26:56 - EC4.BAT AGENCY 11 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:26:56 - EC4.BAT AGENCY 11 ZIP STARTED  
10-24-2003 22:26:59 - EC4.BAT AGENCY 11 ZIP FINISHED  
10-24-2003 22:27:01 - EC4.BAT AGENCY 13 IMPORTED  
10-24-2003 22:27:01 - EC4.BAT AGENCY 13 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:27:02 - EC4.BAT AGENCY 13 STARTED  
10-24-2003 22:28:45 - EC4.BAT AGENCY 13 STATE CLIENTS INSERT STARTED  
10-24-2003 22:28:57 - EC4.BAT AGENCY 13 INSERTED  
10-24-2003 22:28:57 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:29:05 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:29:32 - EC4.BAT AGENCY 13 EXPORTED  
10-24-2003 22:29:42 - EC4.BAT AGENCY 13 DONE  
10-24-2003 22:29:42 - EC4.BAT AGENCY 13 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:29:43 - EC4.BAT AGENCY 13 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:29:43 - EC4.BAT AGENCY 13 ZIP STARTED  
10-24-2003 22:29:45 - EC4.BAT AGENCY 13 ZIP FINISHED  
10-24-2003 22:29:47 - EC4.BAT AGENCY 14 IMPORTED  
10-24-2003 22:29:47 - EC4.BAT AGENCY 14 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:29:48 - EC4.BAT AGENCY 14 STARTED  
10-24-2003 22:31:31 - EC4.BAT AGENCY 14 STATE CLIENTS INSERT STARTED  
10-24-2003 22:31:41 - EC4.BAT AGENCY 14 INSERTED  
10-24-2003 22:31:41 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:31:50 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:32:11 - EC4.BAT AGENCY 14 EXPORTED  
10-24-2003 22:32:21 - EC4.BAT AGENCY 14 DONE  
10-24-2003 22:32:21 - EC4.BAT AGENCY 14 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:32:22 - EC4.BAT AGENCY 14 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:32:22 - EC4.BAT AGENCY 14 ZIP STARTED

10-24-2003 22:32:24 - EC4.BAT AGENCY 14 ZIP FINISHED  
10-24-2003 22:32:26 - EC4.BAT AGENCY 15 IMPORTED  
10-24-2003 22:32:26 - EC4.BAT AGENCY 15 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:32:26 - EC4.BAT AGENCY 15 STARTED  
10-24-2003 22:34:09 - EC4.BAT AGENCY 15 STATE CLIENTS INSERT STARTED  
10-24-2003 22:34:20 - EC4.BAT AGENCY 15 INSERTED  
10-24-2003 22:34:20 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:34:28 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:34:47 - EC4.BAT AGENCY 15 EXPORTED  
10-24-2003 22:34:57 - EC4.BAT AGENCY 15 DONE  
10-24-2003 22:34:57 - EC4.BAT AGENCY 15 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:34:57 - EC4.BAT AGENCY 15 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:34:57 - EC4.BAT AGENCY 15 ZIP STARTED  
10-24-2003 22:34:59 - EC4.BAT AGENCY 15 ZIP FINISHED  
10-24-2003 22:35:01 - EC4.BAT AGENCY 17 IMPORTED  
10-24-2003 22:35:01 - EC4.BAT AGENCY 17 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:35:02 - EC4.BAT AGENCY 17 STARTED  
10-24-2003 22:36:46 - EC4.BAT AGENCY 17 STATE CLIENTS INSERT STARTED  
10-24-2003 22:36:57 - EC4.BAT AGENCY 17 INSERTED  
10-24-2003 22:36:57 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:37:05 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:37:24 - EC4.BAT AGENCY 17 EXPORTED  
10-24-2003 22:37:34 - EC4.BAT AGENCY 17 DONE  
10-24-2003 22:37:34 - EC4.BAT AGENCY 17 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:37:34 - EC4.BAT AGENCY 17 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:37:35 - EC4.BAT AGENCY 17 ZIP STARTED  
10-24-2003 22:37:37 - EC4.BAT AGENCY 17 ZIP FINISHED  
10-24-2003 22:37:38 - EC4.BAT AGENCY 25 IMPORTED  
10-24-2003 22:37:38 - EC4.BAT AGENCY 25 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:37:39 - EC4.BAT AGENCY 25 STARTED  
10-24-2003 22:39:20 - EC4.BAT AGENCY 25 STATE CLIENTS INSERT STARTED  
10-24-2003 22:39:30 - EC4.BAT AGENCY 25 INSERTED  
10-24-2003 22:39:30 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:39:39 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:39:58 - EC4.BAT AGENCY 25 EXPORTED  
10-24-2003 22:40:08 - EC4.BAT AGENCY 25 DONE  
10-24-2003 22:40:08 - EC4.BAT AGENCY 25 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:40:09 - EC4.BAT AGENCY 25 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:40:09 - EC4.BAT AGENCY 25 ZIP STARTED  
10-24-2003 22:40:11 - EC4.BAT AGENCY 25 ZIP FINISHED  
10-24-2003 22:40:13 - EC4.BAT AGENCY 27 IMPORTED  
10-24-2003 22:40:13 - EC4.BAT AGENCY 27 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:40:13 - EC4.BAT AGENCY 27 STARTED  
10-24-2003 22:41:54 - EC4.BAT AGENCY 27 STATE CLIENTS INSERT STARTED  
10-24-2003 22:42:04 - EC4.BAT AGENCY 27 INSERTED  
10-24-2003 22:42:04 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:42:14 - EC4.BAT EC4\_RETRIEVE FINISHED

10-24-2003 22:42:43 - EC4.BAT AGENCY 27 EXPORTED  
10-24-2003 22:42:53 - EC4.BAT AGENCY 27 DONE  
10-24-2003 22:42:53 - EC4.BAT AGENCY 27 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:42:54 - EC4.BAT AGENCY 27 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:42:54 - EC4.BAT AGENCY 27 ZIP STARTED  
10-24-2003 22:42:56 - EC4.BAT AGENCY 27 ZIP FINISHED  
10-24-2003 22:42:58 - EC4.BAT AGENCY 28 IMPORTED  
10-24-2003 22:42:58 - EC4.BAT AGENCY 28 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:42:58 - EC4.BAT AGENCY 28 STARTED  
10-24-2003 22:44:41 - EC4.BAT AGENCY 28 STATE CLIENTS INSERT STARTED  
10-24-2003 22:44:52 - EC4.BAT AGENCY 28 INSERTED  
10-24-2003 22:44:52 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:45:01 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:45:21 - EC4.BAT AGENCY 28 EXPORTED  
10-24-2003 22:45:31 - EC4.BAT AGENCY 28 DONE  
10-24-2003 22:45:31 - EC4.BAT AGENCY 28 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:45:32 - EC4.BAT AGENCY 28 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:45:32 - EC4.BAT AGENCY 28 ZIP STARTED  
10-24-2003 22:45:34 - EC4.BAT AGENCY 28 ZIP FINISHED  
10-24-2003 22:45:36 - EC4.BAT AGENCY 29 IMPORTED  
10-24-2003 22:45:36 - EC4.BAT AGENCY 29 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:45:37 - EC4.BAT AGENCY 29 STARTED  
10-24-2003 22:47:19 - EC4.BAT AGENCY 29 STATE CLIENTS INSERT STARTED  
10-24-2003 22:47:30 - EC4.BAT AGENCY 29 INSERTED  
10-24-2003 22:47:30 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:47:38 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:47:57 - EC4.BAT AGENCY 29 EXPORTED  
10-24-2003 22:48:07 - EC4.BAT AGENCY 29 DONE  
10-24-2003 22:48:07 - EC4.BAT AGENCY 29 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:48:07 - EC4.BAT AGENCY 29 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:48:07 - EC4.BAT AGENCY 29 ZIP STARTED  
10-24-2003 22:48:10 - EC4.BAT AGENCY 29 ZIP FINISHED  
10-24-2003 22:48:11 - EC4.BAT AGENCY 30 IMPORTED  
10-24-2003 22:48:11 - EC4.BAT AGENCY 30 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL  
10-24-2003 22:48:12 - EC4.BAT AGENCY 30 STARTED  
10-24-2003 22:49:54 - EC4.BAT AGENCY 30 STATE CLIENTS INSERT STARTED  
10-24-2003 22:50:05 - EC4.BAT AGENCY 30 INSERTED  
10-24-2003 22:50:05 - EC4.BAT EC4\_RETRIEVE STARTED  
10-24-2003 22:50:13 - EC4.BAT EC4\_RETRIEVE FINISHED  
10-24-2003 22:50:32 - EC4.BAT AGENCY 30 EXPORTED  
10-24-2003 22:50:42 - EC4.BAT AGENCY 30 DONE  
10-24-2003 22:50:42 - EC4.BAT AGENCY 30 EC4\_GET\_UPDATES STARTED  
10-24-2003 22:50:43 - EC4.BAT AGENCY 30 EC4\_GET\_UPDATES FINISHED  
10-24-2003 22:50:43 - EC4.BAT AGENCY 30 ZIP STARTED  
10-24-2003 22:50:45 - EC4.BAT AGENCY 30 ZIP FINISHED  
10-24-2003 22:50:47 - EC4.BAT AGENCY 32 IMPORTED  
10-24-2003 22:50:47 - EC4.BAT AGENCY 32 STARTING  
EC4\_AGCY\_BEING\_PROCESSED.SQL

