LMIS DEVELOPMENT AND TRAINING

Trip Report

Dhaka, Bangladesh

February 3 - March 3, 1993

Peter Paterson, FPLM

Family Planning
Logistics Management
Project

FPLM

1616 N. Fort Myer Drive
11th Floor
Arlington, Virginia 22209 USA
Tel: (703) 528-7474
Telex: 272896 JSIW UR
Fax: (703) 528-7480

TABLE OF CONTENTS

		Page
I.	EXECUTIVE SUMMARY	1
II.	BACKGROUND	3
III.	ACTIVITIES AND ACCOMPLISHMENTS	4
IV.	FINDINGS AND RECOMMENDATIONS	11

ANNEXES:

1. LMIS Menu Technical Documentation
2. Form 7 Source Code
3. Form 7b Source Code

I. EXECUTIVE SUMMARY

This consultancy focused on four related tasks:

- Technical assistance in the MIS Needs Assessment
- Placing the Oracle/LMIS into production
- Enhancing the Oracle/LMIS
- Continuation of local staff training in Oracle and UNIX

This consultant's accomplishments and activities include the following:

This consultant participated in two MIS Needs Assessment meetings in which members of the Needs Assessment team were briefed about current LMIS operations.

The Oracle/LMIS has been moved into production. All LMIS operations are now being done on the Oracle system. This system has been improved through the implementation of a "menu driven" user interface, revised data entry forms, and a data import facility. The PICK system continues to operate and could be placed back into production if necessary.

Further training was provided to Jim Eberle FPLM, Ashraful Islam FPLM, Milton D'Silva FPLM, Nazmul Huda Khan FPLM, and the rest of the LMIS data entry staff. These trainings were focused on Oracle SQL/Forms, Oracle SQL/Report Writer, Oracle database management, and UNIX systems administration.

In summary, this consultants findings are as follows:

The Oracle/LMIS has evolved to the point where the PICK system is no longer necessary. The Oracle/LMIS has been running smoothly with satisfactory performance since the end of November 1992. Local staff have been trained and are familiar with its operations. The Pick system should be phased out and eventually discontinued.

The skill level of the local staff is at a point where they can comfortably operate and maintain the current Oracle/LMIS. In discussions will Jim Eberle FPLM and Patrick Dougherty FPLM, it appears that additional tasks may be placed on LMIS. If this is the case further staff trainings will be necessary. This training could be provided during another

consultancy or by attending a regional Oracle training seminar.

If the scope of the LMIS is increased faster hardware will become necessary. Currently the performance of the Oracle based LMIS is adequate for processing the F7 and MSR data. If additional applications are added to the current system a faster multi-user hard ware platform will be necessary.

II. BACKGROUND

FPLM's Involvement in Bangladesh began in 1988 with a USAID Mission-supported buy-in to provide training and technical assistance to the MOHFW and various family planning NGO's in the area of logistics management. This buy-in was expanded in 1991 under the FPLM II contract. A major effort of the current Project is assisting the MOHFW to coordinate the data collection and processing efforts of the various MCH and family planning Management Information Systems (MIS's) currently in use, and particularly to help ensure that the Logistics MIS (LMIS) is compatible and consistent with the Ministry's overall MIS development effort.

The LMIS was originally developed to capture data for contraceptive forecasting and logistics management that at the time were not available from the Ministry's basic MIS. The system was implemented in Lotus 1-2-3 on a number of PC's prior to FPLM's involvement; FPLM I maintained the system and partially converted it to more robust software packages. Although the system was able to provide valuable information for logistics management, it was cumbersome to operate and maintain.

With the expanded resources of FPLM II, full-time staff are now available for MIS activities. One of the first has been to convert the LMIS into a more modern, multi-user system capable of producing both monthly and annual reports on stock status and problem points. The current revision was developed in the PICK operating system and database management language. This revision, although an improvement over the previous Lotus 1-2-3 system, is being replaced by a more flexible system using the Oracle RDBMS and UNIX operating system.

The latest Oracle system was developed to provide better reporting on an industry standard RDBMS platform. This platform has been selected for internal use by the MOHFW MIS group. This will make the eventual technology transfer and institutionalization much simpler as the MOHFW will already be familiar with the technology.

III. ACTIVITIES AND ACCOMPLISHMENTS

The following pages detail the activities and accomplishments undertaken during this consultancy.

This consultant participated in two MIS Needs Assessment meetings in which members of the Needs Assessment team were briefed about current LMIS operations. This consultant also researched various hardware and software options for possible use.

Excluding the time spent providing technical expertise in the MIS Needs Assessment, most of this consultancy was split between three tasks: Enhancing the Oracle based LMIS, moving the Oracle system into production, and training local staff.

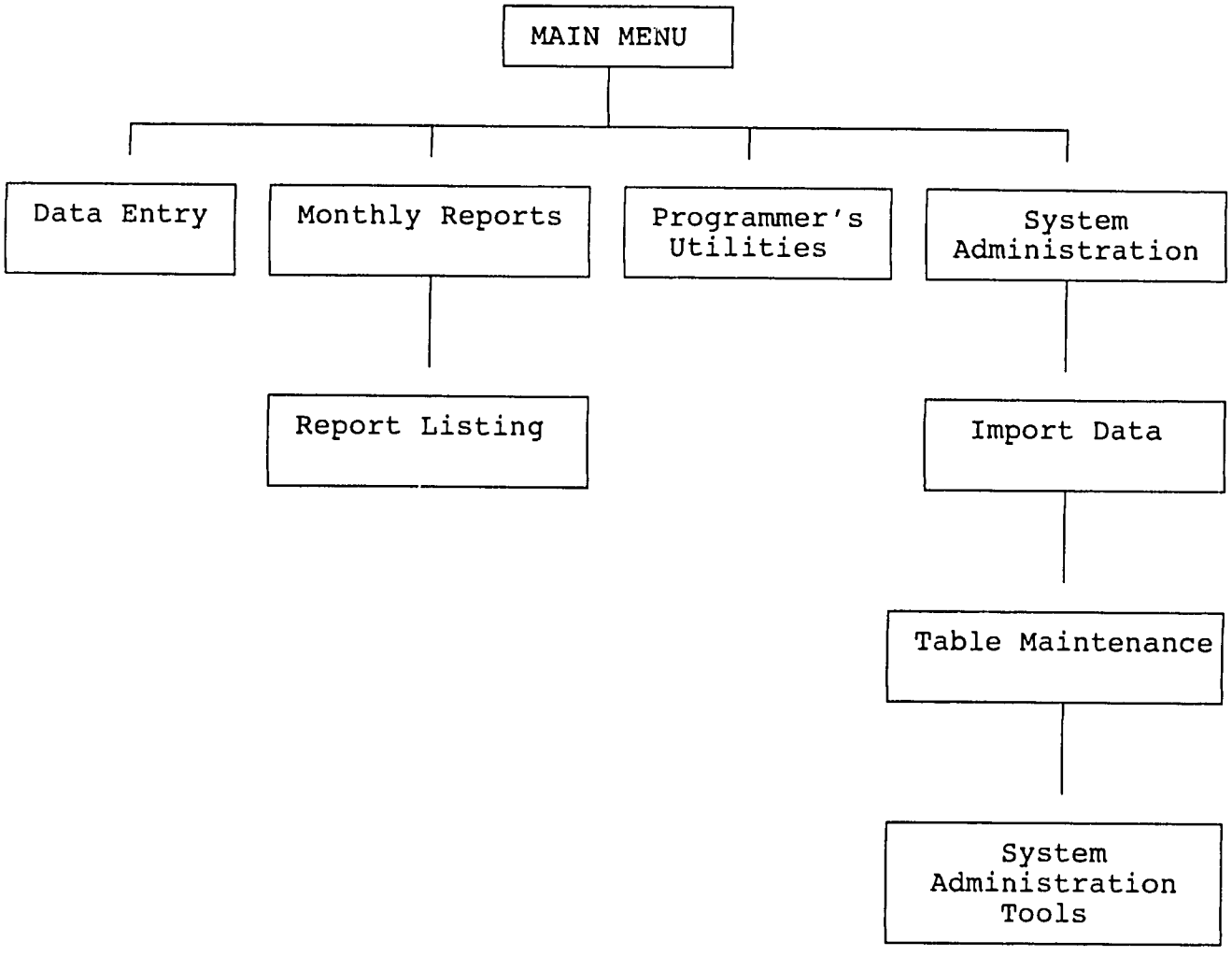
Enhancing the Oracle based LMIS

Enhancements made to the Oracle system include the following:

- o A "menu driven" user interface
- o Enhanced data entry forms
- o Incorporation of the MSR form
- o Addition of an ACSII data import facility

A "menu driven" user interface has been added to the Oracle based LMIS linking the various forms, SQL queries, and reports into one cohesive application. This interface was designed using the Oracle SQL/Menu package and is compatible with all of the other Oracle components. The following table illustrates the LMIS menu structure.

The LMIS Menu System



The use of a menu interface provides better control over an application by restricting access to the system. Users are less likely to get into trouble because they can only select options that are accessible. Menu driven systems are also easier to learn because of the limited number of keystrokes needed to perform a task. In Oracle's SQL/Menu access to menu options is controlled through the use of security groups. The LMIS application has 4 security levels. The security levels are defined as follows:

1. Administrator: Individuals belonging to this group have access to all menu options in the system. Assignment to this group is restricted to personnel who will be responsible for the overall management of the system.

2. Q/A Manager: Individuals belonging to this group have access to table maintenance screens, data entry forms, reports, and some system administration tools. This group's primary function is to provide support to the system manager and to enforce Q/A through the use of system administration tools.

3. Programmer: Individuals belonging to this group have access to the Oracle design tools (SQL/Forms, SQL/Menu, and SQL/Report writer) and the ability to access the database directly through the SQL/Shell.

4. Data Entry Operators: Individuals belonging to this group have access to the data entry forms, E-Mail, and the Lotus 1-2-3 spreadsheet package. The primary function of this group is to enter and update data.

Oracle's SQL/Menu system also has the advantage of being integrated into the Oracle application development environment. This allows reports, forms, and other menus to be added to an application easily. This architecture also allows easy maintenance.

After careful analysis of the PICK system's data entry facility several changes have been made to enhance the data entry screens on the Oracle/LMIS. These changes are highlighted below:

- Individual data entry screens for the F7 and F7b forms
- On screen validation of the Opening and Closing Balance fields
- Collection of Unusables data from Form 7
- Combination of Thana and Field sections of Form 7b

Individual data entry forms have been implemented for the collection of Form 7 and 7b data. This makes data entry easier because the screens now look like the forms. This scheme also allows better verification on data elements on both forms.

The Opening Balance field on the Form 7 and 7b is now verified by looking at the previous months Closing Balance. The Closing Balance field is verified by adding up the Opening Balance, Receipts, Issues, and Returns. By verifying these two key fields data quality of the remaining form is assured.

Four new data elements are now being collected from the Form 7. The Collection of the Unusables Found This Month, Opening Balance of Unusables, Unusables Destroyed This Month, and Closing Balance of Unusables. This additional data will facilitate further studies on the unusable stock situation.

The Thana and Field sections of the Form 7b have been combined on to one data entry screen. This allows cross validation of fields on the Thana and Field sections of the data entry form and speeds the data entry process.

The MSR data form has been incorporated into the Oracle based LMIS prototype. Ashraful Islam, FPLM is currently working on reports for the MSR data. Mr. Islam is also transferring the MSR data from PICK system using the data import facility.

An ASCII data import facility has been added to the Oracle based LMIS prototype. This facility is currently being used to transfer data between PICK and Oracle. Eventually this facility could be used to import data directly from NGOs or other government warehouses.

Moving the Oracle System into Production

After implementing the enhancements discussed above the Oracle based LMIS was moved into production. The PICK system continues to operate as a backup.

Local Staff Training

A large amount of time was spent training local staff during this consultancy. Jim Eberle FPLM, Ashraful Islam FPLM, Milton D'Silva FPLM, and Nazmul Huda Khan FPLM all participated in technical trainings aimed at making the participants familiar with various Oracle design tools. The LMIS Data entry staff have all been trained to use the new data entry forms and have been using them since the middle of the February 1992. Several new reports and table maintenance screens included in the enhanced Oracle/LMIS were implemented by local staff. All of the technical work carried out during this consultancy was done so in conjunction with local technical staff.

IV. FINDINGS AND RECOMMENDATIONS

The following pages detail the recommendations and findings made during this consultancy.

The Oracle/LMIS has evolved to the point where the PICK system is no longer necessary. The Oracle system has been running smoothly with satisfactory performance since the end of November 1992. Local staff have been trained and are familiar with its operations.

Recommendation #1: The Pick system should be phased out and eventually discontinued.

The skill level of the local staff is at a point where they can comfortably operate and maintain the current Oracle/LMIS. However, in discussions will Jim Eberle FPLM and Patrick Dougherty FPLM, it appears that additional tasks may be placed on the LMIS. Some of these proposed tasks, contraceptive demand forecasting, physical inventory, further development of the MSR reports, will require a high level of system resource management. If additional functionality is built into the current Oracle/LMIS it will require advanced systems management and design skills to keep the Oracle/LMIS running smoothly.

Recommendation #2: Ashraful Islam FPLM, Milton D'Silva, and Nazmul Huda Khan FPLM should be given advanced Oracle training. This training could be provided during another consultancy or by attending a regional Oracle training seminar.

The scope of the Oracle/LMIS appears to be increasing. The addition of contraceptive demand forecasting, physical inventory, and further development of the MSR reports, will take the current hardware platform beyond its limits. Currently the performance of the Oracle based LMIS is adequate for processing the F7 and MSR data. If additional functionality is built into the Oracle system a faster multi-user hard ware platform will be necessary.

Recommendation #3: The current Everex 386/33 PC should be upgraded to a more modern 486 PC.

One of the advantages of the current UNIX system is its built in modem support. Milton D'Silva FPLM and Jim Eberle

FPLM spend considerable time traveling between FPLM/Gulsham and the LMIS unit. A high speed modem would allow staff at FPLM/Gulsham to transfer data and process queries without having to visit the LMIS unit.

Recommendation #4: Install high speed modems at FPLM/Gulsham and the LMIS unit.

ANNEX 1

LMIS Menu Technical Documentation

Summary Menu Application Information

A.1 Application Information

Application:	LMIS
Short Name:	LMIS
File Name:	LMIS
Creation Date:	07-FEB-93
Creator:	peter
Last Release Date:	
Menu Directory:	/usr0/peter/menus
Version:	1
Identification:	LMIS Menu System V1.0

A.2 Menu overview

LMIS

- SA_MENU
  - SU_MENU
  - PD_MENU
  - TM_MENU
- PU_MENU
- RR_MENU
  - RL_MENU
- DE_MENU

A.3 DE_MENU

```
+-------------------------------------------------------------------+
|                                                                   |
|            Logistics Management Information System                |
|                                                                   |
|                        Data Entry Screens                         |
|                                                                   |
| 1  Form 7 Data Entry Screen                                       |
| 2  Form 7b Data Entry                                             |
| 3  MSR Form Data Entry Screen                                     |
| 4  Print Form 7 Data                                              |
| 5  Print Form 7b Data                                             |
| 6  Print MSR Data                                                 |
| 7  Return to Previous Menu                                        |
|                                                                   |
| Purpose:                                                          |
|                                                                   |
|                                                                   |
|                                                                   |
+-------------------------------------------------------------------+
```

A.3.1 Item Number: 1

Short Name: Form 7

Command Type: Runform.