10-24-2003 22:50:47 - EC4.BAT AGENCY 32 STARTED  
 10-24-2003 22:52:29 - EC4.BAT AGENCY 32 STATE CLIENTS INSERT STARTED  
 10-24-2003 22:52:40 - EC4.BAT AGENCY 32 INSERTED  
 10-24-2003 22:52:40 - EC4.BAT EC4\_RETRIEVE STARTED  
 10-24-2003 22:52:48 - EC4.BAT EC4\_RETRIEVE FINISHED  
 10-24-2003 22:53:06 - EC4.BAT AGENCY 32 EXPORTED  
 10-24-2003 22:53:16 - EC4.BAT AGENCY 32 DONE  
 10-24-2003 22:53:16 - EC4.BAT AGENCY 32 EC4\_GET\_UPDATES STARTED  
 10-24-2003 22:53:17 - EC4.BAT AGENCY 32 EC4\_GET\_UPDATES FINISHED  
 10-24-2003 22:53:17 - EC4.BAT AGENCY 32 ZIP STARTED  
 10-24-2003 22:53:19 - EC4.BAT AGENCY 32 ZIP FINISHED  
 10-24-2003 22:53:19 - EC4.BAT IS DONE  
 10-24-2003 22:53:20 - EC5.BAT STARTED  
 10-24-2003 22:53:21 - EC\_PRINT.BAT STARTED  
 10-24-2003 22:53:48 - EC\_PRINT.BAT FI REPORTS STARTED  
 10-24-2003 22:53:58 - EC\_PRINT.BAT FI REPORTS FINISHED  
 10-24-2003 22:54:08 - EC\_PRINT.BAT VENDOR REPORTS STARTED  
 10-24-2003 22:54:18 - EC\_PRINT.BAT VENDOR REPORTS FINISHED  
 10-24-2003 22:54:28 - EC\_PRINT.BAT EC5.SQL STARTED  
 10-24-2003 22:54:39 - EC\_PRINT.BAT EC5.SQL FINISHED  
 10-24-2003 22:54:40 - EC\_PRINT.BAT IS DONE  
 10-24-2003 22:54:40 - EC5.BAT IS DONE  
 10-24-2003 22:54:40 - EC1.BAT STARTING ARCHIVE OF EOD LOGS  
 10-24-2003 22:54:41 - EC1.BAT FINISHED ARCHIVE OF EOD LOGS  
 10-24-2003 22:54:41 - EC1.BAT FINISHED EOD PROCESSING IS COMPLETE

Column 1: The date the EOD script was executed (10) characters.  
 Column 2: The time (HH:MM:SS) that the EOD script executed the particular function of record.  
 Column 3: Explanation of what is being executed within the EOD script.

FSMC\_FTP.LOG - A daily log file that details by date, time, and filename when files were transferred to the bank and when they were retrieved. An example of the content of this file is as follows:

Date/Time: 8/9/00 3:22:34 PM Internet Connection made 13369348  
 Date/Time: 8/9/00 3:22:38 PM Connecting to FSMC Connection Successful  
 Date/Time: 8/9/00 3:22:38 PM Directory set at FSMC /client/c0020  
 Date/Time: 8/9/00 3:22:38 PM File Not Transferred: e:\aim\eod\central\load\pdaz0809.DAT  
 Date/Time: 8/9/00 3:22:39 PM File Not Transferred: e:\aim\eod\central\load\RPTS0809.EXE  
 Date/Time: 8/9/00 3:22:39 PM File Not Transferred: e:\aim\eod\central\load\pdaz0808.DAT  
 Date/Time: 8/9/00 3:22:40 PM File Not Transferred: e:\aim\eod\central\load\RPTS0808.EXE  
 Date/Time: 8/9/00 3:22:40 PM File Not Transferred: e:\aim\eod\central\load\pdaz0807.DAT  
 Date/Time: 8/9/00 3:22:40 PM File Not Transferred: e:\aim\eod\central\load\RPTS0807.EXE  
 Date/Time: 8/9/00 3:22:41 PM File Not Transferred: e:\aim\eod\central\load\pdaz0806.DAT  
 Date/Time: 8/9/00 3:22:41 PM File Not Transferred: e:\aim\eod\central\load\RPTS0806.EXE  
 Date/Time: 8/9/00 3:22:42 PM File Not Transferred: e:\aim\eod\central\load\pdaz0805.DAT  
 Date/Time: 8/9/00 3:22:42 PM File Not Transferred: e:\aim\eod\central\load\RPTS0805.EXE

Date/Time: 8/9/00 3:22:50 PM File Transferred to FSMC: \eod\central\dump\AZVENDOR.DAT  
 Date/Time: 8/9/00 3:22:54 PM File Transferred to FSMC: \eod\central\dump\AZPGA.DAT  
 Date/Time: 8/9/00 3:22:57 PM File Transferred to FSMC: isaz0809.DAT  
 Date/Time: 8/9/00 3:22:57 PM SQL Load file generated:  
 sqldr control=e:\aim\eod\central\load\PDZ0809.ctf,  
 userid=eodadm/eodadm@aim00,log=e:\aim\eod\central\logs\import\_files.log

**AGCY\_%\_LOG** - This script details for a single local agency (in this case 01) the table data that was extracted from the agency's base tables, updated at the Central server, then exported from the Central server database. An example of the content of this file is as follows:

```
----- INSERT -----
EA1 inserting :E_EOD_CONTROLS
Table truncated.
1 row created.

EA1 inserting :E_O_STAFF_MEMBERS
Table truncated.
0 rows created.
```

The insert section shows which temporary base table is currently being updated with new values, whether the table is being truncated, and how many rows of new data are being created.

```
----- EXPORT -----
About to export specified tables via Conventional Path ...
.. exporting table      E_O_STAFF_MEMBERS    0 rows exported
.. exporting table      E_EOD_CONTROLS    1 rows exported
.. exporting table      E_I_FOOD_INSTRUMENT_TYPES    4 rows exported
```

The export section is split into three columns. The first column indicates that the process is currently exporting a table, the second column shows the name of the table being exported, and the third column lists how many rows from the table that are being exported.

**AGCY\_%\_HIS.LOG** - This script is a historic log which details for a single local agency (in this case 01) the table data that was extracted from the agency's base tables, updated at the Central server, then exported from the Central server database. An example of the content of this file is as follows:

```
----- TRUNCATE -----
truncating :E_EOD_CONTROLS
truncating :E_O_STAFF_MEMBERS
truncating :E_O_ORGANIZATIONAL_UNITS
truncating :E_O_ORG_PROGRAMS
truncating :E_O_ORG_UNIT_PHONES
```

The truncate section indicates the table that is currently being truncated for the import section.

```

----- IMPORT -----
Export file created by EXPORT:V07.03.03 via conventional path
. importing EODADM's objects into EODADM
.. importing table "E_O_STAFF_MEMBERS" 0 rows imported
.. importing table "E_EOD_CONTROLS" 1 rows imported
.. importing table "E_I_FOOD_INSTRUMENT_TYPES" 0 rows imported
.. importing table "E_O_ORGANIZATIONAL_UNITS" 0 rows imported
.. importing table "E_I_FOOD_INSTRUMENT_FOODS" 0 rows imported

```

The import section is separated into three columns. The first column indicates that a table is currently being imported, the second column shows which table is being imported, and the third column lists how many rows are being imported.

```

---- DELETE INSERT UPDATE DELETE -----
UPDATING EOD_BY
1 row updated.
Commit complete.

```

START\_TIME

-----  
10/24/2003 08:36 PM

RUNNING EC2\_UPD\_E\_C\_CLIENTS.SQL

0 rows updated.

Input truncated to 1 characters

Commit complete.

FINISHED EC2\_UPD\_E\_C\_CLIENTS.SQL

This section shows the output for the EC2\_UPD\_E\_C\_CLIENTS sql script and indicates the time in which the script begins. This script updates the last\_first\_bdate column of the e\_c\_clients to be used in the ec2\_dual\_enroll.sql script later on.

```

START_TIME -----
----- 09/15/1999 07:59 PM
RUNNING EC2_DELTRIG.SQL
PL/SQL procedure successfully completed.

```

FINISHED EC2\_DELTRIG.SQL

This section shows the output for the EC2\_DELTRIG sql script and indicates the time in which the script begins. Each table that has information being deleted is displayed along with the criteria for

the deletion, in this case C\_CLIENTS\_SERVICES has information being deleted based on the client ID and the id.

```

START_TIME
-----
09/15/1999 07:59 PM
RUNNING EC2_IN_TAB.SQL
EC2 inserting :O_STAFF_MEMBERS
0 rows created.

EC2 inserting :O_ORGANIZATIONAL_UNITS
0 rows created.

```

This section shows the output for the EC2\_IN\_TAB.SQL script and indicates the date and time when the script begins. Each table that has data being inserted is listed along with the number of rows being created.

```

START_TIME
-----
09/15/1999 08:00 PM
RUNNING EC2_TAB_UP.SQL
EC2 updating : 0 O_STAFF_MEMBERS

PL/SQL procedure successfully completed.

EC2 updating : 1 O_STAFF_TIME_STUDIES

PL/SQL procedure successfully completed.

```

This section shows the output for the EC2\_TAB\_UP.SQL script and indicates the date and time when the script begins. Each table that is being updated with new values is listed along with the number of rows being updated.

```

START_TIME
-----
10/24/2003 08:37 PM

RUNNING EC2_DELTRIG_AU.SQL

PL/SQL procedure successfully completed.

```

FINISHED EC2\_DELTRIG\_AU.SQL  
This section shows the output for the EC2\_DELTRIG\_AU.SQL script and indicates the date and time when the script begins.

```

START_TIME
-----
10/24/2003 08:37 PM

RUNNING EC2_ST_CLIENTS_SUM.SQL

```

Before c\_state\_clients count:6129  
 Before c\_state\_clients null count:0  
 After c\_state\_clients count:6129  
 After c\_state\_clients null count:0

PL/SQL procedure successfully completed.

FINISHED EC2\_ST\_CLIENTS\_SUM.SQL

**This section shows the output for the EC2\_ST\_CLIENTS\_SUM.SQL script and indicates the date and time when the script begins. This End Of Day Script will populate data in c\_state\_clients**

**based on tables e\_c\_clients, e\_c\_family\_economic\_units and e\_c\_certifications.**

----- EOD IMPORT -----

Export file created by EXPORT:V07.03.03 via conventional path  
 . importing EODADM's objects into EODADM  
 . . importing table "E\_EOD\_CONTROLS" 1 rows imported  
 Import terminated successfully without warnings.

This section imports into the EOD controls base table and lists the number of rows that are imported.

----- EOD INSERT -----

EC4 inserting :A\_EOD\_CONTROLS  
 Table truncated.  
 1 row created.

This section indicates that values are being inserted into a specific table, whether the table is being truncated and how many rows are being created.

EC4\_STATE\_CLIENTS.SQL

----- EOD C\_STATE\_CLIENTS -----

old 17: where agency\_code = '&2'  
 new 17: where agency\_code = '25'  
 old 32: where agency\_code = '&2'  
 new 32: where agency\_code = '25'  
 old 79: where agency\_code = '&2';  
 new 79: where agency\_code = '25';

PL/SQL procedure successfully completed.