Command Line:
==========
runform form7 &UN/&PW

A.3.2 Item Number: 2

Short Name: Form 7b

Command Type: Runform.

Command Line:
==========
runform form7b &UN/&PW

A.3.3 Item Number: 3

Short Name: MSR Form

Command Type: Runform.

Command Line:
==========

runform msr &UN/&PW

A.3.4 Item Number: 4

Short Name: Print Form 7

Command Type: OS command.

Command Line:
==========
runrep report=form7rep userid=&UN/&PW term=&TT

A.3.5 Item Number: 5

Short Name: Print Form 7b

Command Type: OS command.

Command Line:
==========
runrep report=form7brep userid=&UN/&PW term=&TT

A.3.6 Item Number: 6

Short Name: Print MSR

Command Type: OS command.

Command Line:
==========
runrep report=msrrep userid=&UN/&PW term=vt100

A.3.7 Item Number: 7

Short Name: Return

Command Type: Macro command.

Command Line:
==========
PRVMENU;

15

A.4 LMIS

```
+-------------------------------------------------------------------+
|                                                                   |
|              Logistics Managment Information System               |
|                                                                   |
|                              Main Menu                            |
|                                                                   |
| 1  Data Entry (MSR/Form 7-7b)                                     |
| 2  Steven's Advice                                                |
| 3  Run Monthly Reports                                            |
| 4  Programmer's Utilities                                         |
| 5  System Administration                                          |
| 6  Exit LMIS                                                      |
|                                                                   |
| Purpose:                                                          |
|                                                                   |
|                                                                   |
|                                                                   |
+-------------------------------------------------------------------+
```

A.4.1 Item Number: 1

Short Name: Data Entry

Command Type: Menu Selection.

Command Line:
==========
DE_MENU

A.4.2 Item Number: 2

Short Name: Steven

Command Type: OS command + Pause.

Command Line:
==========
/usr/games/steven

A.4.3 Item Number: 3

Short Name: Run Reports

Command Type: Menu Selection.

Command Line:
==========
RR_MENU

A.4.4 Item Number: 4

Short Name: Prog. Utils

Command Type: Menu Selection.

Command Line:
==========
PU_MENU

A.4.5 Item Number: 5

Short Name: Sys. Admin.

Command Type: Menu Selection.

Command Line:
==========
SA_MENU

A.4.6 Item Number: 6

Short Name: Exit

Command Type: No command.

Command Line:
==========
EXIT;

A.5 PD_MENU

```
+-----------------------------------------------------------------+
|                                                                 |
|            Logistics Management Information System              |
|                                                                 |
|                        Process Data Disk                        |
|                                                                 |
| 1  Clear Previous Data Set                                      |
| 2  Load -F7- Data into Oracle from disk                         |
| 3  Display -F7- Load Log                                        |
| 4  Process -F7- Data                                            |
| 5  Load -MSR- Data into Oracle                                  |
| 6  Process -MSR- Data                                           |
| 7  Display -MSR- Load Log                                       |
| 8  Check TRANSLOG Table                                         |
| 9  Return to Previous Menu                                      |
|                                                                 |
| Purpose:                                                        |
|                                                                 |
|                                                                 |
|                                                                 |
+-----------------------------------------------------------------+
```

A.5.1 Item Number: 1

Short Name: Clear Previous

Command Type: SQL*Plus.

Command Line:
==========
sqlplus &UN/&PW @cleardat

A.5.2 Item Number: 2

Short Name: Read F7 Disk

Command Type: OS command.

Command Line:
==========
trans

A.5.3 Item Number: 3

Short Name: Display F7 Log

Command Type: OS command + Pause.

Command Line:
==========
more trans.log

A.5.4 Item Number: 4

Short Name: Process F7 Dat

Command Type: SQL*Plus.

Command Line:
==========
sqlplus &UN/&PW @trans

A.5.5 Item Number: 5

Short Name: Read MSR Disk

Command Type: OS command.

Command Line:
==========
msrtrans

A.5.6 Item Number: 6

Short Name: Process MSR Dat

Command Type: SQL*Plus.

Command Line:
==========
sqlplus &UN/&PW @msrtrans

A.5.7 Item Number: 7

Short Name: Display MSR Log

Command Type: OS command.

Command Line:
==========
more msrtrans.log

A.5.8 Item Number: 8

Short Name: Check TRANSLOG

Command Type: Runform.

Command Line:
==========
runform logtab &UN/&PW

A.5.9 Item Number: 9

Short Name: Return

Command Type: Macro command.

Command Line:
==========
PRVMENU;

A.6 PU_MENU

```
+-------------------------------------------------------------------+
|                                                                   |
|            Logistics Management Information System                |
|                                                                   |
|                      Programmer's Utilities                       |
|                                                                   |
| 1  Enter SQLforms30                                               |
| 2  Enter SQLreport writer                                         |
| 3  Enter SQLmenu50                                                |
| 4  Enter SQLplus Shell                                            |
| 5  Enter C-Shell                                                  |
| 6  Return to Previous Menu                                        |
|                                                                   |
| Purpose:                                                          |
|                                                                   |
|                                                                   |
|                                                                   |
+-------------------------------------------------------------------+
```

A.6.1 Item Number: 1

Short Name: SQLforms30

Command Type: OS command.

Command Line:
==========
sqlforms30 &UN/&PW

A.6.2 Item Number: 2

Short Name: SQLreport

Command Type: OS command.

Command Line:
==========
sqlrep userid=&UN/&PW term=&TT

A.6.3 Item Number: 3

Short Name: SQLmenu50

Command Type: OS command.

Command Line:
==========
sqlmenu50 &UN/&PW

A.6.4 Item Number: 4

Short Name: SQLplus

Command Type: SQL*Plus.

Command Line:
==========
sqlplus &UN/&PW

A.6.5 Item Number: 5

Short Name: C-Shell

Command Type: OS command.

Command Line:
==========
csh

A.6.6 Item Number: 6

Short Name: Return

Command Type: Macro command.

Command Line:
==========
PRVMENU;

A.7 RL_MENU

```
+-----------------------------------------------------------------------+
|                                                                       |
|              Logistics Management Information Systems                 |
|                                                                       |
|                            Report Listing                             |
|                                                                       |
| 1  Run Transit Store Demand Report                                    |
| 2  Run CWH Demand Report                                              |
| 3  Run RWH Demand Report                                              |
| 4  Run DRS Demand Report                                              |
| 5  Run NGO Demand Report                                              |
| 6  Run National Stock Balance Report                                  |
| 7  Run Stock Situation Report                                         |
| 8  Run Stock Situation Report (Summary)                               |
| 9  Run Thana Exception Report                                         |
| 10 Return to Previous Menu                                            |
|                                                                       |
| Purpose:                                                              |
|                                                                       |
|                                                                       |
|                                                                       |
+-----------------------------------------------------------------------+
```

A.7.1 Item Number: 1

Short Name: TS/Demand

Command Type: OS command.

Command Line:
==========
runrep report=allXts USERID=&UN/&PW TERM=&TT

A.7.2 Item Number: 2

Short Name: CWH/Demand

Command Type: OS command.

Command Line:
==========
runrep report=allXcwh userid=&UN/&PW term=&TT

A.7.3 Item Number: 3

Short Name: RWH/Demand

Command Type: OS command.

Command Line:
==========
runrep report=allXrwh userid=&UN/&PW term=&TT

A.7.4 Item Number: 4

Short Name: DRS/Demand

Command Type: OS command.

Command Line:
==========
runrep report=allXdrs userid=&UN/&PW term=&TT

A.7.5 Item Number: 5

Short Name: NGO/Demand

Command Type: OS command.

Command Line:
==========
runrep report=allXngo userid=&UN/&PW term=&TT

A.7.6 Item Number: 6

Short Name: Stock Balance

Command Type: OS command.