**This section shows the output for the EC4\_STATE\_CLIENTS.SQL script. This End Of Day Script will populate data in c\_state\_clients for today's day in a temporary table to catch up with the agencies on which EOD did not run on that day.**

----- EOD RETRIEVE -----

RUNNING EC4\_TRUNC\_ARCH\_CLI.SQL

START\_TIME

-----

10/24/2003 10:41 PM

table containing clients who have been retrieved from the archives

FINISHED EC4\_TRUNC\_ARCH\_CLI.SQL

RUNNING EC4\_INS\_RETR\_CLI.SQL

FINISHED EC4\_INS\_RETR\_CLI.SQL

RUNNING TRUNC\_VEN.SQL

FINISHED TRUNC\_VEN.SQL

RUNNING EC4\_INS\_RETR\_VEN.SQL

FINISHED EC4\_INS\_RETR\_VEN.SQL

This section logs the activity of the EOD Central retrieval scripts which were inserted from client and vendor tables into the temporary tables. This is the **01\_ec4\_retrieve.log** that is added at the end of EOD processing.

----- EXPORT -----

About to export specified tables via Conventional Path ...

.. exporting table A\_C\_TRANSFERS\_INFO 3 rows exported

.. exporting table A\_A\_ATTEND\_STATUSES 0 rows exported

.. exporting table A\_CTI\_C\_BLOODWORK\_DATA 16 rows exported

The export section is split into three columns. The first column indicates that the process is currently exporting a table, the second column shows the name of the table being exported, and the third column lists how many rows from the table that are being exported.

**%\_EC2\_AGENCY.LOG**- This file shows a list of all delete statements from the EC2\_DELTRIG.SQL script which deletes data from the central database that had been deleted from the local agency database since the last end of day run excluding c\_family\_economic\_units.

UPDATING EOD\_BY

1 row updated.

Commit complete.

START\_TIME

-----

11/23/1999 07:59 PM

RUNNING EC2\_DELTRIG.SQL

PL/SQL procedure successfully completed.

FINISHED EC2\_DELTRIG.SQL

RUNNING EC2\_IN\_TAB.SQL

EC2 inserting :O\_STAFF\_MEMBERS

PL/SQL procedure successfully completed.

EC2 inserting :O\_ORGANIZATIONAL\_UNITS

0 rows created.

```
EC2 inserting :O_ORG_PROGRAMS
0 rows created.
```

```
Commit complete.
FINISHED EC2_IN_TAB.SQL
```

```
RUNNING EC2_TAB_UP.SQL
EC2 updating :O_ORGANIZATIONAL_UNITS
0 rows updated.
EC2 updating :O_ORG_PROGRAMS
0 rows updated.
EC2 updating :O_ORG_UNIT_PHONES
0 rows updated.
```

```
RUNNING EC2_DELTRIG_AU.SQL
PL/SQL procedure successfully completed.
FINISHED EC2_DELTRIG_AU.SQL
```

**%\_EC4\_RETRIEVE.LOG** - This script details that the scripts concerning retrieving and storing requested archived data at the Central Database have run. This log is added to **agcy\_01.log** at the end of EOD processing.

**%\_EC4\_TAB.LOG** - This script details the `ec4_tab_%.sql` script that runs for each Local Agency. It is a log based on changes made to selected tables at the State Level, the temporary tables are utilized to facilitate data recovery efforts should they be needed. An example of the content of this file is as follows:

```
EC4 inserting :A_EOD_CONTROLS
Table truncated.
1 row created.
```

```
EC4 inserting :A_A_ATTEND_STATUSES
Table truncated.
0 rows created.
```

**EC4\_TRNC.LOG**- This script details the processing of the `EC4_TRNC.SQL` script which initializes tables for new data to be sent to the Local Agencies. An example of the content of this file is as follows:

```
truncating :E_EOD_CONTROLS
truncating :A_EOD_CONTROLS
truncating :A_CTI_%
```

### **Local Agency**

**EA.LOG** - This script details by date and time that outputs at the local agency have printed. An example of the content of this file is as follows:

```

10-01-2003 19:00:00 - EA1 STARTED
10-01-2003 19:00:01 - EA1.BAT EOD ALREADY RUN CHECK STARTED
10-01-2003 19:00:01 - EA1.BAT EOD ALREADY RUN CHECK FINISHED
10-01-2003 19:00:06 - EA1.BAT AIM APPLICATION FILE UPDATE STARTED
10-01-2003 19:00:46 - EA1.BAT AIM APPLICATION FILE UPDATE FINISHED
10-01-2003 19:00:56 - EA1.BAT SQL STARTED
10-01-2003 19:03:15 - EA1.BAT SQL FINISHED
10-01-2003 19:03:15 - EA1.BAT INSERT STARTED
10-01-2003 19:03:32 - EA1.BAT INSERT FINISHED
10-01-2003 19:03:32 - EA1.BAT INSERT STATE CLIENTS STARTED
10-01-2003 19:03:33 - EA1.BAT INSERT STATE CLIENTS FINISHED
10-01-2003 19:03:33 - EA1.BAT EXPORT STARTED
10-01-2003 19:03:37 - EA1.BAT EXPORT FINISHED
10-01-2003 19:03:38 - EA1.BAT LETTERS STARTED
10-01-2003 19:03:38 - EA1.BAT LETTERS FINISHED
10-01-2003 19:03:38 - EA1.BAT SENDING DUMP FILE TO CENTRAL
10-01-2003 19:03:46 - EA1.BAT DUMP FILE SENT TO CENTRAL
10-01-2003 19:03:46 - EA1.BAT EOD CONTROL STARTED
10-01-2003 19:03:46 - EA1.BAT EOD CONTROLS UPDATED
10-01-2003 19:03:46 - EA1.BAT EA1 FINISHED
10-02-2003 01:30:00 - EA3 STARTED
10-02-2003 01:30:01 - EA3.BAT COPYING DUMP FILE FROM CENTRAL
10-02-2003 01:30:26 - EA3.BAT DUMP FILE COPIED FROM CENTRAL
10-02-2003 01:31:01 - EA3.BAT AGENCY FOUND
10-02-2003 01:31:01 - EA3.BAT UNZIPPING FILE FROM CENTRAL
10-02-2003 01:31:01 - EA3.BAT UNZIP FINISHED
10-02-2003 01:31:35 - EA3.BAT AGENCY IMPORTED
10-02-2003 01:31:41 - EA3.BAT AGENCY DELETES PERFORMED
10-02-2003 01:31:55 - EA3.BAT AGENCY INSERTED
10-02-2003 01:36:21 - EA3.BAT AGENCY UPDATED
10-02-2003 01:36:21 - EA3.BAT SQL STARTED
10-02-2003 01:36:29 - EA3.BAT SQL FINISHED
10-02-2003 01:36:29 - EA3.BAT EA3_RETRIEVE STARTED
10-02-2003 01:36:35 - EA3.BAT EA3_RETRIEVE FINISHED
10-02-2003 01:36:35 - EA3.BAT STARTED DUAL ENROLLMENT REPORT
10-02-2003 01:36:40 - EA3.BAT DUAL ENROLLMENT REPORT FINISHED
10-02-2003 01:36:40 - EA PRINT STARTED
10-02-2003 01:36:40 - EA PRINT FINISHED
10-02-2003 01:36:40 - EA3.BAT DBA MONITORING STARTED
10-02-2003 01:36:49 - EA3.BAT DBA MONITORING FINISHED
10-02-2003 01:36:49 - EA3.BAT EA3 IS DONE

```

Column 1: The date the EOD script was executed (10) characters.

Column 2: The time (HH:MM:SS) that the EOD script executed the particular function of record.

Column 3: Explanation of what is being executed within the EOD script.

**EA\_HIS.LOG-** This file shows a history of the EA.LOG file described above.

```
10-01-2003 19:00:00 - EA1 STARTED
10-01-2003 19:00:01 - EA1.BAT EOD ALREADY RUN CHECK STARTED
10-01-2003 19:00:01 - EA1.BAT EOD ALREADY RUN CHECK FINISHED
10-01-2003 19:00:06 - EA1.BAT AIM APPLICATION FILE UPDATE STARTED
10-01-2003 19:00:46 - EA1.BAT AIM APPLICATION FILE UPDATE FINISHED
10-01-2003 19:00:56 - EA1.BAT SQL STARTED
10-01-2003 19:03:15 - EA1.BAT SQL FINISHED
10-01-2003 19:03:15 - EA1.BAT INSERT STARTED
10-01-2003 19:03:32 - EA1.BAT INSERT FINISHED
10-01-2003 19:03:32 - EA1.BAT INSERT STATE CLIENTS STARTED
10-01-2003 19:03:33 - EA1.BAT INSERT STATE CLIENTS FINISHED
10-01-2003 19:03:33 - EA1.BAT EXPORT STARTED
10-01-2003 19:03:37 - EA1.BAT EXPORT FINISHED
10-01-2003 19:03:38 - EA1.BAT LETTERS STARTED
10-01-2003 19:03:38 - EA1.BAT LETTERS FINISHED
10-01-2003 19:03:38 - EA1.BAT SENDING DUMP FILE TO CENTRAL
10-01-2003 19:03:46 - EA1.BAT DUMP FILE SENT TO CENTRAL
10-01-2003 19:03:46 - EA1.BAT EOD CONTROL STARTED
10-01-2003 19:03:46 - EA1.BAT EOD CONTROLS UPDATED
10-01-2003 19:03:46 - EA1.BAT EA1 FINISHED
10-02-2003 01:30:00 - EA3 STARTED
10-02-2003 01:30:01 - EA3.BAT COPYING DUMP FILE FROM CENTRAL
10-02-2003 01:30:26 - EA3.BAT DUMP FILE COPIED FROM CENTRAL
10-02-2003 01:31:01 - EA3.BAT AGENCY FOUND
10-02-2003 01:31:01 - EA3.BAT UNZIPPING FILE FROM CENTRAL
10-02-2003 01:31:01 - EA3.BAT UNZIP FINISHED
10-02-2003 01:31:35 - EA3.BAT AGENCY IMPORTED
10-02-2003 01:31:41 - EA3.BAT AGENCY DELETES PERFORMED
10-02-2003 01:31:55 - EA3.BAT AGENCY INSERTED
10-02-2003 01:36:21 - EA3.BAT AGENCY UPDATED
10-02-2003 01:36:21 - EA3.BAT SQL STARTED
10-02-2003 01:36:29 - EA3.BAT SQL FINISHED
10-02-2003 01:36:29 - EA3.BAT EA3_RETRIEVE STARTED
10-02-2003 01:36:35 - EA3.BAT EA3_RETRIEVE FINISHED
10-02-2003 01:36:35 - EA3.BAT STARTED DUAL ENROLLMENT REPORT
10-02-2003 01:36:40 - EA3.BAT DUAL ENROLLMENT REPORT FINISHED
10-02-2003 01:36:40 - EA PRINT STARTED
10-02-2003 01:36:40 - EA PRINT FINISHED
10-02-2003 01:36:40 - EA3.BAT DBA MONITORING STARTED
10-02-2003 01:36:49 - EA3.BAT DBA MONITORING FINISHED
10-02-2003 01:36:49 - EA3.BAT EA3 IS DONE
```