Command Line:
==========
runrep report=stkXbal userid=&UN/&PW term=&TT

A.7.7 Item Number: 7

Short Name: Stock Situation

Command Type: OS command.

Command Line:
==========
runrep report=stkXall userid=&UN/&PW term=&TT

A.7.8 Item Number: 8

Short Name: Situation (Sum)

Command Type: OS command.

Command Line:
==========
runrep report=stkXglob userid=&UN/&PW term=&TT

A.7.9 Item Number: 9

Short Name: Exception

Command Type: OS command.

Command Line:
==========
runrep report=naXthn userid=&UN/&PW term=&TT

A.7.10 Item Number: 10

Short Name: Return

Command Type: Macro command.

Command Line:
==========
PRVMENU;

A.8 RR_MENU

```
+-------------------------------------------------------------------+
|                                                                   |
|            Logistics Management Information System                |
|                                                                   |
|                        Run Monthly Reports                        |
|                                                                   |
| 1  Set Monthly Control Parameters                                 |
| 2  Run MKSTAT program                                             |
| 3  Run MKBAL program                                              |
| 4  Report Listing                                                 |
| 5  Copy Report Listings to DOS disk                               |
| 6  Return to Previous Menu                                        |
|                                                                   |
| Purpose:                                                          |
|                                                                   |
|                                                                   |
|                                                                   |
+-------------------------------------------------------------------+
```

A.8.1 Item Number: 1

Short Name: Set Parameters

Command Type: Runform.

Command Line:
==========
runform setparam &UN/&PW

A.8.2 Item Number: 2

Short Name: Run MKSTAT

Command Type: SQL*Plus.

Command Line:
==========
sqlplus &UN/&PW @mkstat

A.8.3 Item Number: 3

Short Name: Run MKBAL

Command Type: SQL*Plus.

Command Line:
==========
sqlplus &UN/&PW @mkbal

A.8.4 Item Number: 4

Short Name: Listing

Command Type: Menu Selection.

Command Line:
==========
RL_MENU

A.8.5 Item Number: 5

Short Name: Copy Reports

Command Type: OS command + Pause.

Command Line:
==========
rpt2dsk

A.8.6 Item Number: 6

Short Name: Return

Command Type: Macro command.

Command Line:
==========
PRVMENU;

A.9 SA_MENU

```
+-------------------------------------------------------------------+
|                                                                   |
|            Logistics Management Information System                |
|                                                                   |
|                 System Administration Utilities                   |
|                                                                   |
| 1  Table Maintenance Screens                                      |
| 2  Process a Transaction Disk                                     |
| 3  Utilities                                                      |
| 4  Enter SQLplus Shell                                            |
| 5  Enter C-Shell                                                  |
| 6  Return to Previous Menu                                        |
|                                                                   |
| Purpose:                                                          |
|                                                                   |
|                                                                   |
|                                                                   |
+-------------------------------------------------------------------+
```

A.9.1 Item Number: 1

Short Name: Table Maint.

Command Type: Menu Selection.

Command Line:
==========
TM_MENU

A.9.2 Item Number: 2

Short Name: Process Disk

Command Type: Menu Selection.

Command Line:
==========
PD_MENU

A.9.3 Item Number: 3

Short Name: Utilities

Command Type: Menu Selection.

Command Line:
==========
SU_MENU

A.9.4 Item Number: 4

Short Name: SQLplus

Command Type: SQL*Plus.

Command Line:
==========
sqlplus &UN/&PW

A.9.5 Item Number: 5

Short Name: C-Shell

Command Type: OS command.

Command Line:
==========
csh

A.9.6 Item Number: 6

Short Name: Return

Command Type: Macro command.

Command Line:
==========
PRVMENU;

A.10 SU_MENU

```
+-------------------------------------------------------------------+
|                                                                   |
|            Logistics Management Information System                |
|                                                                   |
|                              Utilities                            |
|                                                                   |
| 1  Dispense Table Row Count                                       |
| 2  Return to Previous Menu                                        |
|                                                                   |
| Purpose:                                                          |
|                                                                   |
|                                                                   |
|                                                                   |
+-------------------------------------------------------------------+
```

A.10.1 Item Number: 1

Short Name: Row Count

Command Type: OS command.

Command Line:
==========
runrep report=dispXchk userid=&UN/&PW term=&TT

A.10.2 Item Number: 2

Short Name: Return

Command Type: Macro command.

Command Line:
==========
PRVMENU;

A.11 TM_MENU

```
+-------------------------------------------------------------------+
|                                                                   |
|             Logistics Management Information System               |
|                                                                   |
|                    Table Maintenance Screens                      |
|                                                                   |
| 1  Facility Table                                                 |
| 2  Facility Type Table                                            |
| 3  Division Table                                                 |
| 4  Zila Table                                                     |
| 5  Thana Table                                                    |
| 6  Product Table                                                  |
| 7  MSR Product Table                                              |
| 8  Return to Previous Menu                                        |
|                                                                   |
| Purpose:                                                          |
|                                                                   |
|                                                                   |
|                                                                   |
+-------------------------------------------------------------------+
```

A.11.1 Item Number: 1

Short Name: Facility

Command Type: Runform.

Command Line:
==========
runform factab &UN/&PW

A.11.2 Item Number: 2

Short Name: Types

Command Type: Runform.

Command Line:
==========
runform typetab &UN/&PW

A.11.3 Item Number: 3

Short Name: Division

Command Type: Runform.

Command Line:
==========
runform divtab &UN/&PW

A.11.4 Item Number: 4

Short Name: Zila

Command Type: Runform.

Command Line:
==========
runform zilatab &UN/&PW

A.11.5 Item Number: 5

Short Name: Thana

Command Type: Runform.

Command Line:
==========
runform thanatab &UN/&PW

A.11.6 Item Number: 6

Short Name: Product

Command Type: Runform.

Command Line:
==========
runform prodtab &UN/&PW

A.11.7 Item Number: 7

Short Name: MSR Product

Command Type: Runform.

Command Line:
==========
runform msrprodtab &UN/&PW

A.11.8 Item Number: 8

Short Name: Return

Command Type: Macro command.

Command Line:
==========
PRVMENU;

A.12

Role Name	BG Menu	OS Command	Debug
ADMINISTRATOR	Y	Y	Y
Comments: System Administrator Group			
DEVELOPER	Y	Y	Y
Comments: Programmer Group			
DE_STAFF	N	Y	Y
Comments: Data Enter Staff			
LMIS_MGR	Y	Y	Y
Comments:			
QA_MANAGER	N	Y	Y
Comments: QA Manager Group			