**EA1\_SQL.LOG-** This file shows a log for the following SQL scripts:

```

*****
RUNNING EA1_TERM_EOD.SQL
ea1_term_eod executed successfully.
FINISHED EA1_TERM_EOD.SQL
*****
*****
RUNNING EA1_CAT_EOD.SQL
FINISHED EA1_CAT_EOD.SQL
*****
*****
RUNNING ea1_cat_sync_eod.sql
Input truncated to 1 characters
FINISHED ea1_cat_sync_eod.sql
*****
*****
RUNNING EA1_APPT_EOD.SQL
FINISHED EA1_APPT_EOD.SQL
*****
*****
RUNNING EA1_NEW_CLASS_EOD.SQL
Input truncated to 1 characters
FINISHED EA1_NEW_CLASS_EOD.SQL
*****
*****
RUNNING EA1_NEW_APPT_EOD.SQL
Input truncated to 1 characters
FINISHED EA1_NEW_APPT_EOD.SQL
*****
*****
RUNNING EA1_PURGE_OLD_APPT.SQL
Input truncated to 1 characters
FINISHED EA1_PURGE_OLD_APPT.SQL
*****
*****
RUNNING EA1_PURGE_NEW_APPT.SQL
Input truncated to 1 characters
FINISHED EA1_PURGE_NEW_APPT.SQL
*****
*****
RUNNING EA1_OTHER_PURG.SQL
FINISHED EA1_OTHER_PURG.SQL
*****
*****
RUNNING EA1_TERM_WL_CLIENTS.SQL
Input truncated to 1 characters
FINISHED EA1_TERM_WL_CLIENTS.SQL
*****
*****
RUNNING EA1_FOOD_CHNG.SQL
START d:\aim\eod\EA1_FOOD_CHNG.SQL
FINISHED EA1_FOOD_CHNG.SQL

```

```

*****
*****
RUNNING EA1_IEN_ACTIVE.SQL
Input truncated to 1 characters
FINISHED EA1_IEN_ACTIVE.SQL
*****
*****
RUNNING EA1_TRANSFER.SQL
Input truncated to 1 characters
Transfers Successful: 0 Transfers Failed: 0
FINISHED EA1_TRANSFER.SQL
*****
*****
RUNNING EA1_STATE_CLIENTS_UPD.SQL
Input truncated to 1 characters
Updated c_state_clients successfully.
FINISHED EA1_STATE_CLIENTS_UPD.SQL
*****
*****
RUNNING EA1_CSF_OU_FIX.SQL
FINISHED EA1_CSF_OU_FIX.SQL
*****

```

**AGCY\_CTRL.LOG-** A list that details inserts of all new and updated data into the End of Day temporary tables. It also provides information regarding the export of these tables for preparing the zip file to be sent to the Central Server database. An example of the content of this file is as follows:

```

----- INSERT -----
EA1 inserting :E_EOD_CONTROLS
Table truncated.
1 row created.

EA1 inserting :E_O_STAFF_MEMBERS
Table truncated.
0 rows created.

```

The insert section shows which temporary base table is currently being updated with new values, whether the table is being truncated, and how many rows of new data are being created.

```

----- EXPORT -----
About to export specified tables via Conventional Path ...
.. exporting table      E_O_STAFF_MEMBERS    0 rows exported
.. exporting table      E_EOD_CONTROLS    1 rows exported
.. exporting table      E_I_FOOD_INSTRUMENT_TYPES  4 rows exported

```

The export section is split into three columns. The first column indicates that the process is currently exporting a table, the second column shows the name of the table being exported, and the third column lists how many rows from the table that are being exported.

**AGCY\_SQL.LOG** - A list that details participants who have been terminated or experienced a category change during the End of Day process. An example of the content of this file is as follows:

TO\_DATE FROM\_DATE GOOD\_PROC M Q LAST\_QTR\_IPL\_DATE

-----  
23-OCT-03 23-OCT-03 23-OCT-03 Y Y 31-MAR-00

UPDATING CONTROL DATE

UPDATING EOD\_BY

\*\*\*\*\*

RUNNING EA1\_TERM\_EOD.SQL

ea1\_term\_eod executed successfully.

FINISHED EA1\_TERM\_EOD.SQL

\*\*\*\*\*

\*\*\*\*\*

RUNNING EA1\_CAT\_EOD.SQL

FINISHED EA1\_CAT\_EOD.SQL

\*\*\*\*\*

\*\*\*\*\*

RUNNING ea1\_cat\_sync\_eod.sql

Input truncated to 1 characters

FINISHED ea1\_cat\_sync\_eod.sql

\*\*\*\*\*

\*\*\*\*\*

RUNNING EA1\_APPT\_EOD.SQL

FINISHED EA1\_APPT\_EOD.SQL

\*\*\*\*\*

\*\*\*\*\*

RUNNING EA1\_NEW\_CLASS\_EOD.SQL

Input truncated to 1 characters

FINISHED EA1\_NEW\_CLASS\_EOD.SQL

\*\*\*\*\*

\*\*\*\*\*

RUNNING EA1\_NEW\_APPT\_EOD.SQL

Input truncated to 1 characters

FINISHED EA1\_NEW\_APPT\_EOD.SQL

\*\*\*\*\*

\*\*\*\*\*

RUNNING EA1\_PURGE\_OLD\_APPT.SQL

Input truncated to 1 characters

FINISHED EA1\_PURGE\_OLD\_APPT.SQL

\*\*\*\*\*

\*\*\*\*\*

RUNNING EA1\_PURGE\_NEW\_APPT.SQL

```

Input truncated to 1 characters
FINISHED EA1_PURGE_NEW_APPT.SQL
*****
*****
RUNNING EA1_OTHER_PURG.SQL
FINISHED EA1_OTHER_PURG.SQL
*****
*****
RUNNING EA1_TERM_WL_CLIENTS.SQL
Input truncated to 1 characters
FINISHED EA1_TERM_WL_CLIENTS.SQL
*****
*****
RUNNING EA1_FOOD_CHNG.SQL
START d:\aim\eod\EA1_FOOD_CHNG.SQL
FINISHED EA1_FOOD_CHNG.SQL
*****
*****
RUNNING EA1_IEN_ACTIVE.SQL
Input truncated to 1 characters
FINISHED EA1_IEN_ACTIVE.SQL
*****
*****
RUNNING EA1_TRANSFER.SQL
Input truncated to 1 characters
Transfers Successful: 0 Transfers Failed: 0
FINISHED EA1_TRANSFER.SQL
*****
*****
RUNNING EA1_STATE_CLIENTS_UPD.SQL
Input truncated to 1 characters
Updated c_state_clients successfully.
FINISHED EA1_STATE_CLIENTS_UPD.SQL
*****
*****
RUNNING EA1_CSF_OU_FIX.SQL
FINISHED EA1_CSF_OU_FIX.SQL
*****
*****
UPDATING EOD_BY

*****
*****
RUNNING EA3_PROC_FI.SQL
Input truncated to 1 characters
FINISHED EA3_PROC_FI.SQL
*****
*****
*****
*****
RUNNING \agency\load\01_upd.sql

NONE_FAILED
-----

```

There are no failed FIs for this agency

1 row selected.

FINISHED \agency\load\01\_upd.sql

\*\*\*\*\*

\*\*\*\*\*

GENERATING NEW SESSION

Connected.

FINISHED GENERATING NEW SESSION

\*\*\*\*\*

\*\*\*\*\*

RUNNING EA3\_PRG\_ARCH.SQL

FINISHED EA3\_PRG\_ARCH.SQL

\*\*\*\*\*

RUNNING EA3\_INS\_RETR\_CLI.SQL

Input truncated to 1 characters

FINISHED EA3\_INS\_RETR\_CLI.SQL

RUNNING EA3\_INS\_RETR\_VEN.SQL

Input truncated to 1 characters

FINISHED EA3\_INS\_RETR\_VEN.SQL

**CTRL\_AGCY.LOG** - A list that details the transfer of base table, participant, and food instrument data from the Central Server Database to the Agency Server Database. An example of the content of this file is as follows:

```
----- TRUNCATE -----
truncating :A_EOD_CONTROLS
truncating :A_I_FOOD_INSTRUMENTS
truncating :A_A_ATTEND_STATUSES
truncating :A_A_SERVICES
truncating :A_C_BF_PATTERNS
```

The truncate section indicates that the base tables are currently being truncated for importing.

```
----- IMPORT -----
Export file created by EXPORT:V07.03.03 via conventional path
. importing EODADM's objects into EODADM
.. importing table      "A_C_TRANSFERS_INFO"    0 rows imported
.. importing table      "A_A_ATTEND_STATUSES"    0 rows imported
```

The import section is divided into three columns. The first indicates that a table is being imported, the second column shows which table is being imported, and the third shows how many rows of data are being imported.

```
----- INSERT -----
```

EA3 inserting :A\_ATTEND\_STATUSES  
0 rows created.

EA3 inserting :A\_SERVICES  
0 rows created.

The insert section indicates which base table is having information inserted and how many rows of information are being created.

----- UPDATE -----

EA3 updating :A\_ACTIVITIES

0 rows updated.

EA3 updating :A\_APPT\_ITEMS

0 rows updated.

The update section indicates which base table is having information updated and how many rows of information are being updated.

**AGCY\_%.LOG** - A historical log which details for the agency the data that was extracted from the agency's base tables and exported to the central server. An example of the content of this file is as follows:

----- TRUNCATE -----

truncating :E\_EOD\_CONTROLS  
truncating :E\_O\_STAFF\_MEMBERS

----- IMPORT -----

.. importing table	"E_O_STAFF_MEMBERS"	0 rows imported
.. importing table	"E_EOD_CONTROLS"	1 rows imported
.. importing table	"E_I_FOOD_INSTRUMENT_TYPES"	0 rows imported

RUNNING EC2\_UPD\_E\_C\_CLIENTS.SQL  
28 rows updated.  
FINISHED EC2\_UPD\_E\_C\_CLIENTS.SQL

START\_TIME

-----  
10/24/2003 07:40 PM

RUNNING EC2\_DELTRIG.SQL  
PL/SQL procedure successfully completed.  
FINISHED EC2\_DELTRIG.SQL

START\_TIME

-----  
10/24/2003 07:40 PM

----- INSERT -----

RUNNING EC2\_IN\_TAB.SQL  
EC2 inserting :O\_STAFF\_MEMBERS  
PL/SQL procedure successfully completed.  
EC2 inserting :O\_STAFF\_TIME\_STUDIES  
PL/SQL procedure successfully completed.

----- UPDATE -----

RUNNING EC2\_TAB\_UP.SQL  
EC2 updating : 0 O\_STAFF\_MEMBERS  
PL/SQL procedure successfully completed.

EC2 updating : 1 O\_STAFF\_TIME\_STUDIES  
PL/SQL procedure successfully completed.

FINISHED EC2\_TAB\_UP.SQL

START\_TIME

-----  
10/24/2003 07:40 PM

RUNNING EC2\_DELTRIG\_AU.SQL

PL/SQL procedure successfully completed.

FINISHED EC2\_DELTRIG\_AU.SQL

START\_TIME

-----  
10/24/2003 07:40 PM

RUNNING EC2\_ST\_CLIENTS\_SUM.SQL  
Before c\_state\_clients count:0  
Before c\_state\_clients null count:0  
After c\_state\_clients count:28  
After c\_state\_clients null count:0

PL/SQL procedure successfully completed.

FINISHED EC2\_ST\_CLIENTS\_SUM.SQL

----- EOD IMPORT -----

Connected to: Oracle8i Enterprise Edition Release 8.1.7.4.0 - 64bit Production  
With the Partitioning option  
JServer Release 8.1.7.4.0 - 64bit Production

Export file created by EXPORT:V08.01.07 via conventional path

```
import done in WE8ISO8859P1 character set and WE8ISO8859P1 NCHAR character set
. importing EODADM's objects into EODADM
.. importing table      "E_EOD_CONTROLS"      1 rows imported
Import terminated successfully without warnings.
```

```
----- EOD INSERT -----
```

```
EC4 inserting :A_EOD_CONTROLS
```

```
Table truncated.
```

```
1 row created.
```

```
EC4 inserting :A_A_ACTIVITIES
```

```
Table truncated.
```

```
0 rows created.
```

```
----- EOD C_STATE_CLIENTS -----
```

```
old 17:  where agency_code = '&2'
```

```
new 17:  where agency_code = '01'
```

```
old 32:  where agency_code = '&2'
```

```
new 32:  where agency_code = '01'
```

```
old 79:  where agency_code = '&2';
```

```
new 79:  where agency_code = '01';
```

```
PL/SQL procedure successfully completed.
```

```
----- EOD RETRIEVE -----
```

```
RUNNING EC4_TRUNC_ARCH_CLI.SQL
```

```
START_TIME
```

```
-----
```

```
10/24/2003 10:00 PM
```

```
table containing clients who have been retrieved from the archives
```

```
FINISHED EC4_TRUNC_ARCH_CLI.SQL
```

```
RUNNING EC4_INS_RETR_CLI.SQL
```

```
START_TIME
```

```
-----
```

```
10/24/2003 10:00 PM
```

```
Input truncated to 1 characters
```

```
FINISHED EC4_INS_RETR_CLI.SQL
```

```
RUNNING EC4_TRUNC_ARCH_VEN.SQL
```

```
START_TIME
```

```
-----
```

```
10/24/2003 10:00 PM
```

```
table containing vendors who have been retrieved from the archives
```

```
FINISHED EC4_TRUNC_ARCH_VEN.SQL
```

RUNNING EC4\_INS\_RETR\_VEN.SQL

START\_TIME

-----

10/24/2003 10:00 PM

FINISHED EC4\_INS\_RETR\_VEN.SQL

END\_TIME\_EC4\_RETRIE

-----

10/24/2003 10:00 PM

----- EXPORT -----

About to export specified tables via Conventional Path ...

.. exporting table	A_AAS_CLASS_CATEGORIES	0 rows exported
.. exporting table	A_AAS_ITEMS	0 rows exported
.. exporting table	A_F_POVERTY_BASES	0 rows exported
.. exporting table	A_STATE_CLIENTS	0 rows exported

The export section is divided into three columns. The first column shows that the process is currently exporting the tables to the central server, the second column indicates which base table is currently being exported, and the third column shows the number of rows that were exported to the central server.

### 3 Appendix B - FSMC Banking Interface

#### 3.1 FSMC Banking Interface to the AIM System

##### 3.1.1 Overview

The AIM System compiles vendor and food instrument pricing information during the End of Day process at the Central. During the compilation process, the system processes the information and consolidates it into 3 separate files with a **.dat** format. The following files are created:

**isan(mmdd).dat (Created daily)**  
**anpga.dat (Created at the end month)**  
**anvencor.dat \* (Created daily)**

##### 3.1.2 Issuance Information

( tables are: i\_bank\_reports and o\_organizational\_units)

EOD Script is “ec2\_issue\_fsmc.sql”

The first file: **isan(mmdd).dat**, is a file created nightly during the End of Day processing which contains Food Instrument issuance information, has a variable name. The mmdd part of the file name changes based upon the month and day the file is being created. For example, a file for food issuance information entered into the AIM System on September 1<sup>st</sup> will have the name: isaz0901.dat.

An example of the content of this file is as follows:

```
0000729592002400SS**40 000000000008261999D 0925199910251999030000000000000000
0000729593002871SS**40 000000000008261999VB 0831199909301999030000000000000000
0000729594002414SS**40 000000000008261999VB 0831199909301999030000000000000000
0000729595002415SS**40 000000000008261999A 0831199909301999030000000000000000
0000729595002415SS**40 000000000008261999O 0831199909301999030000000000000000
```

Record Layout for Issuance file:

Detail Record (isanmmdd.dat)

Check Serial Number	num (10),	(Aim has 12 digits. Highest number is 0096664470. would take the state 900 years to reach the end of ten digits.)
FI Type Code	num (6),	(the first 6 digits of type code. NOT

		used by FSMC.
FI Type Code	char (2),	(Last two characters of type code.)
Filler	char (2),	(Pre-filled with “**”.)
Organization ID	char (2),	(org code. FSMC does not use.)
Filler	char (2),	(Pre-filled with spaces.)
Filler	num (10),	(Pre-filled with 0’s. FSMC calls this the Client Id field, but is not used.)
Issue Date	num (8),	(Issue date mmddyyyy format. FSMC does NOT use.)
Issue Method	char (1),	(D=Issued on-demand or batch, C=Issued compliance buy, A=Issued replaced or reissued, S=Void stolen, V=all other voids, O=Rejected Override, “=all other types.)
Void code	char (2),	(void reason code. “ “ are spacers.)
First Date to Use	num (8),	(mmddyyyy format.)
Last Date to Use	num (8),	(mmddyyyy format.)
Max Amount	num (6),	(9(4)v99 format. FSMC will not use.)
Filler	char (11)	(Pre-filled with 0’s.)
Total Record Length = 78		

Trailer Record (isanmmdd.dat)

1.	Filler	char(5),	(Pre-filled with “TOTAL”)
2.	Count	num (6)	(Total record count excluding trailer record)
Total Record Length = 11			

The AIM System compiles this information into a file and sends it via File Transfer Protocol (FTP) to FSMC for insertion into their database.

3.1.3 Peer Group Average Information

(v\_peer\_group\_avgs table)

EOD Script is “ec1\_pga\_fsmc.sql”

The second file created: **anpgaver.dat** is a file created at the end of month which combines the peer group id, FC type code and maximum redemption amount for food instruments within a peer group. File is based on monthly peer group averages.

An example of the content of this file is as follows:

```
04000305SS**001075005500
03000326SS**005400005940
03000327SS**006300006993
03000844SS**005860006446
```

**Record Layout for Peer Group Average file:****Detail Record (anpgaver.dat)**

**Peer Group ID** number(2), (peer\_group\_id)  
**FI Type Code** char (6), (lpad with 0's, FSMC doesn't use)  
**FI Type Code** char (2), (last two characters of fi\_type\_code)  
**filler** char (2), (pre-filled with "\*\*\*")  
**Average Redemption Amt** num (6,2), (FSMC doesn't use .. pre-filled with 0's)  
**Maximum Redemption Amt** num (6,2) (format is 9(4)v99)  
**Total Record Length = 24**

**Trailer Record (anpgaver.dat)**

1. **Filler** char(5), (Pre-filled with "TOTAL")  
 2. **Count** num (6) (Total record count excluding trailer record)  
**Total Record Length = 11**

## 3.1.4 Vendor Information

( uses the following tables ... v\_vendors, v\_authorizations, s\_geo\_locations, v\_disqualifications, v\_vendor\_accounts, and v\_bank\_branches)

EOD Script is "ec1\_vendor\_fsmc.sql"

The third file created: **anvendor.dat** is a file created nightly which compiles all vendor demographic information.

An example of the content of this file is as follows:

```

00111FOODLAND SUPER MARKET, LTD. #32 55-510 KAMEHAMEHA HIGHWAY LAIE HI967620000
0300000000AMERICAN SAVINGS BANK 100000001111
00115FOODLAND SUPER MARKET, LTD., #27 59-720 KAMEHAMEHA HIGHWAY HALEIWA HI967120000
0300000000AMERICAN SAVINGS BANK 1000000011222
00108SACK 'N SAVE #34 94-887 LUMIANINA STREET WAIPAHU HI967970000
0300000000BANK OF HAWAII 100000003125
  
```

**Record Layout for Vendor file:****Detail Record (anvendor.dat)**

1. **Vendor ID** num (5), (lpad with leading 0 as ours is 4 digits)  
 2. **Vendor Name** char (35), (vendor name)  
 3. **Vendor Addr1** char (35), (vendor mailing address 1)  
 4. **Vendor Addr2** char (35), (vendor mailing address 2)  
 5. **City** char (35), (vendor city)  
 6. **State** char (2), (vendor state)  
 7. **Zip5** num (9), (vendor zip code)  
 8. **Owner/Corp Code** char (4), (Pre-fill with spaces, not used)

9. <b>Peer Group ID</b>	num(2),	(pee_peer_group_id)
10. <b>Termination Date</b>	num (8),	(mmddyyyy format term or disqualification date)
11. <b>Bank Name</b>	char (30),	(bank name)
12. <b>Bank ABA</b>	num (9),	(bank routing number)
13. <b>Bank Account</b>	char (10),	(bank vendor account number)
14. <b>Filler</b>	char (7)	(pre-fill with spaces)
<b>Total Record Length = 226</b>		

**NOTES:**

1. Termination Date – If a vendors status changed to disqualified/terminated now or in the future, put that date in. If the vendor is authorized fill the field with 0's. The bank will update the field as follows:  
     Authorized = Termination date is = 0 or > current date;  
     Disqualified/Terminated = Termination date is = current date or < current Date.
2. A complete vendor file is sent each night.
3. Bank name, vendor name, address1, address2, city, state and zip are not used by FSMC and therefore have been shortened.