ANNEX 2

Form 7 Source Code

```
 Copyright (c) 1988 by the Oracle Corporation */

L*FORMS_VERSION = 03.00.16.08.01
RSE = ON

FINE FORM

 COMMENT = <<<
 This version uses the table F7B_BUFFER to store entered values before being
  written to the DISPENSE table.
 >>>

 NAME = form7
 TITLE = Form 7b Data Entry Screen
 DEFAULT_MENU_APPLICATION = DEFAULT

 DEFINE TRIGGER

    NAME = ON-ERROR
    TRIGGER_TYPE = V3
    SHOW_KEY = OFF
    TEXT = <<<
    if ERROR_CODE = 40401
      then message(:GLOBAL.status || ' - Form 7 Data Saved (V1.3)');
      else message('FRM-' || to_char(ERROR_CODE) || ': ' || ERROR_TEXT);
    end if;
    >>>

ENDDEFINE TRIGGER

DEFINE BLOCK

   NAME = datablk
   DESCRIPTION = datablk
   ROWS_DISPLAYED = 1
   ROWS_BUFFERED = 1
   BASE_LINE = 1
   LINES_PER_ROW = 14
   ARRAY_SIZE = 1

   DEFINE TRIGGER

      NAME = KEY-CLRFRM
      TRIGGER_TYPE = V3
      TEXT = <<<
      CLEAR_FORM(no_validate);
      >>>

   ENDDEFINE TRIGGER

   DEFINE TRIGGER

      NAME = KEY-COMMIT
      TRIGGER_TYPE = V3
```

```
SHOW_KEY = OFF
TEXT = <<<
Declare
  srch_code   char(4);
  srch_yymm   char(4);
Begin
/**------------------------------------------------------------------**
 ** User has committed SAVE THE DATA! Store in dispense file        **
 **------------------------------------------------------------------**
  :GLOBAL.verify := 'OFF';              -- ** Turn off Validation
  :datablk.cd_recpt := NVL(:datablk.cd_recpt,0);
  :datablk.cd_issue := NVL(:datablk.cd_issue,0);
  :datablk.cd_opbal := NVL(:datablk.cd_opbal,0);
  :datablk.cd_clbal := NVL(:datablk.cd_clbal,0);
  :datablk.cd_found := NVL(:datablk.cd_found,0);
  :datablk.cd_unopn := NVL(:datablk.cd_unopn,0);
  :datablk.cd_dstry := NVL(:datablk.cd_dstry,0);
  :datablk.c5_recpt := NVL(:datablk.c5_recpt,0);
  :datablk.c5_issue := NVL(:datablk.c5_issue,0);
  :datablk.c5_opbal := NVL(:datablk.c5_opbal,0);
  :datablk.c5_clbal := NVL(:datablk.c5_clbal,0);
  :datablk.c5_found := NVL(:datablk.c5_found,0);
  :datablk.c5_unopn := NVL(:datablk.c5_unopn,0);
  :datablk.c5_dstry := NVL(:datablk.c5_dstry,0);
  :datablk.ov_recpt := NVL(:datablk.ov_recpt,0);
  :datablk.ov_issue := NVL(:datablk.ov_issue,0);
  :datablk.ov_opbal := NVL(:datablk.ov_opbal,0);
  :datablk.ov_clbal := NVL(:datablk.ov_clbal,0);
  :datablk.ov_found := NVL(:datablk.ov_found,0);
  :datablk.ov_unopn := NVL(:datablk.ov_unopn,0);
  :datablk.ov_dstry := NVL(:datablk.ov_dstry,0);
  :datablk.ct_recpt := NVL(:datablk.ct_recpt,0);
  :datablk.ct_issue := NVL(:datablk.ct_issue,0);
  :datablk.ct_opbal := NVL(:datablk.ct_opbal,0);
  :datablk.ct_clbal := NVL(:datablk.ct_clbal,0);
  :datablk.ct_found := NVL(:datablk.ct_found,0);
  :datablk.ct_unopn := NVL(:datablk.ct_unopn,0);
  :datablk.ct_dstry := NVL(:datablk.ct_dstry,0);
  :datablk.dp_recpt := NVL(:datablk.dp_recpt,0);
  :datablk.dp_issue := NVL(:datablk.dp_issue,0);
  :datablk.dp_opbal := NVL(:datablk.dp_opbal,0);
  :datablk.dp_clbal := NVL(:datablk.dp_clbal,0);
  :datablk.dp_found := NVL(:datablk.dp_found,0);
  :datablk.dp_unopn := NVL(:datablk.dp_unopn,0);
  :datablk.dp_dstry := NVL(:datablk.dp_dstry,0);
  :datablk.ns_recpt := NVL(:datablk.ns_recpt,0);
  :datablk.ns_issue := NVL(:datablk.ns_issue,0);
  :datablk.ns_opbal := NVL(:datablk.ns_opbal,0);
  :datablk.ns_clbal := NVL(:datablk.ns_clbal,0);
  :datablk.ns_found := NVL(:datablk.ns_found,0);
  :datablk.ns_unopn := NVL(:datablk.ns_unopn,0);
  :datablk.ns_dstry := NVL(:datablk.ns_dstry,0);
  srch_code   := :datablk.fac_code;
  srch_yymm   := :datablk.yymm;
  if :GLOBAL.status = 'NEW'
    then
```

```
        insert into lmis.dispense
        values (srch_code,srch_yymm,1,
                :datablk.cd_recpt,:datablk.cd_issue,0,
                :datablk.cd_opbal,:datablk.cd_clbal,0,
                :datablk.cd_found,:datablk.cd_unopn,:datablk.cd_dstry);
        insert into lmis.dispense
        values (srch_code,srch_yymm,2,
                :datablk.c5_recpt,:datablk.c5_issue,0,
                :datablk.c5_opbal,:datablk.c5_clbal,0,
                :datablk.c5_found,:datablk.c5_unopn,:datablk.c5_dstry);
        insert into lmis.dispense
        values (srch_code,srch_yymm,3,
                :datablk.ov_recpt,:datablk.ov_issue,0,
                :datablk.ov_opbal,:datablk.ov_clbal,0,
                :datablk.ov_found,:datablk.ov_unopn,:datablk.ov_dstry);
        insert into lmis.dispense
        values (srch_code,srch_yymm,4,
                :datablk.ct_recpt,:datablk.ct_issue,0,
                :datablk.ct_opbal,:datablk.ct_clbal,0,
                :datablk.ct_found,:datablk.ct_unopn,:datablk.ct_dstry);
        insert into lmis.dispense
        values (srch_code,srch_yymm,5,
                :datablk.dp_recpt,:datablk.dp_issue,0,
                :datablk.dp_opbal,:datablk.dp_clbal,0,
                :datablk.dp_found,:datablk.dp_unopn,:datablk.dp_dstry);
        insert into lmis.dispense
        values (srch_code,srch_yymm,6,
                :datablk.ns_recpt,:datablk.ns_issue,0,
                :datablk.ns_opbal,:datablk.ns_clbal,0,
                :datablk.ns_found,:datablk.ns_unopn,:datablk.ns_dstry);
      else
        update lmis.dispense
          set open_bal  = :datablk.cd_opbal,
              receipts  = :datablk.cd_recpt,
              issues    = :datablk.cd_issue,
              found     = :datablk.cd_found,
              un_opnbal = :datablk.cd_unopn,
              destroy   = :datablk.cd_dstry,
              clos_bal = :datablk.cd_clbal
          where dispense.fac_code  = srch_code and
                dispense.yymm      = srch_yymm and
                dispense.prod_code = 1;                -- ** Condoms
        update lmis.dispense
          set open_bal  = :datablk.c5_opbal,
              receipts  = :datablk.c5_recpt,
              issues    = :datablk.c5_issue,
              found     = :datablk.c5_found,
              un_opnbal = :datablk.c5_unopn,
              destroy   = :datablk.c5_dstry,
              clos_bal = :datablk.c5_clbal
          where dispense.fac_code  = srch_code and
                dispense.yymm      = srch_yymm and
                dispense.prod_code = 2;                -- ** OPC-5
        update lmis.dispense
          set open_bal  = :datablk.ov_opbal,
              receipts  = :datablk.ov_recpt,
```

```
                issues    = :datablk.ov_issue,
                found     = :datablk.ov_found,
                un_opnbal = :datablk.ov_unopn,
                destroy   = :datablk.ov_dstry,
                clos_bal = :datablk.ov_clbal
            where dispense.fac_code  = srch_code and
                  dispense.yymm      = srch_yymm and
                  dispense.prod_code = 3;                -- ** Ovral
        update lmis.dispense
            set open_bal  = :datablk.ct_opbal,
                receipts  = :datablk.ct_recpt,
                issues    = :datablk.ct_issue,
                found     = :datablk.ct_found,
                un_opnbal = :datablk.ct_unopn,
                destroy   = :datablk.ct_dstry,
                clos_bal = :datablk.ct_clbal
            where dispense.fac_code  = srch_code and
                  dispense.yymm      = srch_yymm and