**Trailer Record (anvendor.dat)**

1. **Filler** char(5), (Pre-filled with "TOTAL")
2. **Count** num (6) (Total record count excluding trailer record)

**Total Record Length = 11**

After the AIM system compiles its 3 files of issuance, peer group average and vendor information and sends them to the bank via FTP processing, the bank then returns a file to the AIM system consisting of paid/rejected check information. For the purpose of discussing functionality, we'll refer to the file returned by the bank as: **pdanmmdd.dat** where the mmdd part of the file name is variable depending upon the paid/rejected date of the check information included within the file.

An example of the content of this file is as follows:

```
00861720110231104012003PX 003317000000
00374149000014408181999PY 003339000000
00374435000022908181999R E000000003210
00374604000014608181999SW 001948000000
00375221000014008181999R C000000006455
```

**Record Layout for Paid Redemption file:****Detail Record (pdanmmdd.dat)**

1. **Filler** num (2), (pre-filled with 0's)
2. **FI serial Number** num (8), (AIM highest FI is currently 0096664470. If Only 8 returned would take 3 months to go Over. **NEED TO UPDATE ASAP \*\*\*\*\***)
3. **Filler** num (1), (Pre-filled with 0.)
4. **Vendor Number** num (4), (vendor id.)

- 5. **Process Date** num (8), (mmddyyyy format. Date bank processed the FI's.)
  - 6. **Status** char (1), (**P**=Paid; **R**=Rejected; **S**=Missing Issuance; **M**=Manual Adjustment)
  - 7. **Accept Code** char (1), (**X**=Paid-Normal; **Y**=Paid FSMC Override/CB; **Z**=Paid Arizona Override/2<sup>nd</sup> Lead; **W**=Paid Missing Issuance)
  - 8. **Reject Code** char (1), ( **A**=Altered/Void, **B**=Date of Use/Redeemed Early, **C**=Date of Use Invalid, **D**=Missing Clinic Signature, **E**=Invalid Vendor Number **F**=Deposited Too Late,**G**=Altered Dollar Amount **H**=Over Maximum Dollar Amount, **I**=Exceed Amount for FI Type, **J**=Date of Use Missing, **K**=Stop Payment Issued, **L**=FI Number Invalid/Void, **M**=Previously Rejected, **N**=Missing Vendor Number, **O**=Rejected For Replacement, **P**=Missing Countersignature, **Q**=Mismatched Signature, **R**=Rejected For Replacement, **U**=Unreadable Vendor Number)
  - 9. **Paid Amount** num (6), (9(4)v99 format.)
  - 10. **Rejected Amount** num (6)(9(4)v99 format.)
- TOTAL Record Length = 38**

**NOTE (1):** The following are the valid combinations for Status, Accept Code, and Rejection Code:

3.1.4.1.1 STATUS	ACCEPT CODE	REJECTION CODE
R	Leave blank	any rejection code
P	X,Y,Z	Leave blank
M	X,Y,Z	Leave blank
S	W	Leave blank

The end summary (not shown) needs to be updated to display two separate totals (Total Paid, Total Rejected)

**Trailer Record (pdanmmdd.dat)**

- 1. **Filler** char(5), (Pre-filled with "TOTAL")
  - 2. **Count** num (9), (Total record count excluding trailer record)
  - 3. **Paid Amt** num (11) (Total paid amount, 9(9)v99 format.)
- Total Record Length = 11**

## 4 Appendix C – Start Of Month Caseload Process

## 4.1 **Monthly population of participation information for clients**

### 4.1.1 Overview

The AIM System compiles the caseload information for the clients that were issued the food instruments in the past and for those whose certifications were created in the past months during the start of the month process at the Central. During the compilation process, the system processes the information and inserts it in caseload tables. The following tables are updated:

**F\_caseload\_type\_details**  
**F\_caseloads**

### 4.1.2 Participation Information

(Tables are f\_caseloads and f\_caseload\_type\_details)

EOD Script is “insert\_monthly\_caseloads.sql”

The key fields in f\_caseload\_type\_details table are as follows :

Cc\_Client\_Id - Client\_id from c\_clients  
 Ou\_Seq\_Id – the agency/clinic they are enrolled in from c\_family\_economic\_units table  
 Fbu\_Ffy\_Month – Fiscal month  
 Fbu\_Fff\_Ffy – Fiscal Year  
 Cat\_Category\_Code – Category code from c\_certifications table  
 Ceg\_Ethnic\_Group\_Code – Ethnic Group code from c\_clients  
 Lang\_Language\_Code – Primary language code from c\_family\_economic\_units table  
 Cp2\_Id – Priority from c\_certifications table  
 Program – The Program under which they are certified i.e. ‘WIC’  
 State\_Fund\_Flag – State fund flag from certifications table  
 Date\_Created – Date on which caseload record is created  
 Created\_By – Database User who created this record  
 Fpl\_Id – Determine fpl\_id based on the family’s income from f\_poverties\_to\_incomes table  
 Certified\_Flag – This flag will be ‘Y’ i.e. Yes if the client is enrolled  
 Participant\_Flag - This flag will be ‘Y’ i.e. Yes if the client is a participant  
 Migrant\_Flag - This flag will be ‘Y’ i.e. Yes if the client is Migrant  
 Adj\_Eligible\_Flag – Client’s adjunct eligibility flag  
 Refugee\_Flag - This flag will be ‘Y’ i.e. Yes if the client is Refugee  
 Income – Family’s Recent Income in that fiscal month  
 Family\_Size – Total number of family members from c\_family\_economic\_units table  
 Date\_Modified - Date on which caseload record is last modified  
 Modified\_By - Database User who modified this record last

The Rollup table from type details table is f\_caseloads table , it has the sum of all the caseload client in the client\_count field grouped by category\_code, ethnic\_group\_code, agency/clinic, language, povery\_id, fiscal\_month, fiscal\_year, priority, program. The structure of this table is as follows :

Caseload\_Id – Unique identifier  
 Cat\_Category\_Code – Category code  
 Ceg\_Ethnic\_Group\_Code – Ethnic Group Code  
 Ou\_Seq\_Id – Agency/Clinic  
 Lang\_Language\_Code – Language Code  
 Fpl\_Id – Poverty Id  
 Fbu\_Ffy\_Month – Fiscal Month  
 Fbu\_Fff\_Ffy – Fiscal Year  
 Fc\_Caseload\_Type\_Code – The type code will determine if the record gives total enrollee, participant, migrant or refugee count (1-Participant, 2- Enrollee, 3-Migrant, 4-Refugee, 5-Migrant Participant, 6-Refugee Participant)  
 Cp2\_Id - Priority  
 Client\_Count – Total count of clients  
 Program - WIC  
 State\_Fund\_Flag – State fund flag from certifications table  
 Date\_Created – Date on which caseload record is created  
 Created\_By – Database User who created this record  
 Bf\_Inf\_Count – Never used  
 Date\_Modified - Date on which caseload record is last modified  
 Modified\_By - Database User who modified this record last

The logic to populate caseloads table:

1. Enrollment Logic:

- The client will be considered as enrollee if:
  - The client is certified for least one day in the month:  
 So If the client's cert\_start\_date <= last day of month and  
 Client's cert\_end\_date >= first day of month and  
 (1.4) first day of month – cert\_start\_date <= 60 and  
 Cert\_Termination\_date is null or >= first day of month  
 Client is certified under the WIC program.
- A no-show is a client who is certified and has not participated (this is not related to missing appointments). As per USDA, a client who is no-show for 2 or more months may be removed from the enrollment count. After two months of no-shows the client should not be counted in the enrollee total. We will not change the certification table but make his change on the report side. This creates a distinction between “Certification” and “Enrollment”. Certification is what is in the Certification table, and Enrollment is a subset of Certification, which excludes no-shows. Also, Certification occurs by the day and Enrollment occurs by the month.
- No-Show = certified but not participating for 60 days 4(Not really participation because participation is by the month. The client is certified by does not receive a valid FI for 60 days.) Out of state transfers can't be checked for no show clients since AIM will not have the history of food instruments received from other states.
- If a client is transferred within a certificated month, then the client is counted as enrolled in the last clinic they are certified in for that month.
- For the category changed clients consider original cert start date.

- If the client doesn't have a valid income, then check for the dual enrollment in c\_resolutions table, if a record is found in the dual enrollment table then consider it as a dual enrollment and don't report the client for caseloads.

2. Participation Logic:

**The client should be considered as a participant if:**

- The client has received Food instruments for the reporting month and at least one food instrument is not voided.

OR

- If all the food instruments for the client in the reporting month are voided, but
  - a. There exists at least one FI with one of the following void reasons (No matter if the void date is before, equal to, or after first date to use)

- B - Lost Or Stolen
  - E - Returned By Client/Unused
  - Z - Stale Dated By EOD
  - L - Signed Before Going To Store

OR

- b. There exists at least one food instrument voided after the first date to use with one of the following void reasons

- D - Replaced/Food Package Change
    - H - Termination
    - J - Late Pickup/Reduced Food Package

**The client should be considered as a Non-participant if:**

**A.**

If the client has all the Food Instruments voided for the reporting month with one of the following reasons; and there is no second reissued set of food instruments.

- C - No-Show/Unclaimed
- A - Damaged, Unissued
- F - Printed/Not Issued
- G - Misprinted Text
- I - Categorical Ineligibility

**B.**

- If a participant is breastfeeding an infant (Infant category IEN or IPN), then that infant is also a WIC participant.
- If the breastfeeding infant's mother is a participant as per above logic then count the IPN/IEN as a participant too.
- If there is no linked mother for the IEN or if the mother is not a participant, then check if IEN has been assigned a dummy food package 'AA900888' else use the above logic for the infant to determine their participation.

**C.**

If a client transfers, then the client is counted as a participant in the last clinic they participated in for the month in question.