                  dispense.prod_code = 4;                -- ** Copper-T
        update lmis.dispense
            set open_bal  = :datablk.dp_opbal,
                receipts  = :datablk.dp_recpt,
                issues    = :datablk.dp_issue,
                found     = :datablk.dp_found,
                un_opnbal = :datablk.dp_unopn,
                destroy   = :datablk.dp_dstry,
                clos_bal = :datablk.dp_clbal
            where dispense.fac_code  = srch_code and
                  dispense.yymm      = srch_yymm and
                  dispense.prod_code = 5;                -- ** Deprovera
        update lmis.dispense
            set open_bal  = :datablk.ns_opbal,
                receipts  = :datablk.ns_recpt,
                issues    = :datablk.ns_issue,
                found     = :datablk.ns_found,
                un_opnbal = :datablk.ns_unopn,
                destroy   = :datablk.ns_dstry,
                clos_bal = :datablk.ns_clbal
            where dispense.fac_code  = srch_code and
                  dispense.yymm      = srch_yymm and
                  dispense.prod_code = 6;                -- ** Noristerat
    end if;
    if :GLOBAL.status = 'NEW'                        -- ** Add Audit Inf.
      then
        insert into lmis.f7_audit
        values (:datablk.fac_code, :datablk.yymm,
                USER, SYSDATE, USER, SYSDATE);
      else
        update lmis.f7_audit
           set mdop_code = USER, md_date = SYSDATE
         where f7_audit.fac_code = :datablk.fac_code and
               f7_audit.yymm = :datablk.yymm;
    end if;
    COMMIT;
    CLEAR_FORM(NO_VALIDATE);
  END;
```

```
      >>>

    ENDDEFINE TRIGGER

    DEFINE TRIGGER

       NAME = KEY-DOWN
       TRIGGER_TYPE = V3
       SHOW_KEY = OFF
       TEXT = <<<
       Declare
         src_fld char(3);
         nxt_fld char(4);
         tar_fld char(10);
         bas_fld char(5);
       Begin
         nxt_fld := 'NULL';
         src_fld := substr(:SYSTEM.cursor_field,9,3);
         bas_fld := substr(:SYSTEM.cursor_field,12,5);
         if    src_fld = 'CD_' then nxt_fld := 'C5_';
         elsif src_fld = 'C5_' then nxt_fld := 'OV_';
         elsif src_fld = 'OV_' then nxt_fld := 'CT_';
         elsif src_fld = 'CT_' then nxt_fld := 'DP_';
         elsif src_fld = 'DP_' then nxt_fld := 'NS_';
         else nxt_fld := 'NULL';
         end if;
         if    (src_fld||bas_fld) = 'NS_OPBAL'
           then
             nxt_fld := 'CD_';
             bas_fld := 'UNOPN';
         elsif (src_fld||bas_fld) = 'NS_RECPT'
           then
             nxt_fld := 'CD_';
             bas_fld := 'DSTRY';
         end if;
         if nxt_fld <> 'NULL'
           then
             tar_fld := nxt_fld || bas_fld;
             GO_FIELD(tar_fld);
           else
             next_field;
         end if;
       END;
       >>>

    ENDDEFINE TRIGGER

    DEFINE TRIGGER

       NAME = KEY-NXTREC
       TRIGGER_TYPE = V3
       TEXT = <<<
       EXECUTE_TRIGGER('KEY-DOWN');
       >>>

    ENDDEFINE TRIGGER
```

```
DEFINE TRIGGER

   NAME = KEY-PRVREC
   TRIGGER_TYPE = V3
   SHOW_KEY = OFF
   TEXT = <<<
   EXECUTE_TRIGGER('KEY-UP');
   >>>

ENDDEFINE TRIGGER

DEFINE TRIGGER

   NAME = KEY-UP
   TRIGGER_TYPE = V3
   SHOW_KEY = OFF
   TEXT = <<<
   Declare
     src_fld char(3);
     nxt_fld char(4);
     tar_fld char(10);
     bas_fld char(5);
   Begin
     nxt_fld := 'NULL';
     src_fld := substr(:SYSTEM.cursor_field,9,3);
     bas_fld := substr(:SYSTEM.cursor_field,12,5);
     if    src_fld = 'C5_' then nxt_fld := 'CD_';
     elsif src_fld = 'OV_' then nxt_fld := 'C5_';
     elsif src_fld = 'CT_' then nxt_fld := 'OV_';
     elsif src_fld = 'DP_' then nxt_fld := 'CT_';
     elsif src_fld = 'NS_' then nxt_fld := 'DP_';
     else nxt_fld := 'NULL';
     end if;
     if    (src_fld||bas_fld) = 'CD_UNOPN'
       then
         nxt_fld := 'NS_';
         bas_fld := 'OPBAL';
     elsif (src_fld||bas_fld) = 'CD_DSTRY'
       then
         nxt_fld := 'NS_';
         bas_fld := 'RECPT';
     end if;
     if nxt_fld <> 'NULL'
       then
         tar_fld := nxt_fld || bas_fld;
         GO_FIELD(tar_fld);
       else
         next_field;
     end if;
   END;
   >>>

ENDDEFINE TRIGGER

DEFINE FIELD
```

```
NAME = FAC_CODE
DATATYPE = CHAR
LENGTH = 4
DISPLAY_LENGTH = 4
QUERY_LENGTH = 4
BASE_TABLE = OFF
MANDATORY = ON
PAGE = 1
LINE = 4
COLUMN = 26
HELP = Enter the Facility Code
LOV_TEXT = <<<
select fac_name, fac_code, type_name
  into :datablk.fac_name, :datablk.fac_code, :datablk.type_name
  from lmis.facility,lmis.fac_types
 where substr(facility.fac_code,1,1) = fac_types.type_code
       and
       substr(facility.fac_code,1,1) not in ('T','F')
order by fac_name
>>>
LOV_TITLE = Facilities Listing
LOV_X = 15
LOV_Y = 6
AUTOSKIP = ON
AUTOHELP = ON

DEFINE TRIGGER

   NAME = ON-VALIDATE-FIELD
   TRIGGER_TYPE = V3
   SHOW_KEY = OFF
   TEXT = <<<
   Begin
     :GLOBAL.verify   := 'NO';
     :GLOBAL.status   := 'NEW';
     :GLOBAL.no_bal   := '0';
     :datablk.fac_code := upper(:datablk.fac_code);
     if substr(:datablk.fac_code,1,1) in ('T','F')
       then
         message('Only CWHs,RWHs,DRSs and NGOs may submit FORM 7s');
         BELL;
         raise FORM_TRIGGER_FAILURE;
     end if;
     select facility.fac_name,
            fac_types.type_name
       into :datablk.fac_name,:datablk.type_name
       from lmis.facility,lmis.fac_types
      where facility.fac_code = :datablk.fac_code
            and
            fac_types.type_code = substr(facility.fac_code,1,1);
  Exception
    when NO_DATA_FOUND then
      BELL;
      message('Facility Code Not Found In Facility File - Press [LIST
      raise FORM_TRIGGER_FAILURE;
```

```
            End;
            >>>

         ENDDEFINE TRIGGER

      ENDDEFINE FIELD

      DEFINE FIELD

         NAME = YYMM
         DATATYPE = CHAR
         LENGTH = 4
         DISPLAY_LENGTH = 4
         QUERY_LENGTH = 4
         BASE_TABLE = OFF
         MANDATORY = ON
         PAGE = 1
         LINE = 5
         COLUMN = 26
         HELP = Enter the Processing Year and Month. i.e. January 1993 = 9301
         AUTOHELP = ON

         DEFINE TRIGGER

            NAME = ON-VALIDATE-FIELD
            TRIGGER_TYPE = V3
            SHOW_KEY = OFF
            TEXT = <<<
            Begin
              if substr(:datablk.yymm,3,2) > '12' or
                 substr(:datablk.yymm,3,2) < '01' or
                 length(:datablk.yymm) < 4
               then
                 message('Invalid Month Code - Try Again!');
                 raise FORM_TRIGGER_FAILURE;
              end if;
            End;
            >>>

         ENDDEFINE TRIGGER

         DEFINE TRIGGER

            NAME = KEY-NXTFLD
            TRIGGER_TYPE = V3
            SHOW_KEY = OFF
            TEXT = <<<
            Declare
            /* */
              prevyy      number(2);
              prevmm      number(2);
              prevnum     number(4);
              srch_code    char(4);
              srch_yymm    char(4);
              update_flag number(1);
            /* */
```

```
    CURSOR disp is select dispense.prod_code,
                          dispense.open_bal,
                          dispense.receipts,
                          dispense.found,
                          dispense.un_opnbal,
                          dispense.destroy,
                          dispense.issues,
                          dispense.clos_bal
                     from lmis.dispense
                    where dispense.fac_code  = srch_code and
                          dispense.yymm      = srch_yymm;
/* */
Begin
  :GLOBAL.verify   := 'NO';
  :GLOBAL.status   := 'NEW';
  prevyy := to_number(substr(:datablk.yymm,1,2));
  prevmm := to_number(substr(:datablk.yymm,3,2));
  if prevmm = 1
    then
      prevmm := 12;
      prevyy := prevyy - 1;
    else
      prevmm := prevmm - 1;
  end if;
  prevnum  := (prevyy*100)+prevmm;
  :GLOBAL.prevdate := substr(to_char(prevnum,'9999'),2,4);
/**-------------------------------------------------------------------
 ** Search for Previous F7  Data if none found then we have a new F
 **-------------------------------------------------------------------
  srch_code    := :datablk.fac_code;
  srch_yymm    := :datablk.yymm;
  update_flag := 0;
  For disprec in disp LOOP
    update_flag := 1;              -- ** Got a previously stored form
    if disprec.prod_code = 1       -- ** Condoms
      then
        :datablk.cd_opbal := disprec.open_bal;
        :datablk.cd_recpt := disprec.receipts;
        :datablk.cd_found := NVL(disprec.found,0);
        :datablk.cd_unopn := NVL(disprec.un_opnbal,0);
        :datablk.cd_dstry := NVL(disprec.destroy,0);
        :datablk.cd_issue := disprec.issues;
        :datablk.cd_clbal := disprec.clos_bal;
        :datablk.cd_uncls := :datablk.cd_unopn - :datablk.cd_dstry;
    elsif disprec.prod_code = 2  -- ** OPC-5
      then
        :datablk.c5_opbal := disprec.open_bal;
        :datablk.c5_recpt := disprec.receipts;
        :datablk.c5_found := NVL(disprec.found,0);
        :datablk.c5_unopn := NVL(disprec.un_opnbal,0);
        :datablk.c5_dstry := NVL(disprec.destroy,0);
        :datablk.c5_issue := disprec.issues;
        :datablk.c5_clbal := disprec.clos_bal;
        :datablk.c5_uncls := :datablk.c5_unopn - :datablk.c5_dstry;
    elsif disprec.prod_code = 3  -- ** OVRAL
      then
```

```
                :datablk.ov_opbal := disprec.open_bal;
                :datablk.ov_recpt := disprec.receipts;
                :datablk.ov_found := NVL(disprec.found,0);
                :datablk.ov_unopn := NVL(disprec.un_opnbal,0);
                :datablk.ov_dstry := NVL(disprec.destroy,0);
                :datablk.ov_issue := disprec.issues;
                :datablk.ov_clbal := disprec.clos_bal;
                :datablk.ov_uncls := :datablk.ov_unopn - :datablk.ov_dstry
            elsif disprec.prod_code = 4  -- ** COPPER-T
              then
                :datablk.ct_opbal := disprec.open_bal;
                :datablk.ct_recpt := disprec.receipts;
                :datablk.ct_found := NVL(disprec.found,0);
                :datablk.ct_unopn := NVL(disprec.un_opnbal,0);
                :datablk.ct_dstry := NVL(disprec.destroy,0);
                :datablk.ct_issue := disprec.issues;
                :datablk.ct_clbal := disprec.clos_bal;
                :datablk.ct_uncls := :datablk.ct_unopn - :datablk.ct_dstry
            elsif disprec.prod_code = 5  -- ** DEPO-PROVERA
              then
                :datablk.dp_opbal := disprec.open_bal;
                :datablk.dp_recpt := disprec.receipts;
                :datablk.dp_found := NVL(disprec.found,0);
                :datablk.dp_unopn := NVL(disprec.un_opnbal,0);
                :datablk.dp_dstry := NVL(disprec.destroy,0);
                :datablk.dp_issue := disprec.issues;
                :datablk.dp_clbal := disprec.clos_bal;
                :datablk.dp_uncls := :datablk.dp_unopn - :datablk.dp_dstry;
            elsif disprec.prod_code = 6  -- ** NORISTERAT
              then
                :datablk.ns_opbal := disprec.open_bal;
                :datablk.ns_recpt := disprec.receipts;
                :datablk.ns_found := NVL(disprec.found,0);
                :datablk.ns_unopn := NVL(disprec.un_opnbal,0);
                :datablk.ns_dstry := NVL(disprec.destroy,0);
                :datablk.ns_issue := disprec.issues;
                :datablk.ns_clbal := disprec.clos_bal;
                :datablk.ns_uncls := :datablk.ns_unopn - :datablk.ns_dstry;
            END IF;
          END LOOP;
          if update_flag = 1
            then :GLOBAL.status := 'OLD';
          end if;
          GO_FIELD('CD_OPBAL');
        END;
        >>>

      ENDDEFINE TRIGGER

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = CD_OPBAL
      DATATYPE = RINT
      LENGTH = 12
```

```
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 9
      COLUMN = 15
      HELP = Enter value for : CD_OPBAL
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = ON-VALIDATE-FIELD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         Declare
           prevbal    number(10);
         Begin
           select dispense.clos_bal
             into prevbal
             from lmis.dispense
            where dispense.fac_code = :datablk.fac_code and
                  dispense.yymm = :GLOBAL.prevdate and
                  dispense.prod_code = 1;
           if prevbal <> :datablk.cd_opbal
             then
               if :GLOBAL.verify = 'NO'
                 then
                   :GLOBAL.verify := 'YES';
                   message
         ('Opening Balance not equal to previous Closing Balance -- CHECK VA.
                   bell;
                   raise FORM_TRIGGER_FAILURE;
               end if;
           end if;
           if :GLOBAL.verify <> 'OFF'
             then :GLOBAL.verify := 'NO';
           end if;
         Exception
           when NO_DATA_FOUND then
              :GLOBAL.no_bal := '1';
         End;
         >>>

      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = CD_RECPT
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
```

```
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 9
        COLUMN = 28
        HELP = Enter value for : CD_RECPT
        INPUT_MASK = 999,999,999
        OUTPUT_MASK = 999,999,999
        DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

        NAME = CD_ISSUE
        DATATYPE = RINT
        LENGTH = 12
        DISPLAY_LENGTH = 12
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        MANDATORY = ON
        PAGE = 1
        LINE = 9
        COLUMN = 41
        INPUT_MASK = 999,999,999
        OUTPUT_MASK = 999,999,999
        DEFAULT = 0
        AUTOHELP = ON

    ENDDEFINE FIELD

    DEFINE FIELD

        NAME = CD_FOUND
        DATATYPE = RINT
        LENGTH = 12
        DISPLAY_LENGTH = 12
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 9
        COLUMN = 54
        HELP = Enter value for : CD_FOUND
        INPUT_MASK = 999,999,999
        OUTPUT_MASK = 999,999,999
        DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

        NAME = CD_CLBAL
        DATATYPE = RINT
        LENGTH = 12
        DISPLAY_LENGTH = 12
        QUERY_LENGTH = 12
```

```
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 9
      COLUMN = 67
      HELP = Enter value for : CD_CLBAL