**Income-related Case Exceptions for Participation and Enrollment**

- All exceptions should be reported as an "Other" category. Any data problems in report fields such as the client's race/ethnicity, poverty level, family size, and income eligibility should be reported as "Other". Since the family\_size and income defines the poverty level, creation of a new "Other" category for just the poverty level should handle the exceptional cases for income or family size too. We will create a new "Other" category for race ethnicity too in order to handle undefined race ethnicity cases. None of these report fields can exclude a client from participation; however participants should not have exceptions in required fields. If exceptions occur they need to be documented and investigated.
- If the client's income changes to income ineligible, then the client should return all the remaining FIs and should be terminated from certification. As a practical matter, the client may retain valid FIs for the current month and up to the next two following months.
- Currently AIM doesn't have automatic termination facility because the client can have valid food instruments issued, so those FI's need to be returned by client and need to manually void from AIM and terminate the client. But if the client doesn't return unused FI's then the client will still be a participant.
- If a client's certification is terminated but they still have valid FIs for future months, then the client will be a participant but not an enrollee. This should be very rare.
- For the Out of State transfers, since the income entry is not mandatory, it should be reported under "Other Income" category. The report should have comments indicating the possible reasons for any client in an "Other" category.

If a valid income record is not found between the certification start date and certification end date for the client then the income "Other" category will be used to report participation.

**4.2 Monthly population of participation information for farmer market****4.2.1 Overview**

The AIM System compiles the caseload participation information for the clients that were redeemed the coupons in the past and for those whose coupons were created in the past months. During the compilation process, the system processes the information and inserts it in caseload tables. The following tables are updated:

**M\_caseload\_type\_details**  
**f\_caseloads**

#### 4.2.2 Participation Information

(Tables are f\_caseloads and m\_caseload\_type\_details)

EOD Script is “ec2\_populate\_clinic.sql”

The key fields in m\_caseload\_type\_details table are as follows :

Cc\_Client\_Id - Client\_id from c\_clients  
 Ou\_Seq\_Id – the agency/clinic they are enrolled in from c\_family\_economic\_units table  
 Fbu\_Ffy\_Month – Fiscal month  
 Fbu\_Fff\_Ffy – Fiscal Year  
 Cat\_Category\_Code – Category code from c\_certifications table  
 Ceg\_Ethnic\_Group\_Code – Ethnic Group code from c\_clients  
 Lang\_Language\_Code – Primary language code from c\_family\_economic\_units table  
 Cp2\_Id – Priority from c\_certifications table  
 Program – The Program under which they are certified i.e. ‘WIC’  
 State\_Fund\_Flag – State fund flag from certifications table  
 Date\_Created – Date on which caseload record is created  
 Created\_By – Database User who created this record  
 Fpl\_Id – Determine fpl\_id based on the family’s income from f\_poverties\_to\_incomes table  
 Certified\_Flag – This flag will be ‘Y’ i.e. Yes if the client is enrolled  
 Participant\_Flag - This flag will be ‘Y’ i.e. Yes if the client is a participant  
 Migrant\_Flag - This flag will be ‘Y’ i.e. Yes if the client is Migrant  
 Adj\_Eligible\_Flag – Client’s adjunct eligibility flag  
 Refugee\_Flag - This flag will be ‘Y’ i.e. Yes if the client is Refugee  
 Income – Family’s Recent Income in that fiscal month  
 Family\_Size – Total number of family members from c\_family\_economic\_units table  
 Date\_Modified - Date on which caseload record is last modified  
 Modified\_By - Database User who modified this record last

The Rollup table from type details table is f\_caseloads table , it has the sum of all the caseload client in the client\_count field grouped by category\_code, ethnic\_group\_code, agency/clinic, language, poverty\_id, fiscal\_month, fiscal\_year, priority, program. The structure of this table is as follows :

Caseload\_Id – Unique identifier  
 Cat\_Category\_Code – Category code  
 Ceg\_Ethnic\_Group\_Code – Ethnic Group Code  
 Ou\_Seq\_Id – Agency/Clinic  
 Lang\_Language\_Code – Language Code  
 Fpl\_Id – Poverty Id

Fbu\_Ffy\_Month – Fiscal Month  
 Fbu\_Fff\_Ffy – Fiscal Year  
 Fc\_Caseload\_Type\_Code – The type code will determine if the record gives total enrollee, participant, migrant or refugee count  
 (8- farmer market Participant, 7- Farmers Market Enrollee)  
 Cp2\_Id - Priority  
 Client\_Count – Total count of clients  
 Program - WIC  
 State\_Fund\_Flag – State fund flag from certifications table  
 Date\_Created – Date on which caseload record is created  
 Created\_By – Database User who created this record  
 Bf\_Inf\_Count – Never used  
 Date\_Modified - Date on which caseload record is last modified  
 Modified\_By - Database User who modified this record last

The logic to populate caseloads table:

A new procedure will be added in the caseload package to take care of participation for farmers' market client which will use m\_caseload\_type\_details and f\_caseloads tables.

Caseload. Insert\_caseload\_type\_farmers:

This caseload procedure will run during EOD process and will mark the clients who received the coupons for that day as enrolled clients for farmers market. Type code is “7”.

Caseload. Update\_caseload\_type\_farmers:

This caseload procedure will run when we receive a paid file for farmers market from FSMC, and it will mark the enrolled clients as the participant for the month in which their first coupon is redeemed. Type code is “8”.

E.g.: If client C1 has been issued the coupons in January, but found a redeemed coupon for the first time in March then client will be considered as a participant for March. If another coupon is redeemed in April, the client is not counted again as a participant.

### **4.3 Monthly population of participation information for CSFP clients**

#### 4.3.1 Overview

The AIM System compiles the caseload participation information for the clients that were redeemed the coupons in the past and for those whose coupons were created in the past months. During the compilation process, the system processes the information and inserts it in caseload tables. The following tables are updated:

**f\_caseload\_type\_details\_csfp**  
**f\_caseloads\_csfp**

### 4.3.2 Process

As a CSFP client is enrolled into AIM, EOD will create a record in the F\_Caseload\_Type\_Details\_csfp table and flag the client as CSFP. EOD will set the participant flag to 'Y' if a food box was issued. There will be a monthly process to enroll the csfp clients for that month in caseload tables.

## 5 Appendix D – Transfer Process

### 5.1 Processing Transfers during EOD

This database procedure is used to transfer the client from one agency/clinic to another agency/clinic. The clinic to clinic transfer could be in the same agency or in different agencies. The database procedure could get invoke from the transfers screen or from the EOD process.

There are two main database procedures in the transfer process, one procedure resides on the state database and another resides on the agency database. The transfer request is send through the database package residing on the to agency and which in turn calls a database package procedure residing on state agency to co-ordinate with the from agency.

The transfer request from the to\_agency asks the from\_agency to put all the client data in the temperory tables through the state database, this temperory data ultimately gets copied to the to\_agency finally.

#### 5.1.1 Transfer\_la\_pkg :

1. PROCEDURE fill\_transfer\_tables : Procedure to select from transfer tables and create data in new local agency.
2. PROCEDURE insert\_clients : Procedure to select from transfer tables and create data in new local agency.
3. PROCEDURE clean\_up\_la : Procedure to update losing local agency and then delete temporary information.
4. PROCEDURE terminate\_client : Procedure to terminate the client in losing local agency.
5. PROCEDURE delete\_future\_appts : Procedure to delete future appointments
6. PROCEDURE transfer\_in\_la: Procedure to perform transfers within a local agency
7. FUNCTION initiate\_transfer : Function to execute the transfer

#### 5.1.2 Transfer\_state\_pkg :

1. PROCEDURE fill\_central\_tables : Procedure to copy the transfer table data from the losing local agency to central.
2. PROCEDURE delete\_t\_data: Procedure to delete data from transfer tables
3. PROCEDURE transfer\_between\_las : Procedure to perform transfers between local agencies
4. PROCEDURE clean\_up\_central\_and\_la : Procedure to update losing local agency and then delete temporary information.

#### 5.1.3 Here is the sequence of the transfer procedure between two different agencies:

To\_agency : 1. transfer\_la\_pkg.initiate\_transfer : 1. Checks if the laptop clinic has been checked out , if so it errors out with the message , else proceeds  
 State: 2. transfer\_state\_pkg.transfer\_between\_las

From\_agency: 3. transfer\_la\_pkg.fill\_transfer\_tables  
 State 4. Transfer\_State\_Pkg.fill\_central\_tables  
 To\_agency 5. transfer\_la\_pkg.insert\_clients  
 State 6. transfer\_state\_pkg.clean\_up\_central\_and\_la  
 From\_agency 7. transfer\_la\_pkg.clean\_up\_la  
 State 8. Transfer\_State\_Pkg.delete\_t\_data  
 From\_agency 9. transfer\_la\_pkg.terminate\_client  
 From\_agency 10. transfer\_la\_pkg.delete\_future\_appts  
 State 11. transfer\_state\_pkg.delete\_t\_data

**5.1.4 Here is the sequence of the transfer procedure between two different clinics in the same agency:**

To\_agency : 1. transfer\_la\_pkg.initiate\_transfer : 1. Checks if the laptop clinic has been checked out , if so it errors out with the message , else proceeds  
 To\_agency : 2. transfer\_la\_pkg.transfer\_in\_la  
 To\_agency 3. transfer\_la\_pkg.delete\_future\_appts

**5.2 Transfer\_la\_pkg :**

**5.2.1. PROCEDURE fill\_transfer\_tables :**

Procedure to select from transfer tables and create data in new local agency.

Input : c\_transfers\_info.transfer\_id  
 c\_clients.client\_id  
 c\_family\_economic\_units.family\_id

Processing : 1. Procedure to fill transfer tables with data from the losing local agency.  
 2. Delete any data in the client and family transfer temporary tables for this transfer. There may be data if the transfer was not successful in a previous attempt.  
 3. Insert the data in the following tables :

t\_c\_clients  
 t\_c\_family\_economic\_units  
 t\_i\_bf\_promo\_vchrs  
 t\_i\_bf\_vchr\_items  
 T\_C\_CLIENT\_RACE\_ETHNICITIES  
 t\_c\_more\_ethnic\_groups  
 t\_c\_prev\_names  
 t\_c\_resolutions  
 t\_c\_certifications  
 t\_c\_cert\_term\_reasons  
 T\_C\_CLIENTS\_NOT\_LINKED  
 t\_c\_cert\_nutr\_ed  
 t\_c\_health\_risk\_factors  
 t\_c\_food\_package\_prescriptions  
 t\_c\_food\_box\_dists  
 t\_c\_dietary\_assessments  
 t\_c\_diet\_nutrients  
 t\_c\_bloodwork\_data

t\_c\_bf\_promo\_issuances  
t\_c\_allergy\_foods  
t\_c\_woman\_medicals  
t\_c\_transfer\_histories  
t\_c\_i\_c\_healths  
t\_c\_i\_c\_hh\_reasons\_bfends  
T\_C\_BF\_SURVEILLANCE  
t\_c\_infant\_data  
t\_c\_infant\_child\_medicals  
t\_c\_income\_histories  
t\_c\_w\_healths  
t\_c\_w\_hh\_reasons\_bfends  
t\_c\_client\_services  
t\_c\_client\_ne\_topics  
t\_c\_client\_svc\_ne\_materials  
t\_c\_ps\_responses  
t\_c\_prev\_families  
t\_c\_client\_bp\_issuances  
t\_c\_cert\_peer\_counsels  
t\_c\_peer\_rbfends  
t\_c\_contacts  
t\_c\_p\_couns\_notes  
t\_c\_client\_referrals  
t\_c\_client\_progs  
t\_c\_client\_notes  
t\_c\_client\_goals  
t\_c\_client\_communications  
t\_c\_resolved\_clients  
t\_i\_food\_packages  
t\_i\_food\_package\_foods  
t\_i\_package\_distributions  
t\_i\_food\_package\_fi\_types  
t\_i\_food\_instrument\_types  
t\_i\_food\_instrument\_foods  
t\_c\_income\_histories  
t\_c\_economic\_unit\_members  
t\_c\_incomes  
t\_c\_feu\_communications  
t\_c\_family\_phones  
t\_c\_smoking\_histories  
t\_c\_fam\_referrals

Output : message-If its null then process is successful, else the message will be the actual error message

### 5.2.2. PROCEDURE insert\_clients :

Procedure to select from transfer tables and create data in new local agency.

Input : c\_transfers\_info.transfer\_id  
c\_clients.client\_id  
c\_family\_economic\_units.family\_id

Processing: This procedure inserts the clients and family data from state databases transfer temporary tables in the aim owned tables on the to\_agency database. It inserts the data in the following tables. If the client data already exist in the table then it first deletes the data and inserts the fresh data.

- c\_clients
- c\_family\_economic\_units
- i\_bf\_promo\_vchrs
- i\_bf\_vchr\_items
- c\_CLIENT\_RACE\_ETHNICITIES
- c\_more\_ethnic\_groups
- c\_prev\_names
- c\_resolutions
- c\_certifications
- c\_cert\_term\_reasons
- c\_CLIENTS\_NOT\_LINKED
- c\_cert\_nutr\_eds
- c\_health\_risk\_factors
- c\_food\_package\_prescriptions
- c\_food\_box\_dists
- c\_dietary\_assessments
- c\_diet\_nutrients
- c\_bloodwork\_data
- c\_bf\_promo\_issuances
- c\_allergy\_foods
- c\_woman\_medicals
- c\_transfer\_histories
- c\_i\_c\_healths
- c\_i\_c\_hh\_reasons\_bfends
- c\_BF\_SURVEILLANCE
- c\_infant\_data
- c\_infant\_child\_medicals
- c\_income\_histories
- c\_w\_healths
- c\_w\_hh\_reasons\_bfends
- c\_client\_services
- c\_client\_ne\_topics
- c\_client\_svc\_ne\_materials
- c\_ps\_responses
- c\_prev\_families
- c\_client\_bp\_issuances
- c\_cert\_peer\_counsels
- c\_peer\_rbfends
- c\_contacts
- c\_p\_couns\_notes
- c\_client\_referrals
- c\_client\_progs
- c\_client\_notes
- c\_client\_goals
- c\_client\_communications
- c\_resolved\_clients

i\_food\_packages  
 i\_food\_package\_foods  
 i\_package\_distributions  
 i\_food\_package\_fi\_types  
 i\_food\_instrument\_types  
 i\_food\_instrument\_foods  
 c\_income\_histories  
 c\_economic\_unit\_members  
 c\_incomes  
 c\_feu\_communications  
 c\_family\_phones  
 c\_smoking\_histories  
 c\_fam\_referrals

Update c\_state\_clients to reflect proper agency/clinic for the transferred client.

Output :p\_transfer\_date

p\_ou\_seq\_id o\_organizational\_units.seq\_id%TYPE,  
 p\_sgeo\_geo\_location\_id s\_geo\_locations.geo\_location\_id%TYPE,  
 p\_new\_family\_id c\_family\_economic\_units.family\_id%TYPE,  
 p\_new\_mailing\_address1 c\_family\_economic\_units.mailing\_address1%TYPE,  
 p\_new\_mailing\_address2 c\_family\_economic\_units.mailing\_address2%TYPE,  
 p\_new\_mailing\_zip4 c\_family\_economic\_units.mailing\_zip4%TYPE,  
 p\_authorized\_rep1\_last\_name  
 c\_family\_economic\_units.authorized\_rep1\_last\_name%TYPE,  
 p\_authorized\_rep1\_first\_name  
 c\_family\_economic\_units.authorized\_rep1\_first\_name%TYPE,  
 p\_authorized\_rep1\_mi1 c\_family\_economic\_units.authorized\_rep1\_mi1%TYPE,  
 p\_authorized\_rep1\_mi2 c\_family\_economic\_units.authorized\_rep1\_mi2%TYPE,  
 p\_message-If its null then process is successful, else the message will be the actual error  
 message

### 5.2.3. PROCEDURE clean\_up\_la:

Procedure to update losing local agency and then delete temporary information

Input :p\_transfer\_id IN c\_transfers\_info.transfer\_id%TYPE  
 p\_transfer\_date IN DATE,  
 p\_ou\_seq\_id IN o\_organizational\_units.seq\_id%TYPE,  
 p\_sgeo\_geo\_location\_id IN s\_geo\_locations.geo\_location\_id%TYPE,  
 p\_new\_family\_id IN c\_family\_economic\_units.family\_id%TYPE,  
 p\_new\_mailing\_address1 IN c\_family\_economic\_units.mailing\_address1%TYPE,  
 p\_new\_mailing\_address2 IN c\_family\_economic\_units.mailing\_address2%TYPE,  
 p\_new\_mailing\_zip4 IN c\_family\_economic\_units.mailing\_zip4%TYPE,  
 p\_authorized\_rep1\_last\_name IN  
 c\_family\_economic\_units.authorized\_rep1\_last\_name%TYPE,  
 p\_authorized\_rep1\_first\_name IN  
 c\_family\_economic\_units.authorized\_rep1\_first\_name%TYPE,  
 p\_authorized\_rep1\_mi1 IN c\_family\_economic\_units.authorized\_rep1\_mi1%TYPE,  
 p\_authorized\_rep1\_mi2 IN c\_family\_economic\_units.authorized\_rep1\_mi2%TYPE,

Processing : Inserts a record in c\_transfer\_histories table to keep track of transfers, and updates c\_state\_clients table to reflect the correct agency/clinic for that client. Calls procedure transfer\_la\_pkg.terminate\_client to terminate the client at the from\_agency. Calls transfer\_state\_pkg.delete\_t\_data to delete data from transfer temporary tables.

Output : p\_message - If its null then process is successful, else the message will be the actual error message

#### 5.2.4. PROCEDURE terminate\_client:

Procedure to terminate the client in losing local agency.

Input : p\_transfer\_id

Processing : updates c\_clients table with termination\_date as sysdate and termination\_code as '2' i.e transferred out of agency/clinic. Also terminates the current certification from the from agency and inserts a record in c\_cert\_term\_reasons table to reflect the certification termination. calls procedure transfer\_la\_pkg.delete\_future\_appts to delete future appointments in the from\_agency.

Output : p\_message-If its null then process is successful, else the message will be the actual error message

#### 5.2.5. PROCEDURE delete\_future\_appts:

Procedure to delete future appointments

Input : p\_client\_id

p\_la\_in\_out - Flag to determine if the client is getting transferred out and in.

Processing : Delete following tables for the transferred client/family. Insert into c\_client\_ne\_topics, all the topics associated with that client from the from\_agency to to\_agency. Delete c\_client\_ne\_topics and c\_client\_svc\_materials, aas\_appt\_items, aas\_appt\_clients, c\_client\_services tables for all the future appointments from the from\_agency.

Update aas\_appointments table to cancel the future appointments for that client in the from\_agency. If the family is enrolled in future classes then delete that information from aas\_class\_families table.

Output : p\_message - If its null then process is successful, else the message will be the actual error message

#### 5.2.6. PROCEDURE transfer\_in\_la:

6. PROCEDURE transfer\_in\_la: Procedure to perform transfers within a local agency in the same database.

Input : p\_transfer\_id

Processing : If the transfer type is client transfer,

    Create a new family record in c\_family\_economic\_units table.

    Insert a new record in c\_prev\_families table.

    Update c\_clients table with new family\_id.

    Else if family transfer then

Update c\_family\_economic\_units with the new agency/clinic and address information.

Delete the old phone information from the c\_family\_phones and add new information.

Update c\_feu\_communications table with new family communication record.

Insert a new record in c\_transfer\_histories table recording the transfer information for the clients.

Update c\_state\_clients table with new agency/clinic and address, family etc information.

Update c\_transfers\_info table with the the successful information.

Delete c\_transfer\_comms and c\_transfer\_phones table for the successful transfer.

Output : p\_message - If its null then process is successful, else the message will be the actual error message

7. FUNCTION initiate\_transfer : Function to execute the transfer

Input : p\_processing - (D - Transfer initiated during the day, E - Transfers running at night during End of Day)

p\_transfer\_id

Processing : If the p\_processing flag is D then its a real time transfer, vs E - which indicates pending transfers

to be done during end of day processing.

In any type of transfers it takes all the information from c\_transfers\_info table and check if the client's destination agency/clinic is checked out, if so then give a warning message

saying that 'This participant's record cannot be accessed, the transfer request has been logged,

and the record will be transferred during the next End of Day process.'

Execute the procedure transfer\_state\_pkg.transfer\_between\_las at state. This procedure mainly copies over

data from the transferred\_from agency/clinic to the state in a set of temporary transfer tables.

Execute the procedure transfer\_la\_pkg.insert\_clients at the transferred to agency, to copy over the

client/family data from

Output : p\_message - If its null then process is successful, else the message will be the actual error message