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = ON-VALIDATE-FIELD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         Begin
           if ((:datablk.cd_opbal + :datablk.cd_recpt) -
               (:datablk.cd_issue - :datablk.cd_found))
             <> :datablk.cd_clbal
             then
               if :GLOBAL.verify = 'NO'
                 then
                   :GLOBAL.verify := 'YES';
                   message('Closing Balance does not add up -- CHECK VALUE!')
                   bell;
                   raise FORM_TRIGGER_FAILURE;
               end if;
           end if;
           if :GLOBAL.verify <> 'OFF'
             then :GLOBAL.verify := 'NO';
           end if;
         End;
         >>>

      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = CD_UNOPN
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 16
      COLUMN = 15
      HELP = Enter value for : CD_UNOPN
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

   ENDDEFINE FIELD
```

```
DEFINE FIELD

   NAME = CD_DSTRY
   DATATYPE = RINT
   LENGTH = 12
   DISPLAY_LENGTH = 12
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 16
   COLUMN = 28
   HELP = Enter value for : CD_DSTRY
   INPUT_MASK = 999,999,999
   OUTPUT_MASK = 999,999,999
   DEFAULT = 0

   DEFINE TRIGGER

      NAME = KEY-NXTFLD
      TRIGGER_TYPE = V3
      SHOW_KEY = OFF
      TEXT = <<<
      :datablk.cd_uncls := :datablk.cd_unopn - :datablk.cd_dstry;
      NEXT_FIELD;
      >>>

   ENDDEFINE TRIGGER

ENDDEFINE FIELD

DEFINE FIELD

   NAME = C5_OPBAL
   DATATYPE = RINT
   LENGTH = 12
   DISPLAY_LENGTH = 12
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 10
   COLUMN = 15
   HELP = Enter value for : C5_OPBAL
   INPUT_MASK = 999,999,999
   OUTPUT_MASK = 999,999,999
   DEFAULT = 0

   DEFINE TRIGGER

      NAME = ON-VALIDATE-FIELD
      TRIGGER_TYPE = V3
      SHOW_KEY = OFF
      TEXT = <<<
      Declare
        prevbal    number(10);
      Begin
        select dispense.clos_bal
```

```
            into prevbal
            from lmis.dispense
           where dispense.fac_code = :datablk.fac_code and
                 dispense.yymm = :GLOBAL.prevdate and
                 dispense.prod_code = 2;
         if prevbal <> :datablk.c5_opbal
           then
             if :GLOBAL.verify = 'NO'
               then
                 :GLOBAL.verify := 'YES';
                 message
       ('Opening Balance not equal to previous Closing Balance -- CHECK VA
                 bell;
                 raise FORM_TRIGGER_FAILURE;
             end if;
         end if;
         if :GLOBAL.verify <> 'OFF'
           then :GLOBAL.verify := 'NO';
         end if;
       Exception
         when NO_DATA_FOUND then
           :GLOBAL.no_bal := '1';
       End;
       >>>

    ENDDEFINE TRIGGER

  ENDDEFINE FIELD

  DEFINE FIELD

     NAME = C5_RECPT
     DATATYPE = RINT
     LENGTH = 12
     DISPLAY_LENGTH = 12
     QUERY_LENGTH = 12
     BASE_TABLE = OFF
     PAGE = 1
     LINE = 10
     COLUMN = 28
     HELP = Enter value for : C5_RECPT
     INPUT_MASK = 999,999,999
     OUTPUT_MASK = 999,999,999
     DEFAULT = 0

  ENDDEFINE FIELD

  DEFINE FIELD

     NAME = C5_ISSUE
     DATATYPE = RINT
     LENGTH = 12
     DISPLAY_LENGTH = 12
     QUERY_LENGTH = 12
     BASE_TABLE = OFF
     MANDATORY = ON
```

```
      PAGE = 1
      LINE = 10
      COLUMN = 41
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0
      AUTOHELP = ON

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = C5_FOUND
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 10
      COLUMN = 54
      HELP = Enter value for : C5_FOUND
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = C5_CLBAL
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 10
      COLUMN = 67
      HELP = Enter value for : C5_CLBAL
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = ON-VALIDATE-FIELD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         Begin
           if ((:datablk.c5_opbal + :datablk.c5_recpt) -
               (:datablk.c5_issue - :datablk.c5_found))
             <> :datablk.c5_clbal
             then
               if :GLOBAL.verify = 'NO'
```

```
                then
                  :GLOBAL.verify := 'YES';
                  message('Closing Balance does not add up -- CHECK VALUE!')
                  bell;
                  raise FORM_TRIGGER_FAILURE;
              end if;
          end if;
          if :GLOBAL.verify <> 'OFF'
            then :GLOBAL.verify := 'NO';
          end if;
        End;
        >>>

    ENDDEFINE TRIGGER

  ENDDEFINE FIELD

  DEFINE FIELD

     NAME = C5_UNOPN
     DATATYPE = RINT
     LENGTH = 12
     DISPLAY_LENGTH = 12
     QUERY_LENGTH = 12
     BASE_TABLE = OFF
     PAGE = 1
     LINE = 17
     COLUMN = 15
     HELP = Enter value for : C5_UNOPN
     INPUT_MASK = 999,999,999
     OUTPUT_MASK = 999,999,999
     DEFAULT = 0

  ENDDEFINE FIELD

  DEFINE FIELD

     NAME = C5_DSTRY
     DATATYPE = RINT
     LENGTH = 12
     DISPLAY_LENGTH = 12
     QUERY_LENGTH = 12
     BASE_TABLE = OFF
     PAGE = 1
     LINE = 17
     COLUMN = 28
     HELP = Enter value for : C5_DSTRY
     INPUT_MASK = 999,999,999
     OUTPUT_MASK = 999,999,999
     DEFAULT = 0

     DEFINE TRIGGER

        NAME = key-nxtfld
        TRIGGER_TYPE = V3
        SHOW_KEY = OFF
```

```
        TEXT = <<<
        :datablk.c5_uncls := :datablk.c5_unopn - :datablk.c5_dstry;
        NEXT_FIELD;
        >>>

      ENDDEFINE TRIGGER

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = OV_OPBAL
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 11
      COLUMN = 15
      HELP = Enter value for : OV_OPBAL
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

      DEFINE TRIGGER

        NAME = ON-VALIDATE-FIELD
        TRIGGER_TYPE = V3
        SHOW_KEY = OFF
        TEXT = <<<
        Declare
          prevbal    number(10);
        Begin
          select dispense.clos_bal
            into prevbal
            from lmis.dispense
           where dispense.fac_code = :datablk.fac_code and
                 dispense.yymm = :GLOBAL.prevdate and
                 dispense.prod_code = 3;
          if prevbal <> :datablk.ov_opbal
            then
              if :GLOBAL.verify = 'NO'
                then
                  :GLOBAL.verify := 'YES';
                  message
        ('Opening Balance not equal to previous Closing Balance -- CHECK VA
                  bell;
                  raise FORM_TRIGGER_FAILURE;
              end if;
          end if;
          if :GLOBAL.verify <> 'OFF'
            then :GLOBAL.verify := 'NO';
          end if;
        Exception
          when NO_DATA_FOUND then
```

```
            :GLOBAL.no_bal := '1';
        End;
        >>>

      ENDDEFINE TRIGGER

    ENDDEFINE FIELD

    DEFINE FIELD

       NAME = OV_RECPT
       DATATYPE = RINT
       LENGTH = 12
       DISPLAY_LENGTH = 12
       QUERY_LENGTH = 12
       BASE_TABLE = OFF
       PAGE = 1
       LINE = 11
       COLUMN = 28
       HELP = Enter value for : OV_RECPT
       INPUT_MASK = 999,999,999
       OUTPUT_MASK = 999,999,999
       DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

       NAME = OV_ISSUE
       DATATYPE = RINT
       LENGTH = 12
       DISPLAY_LENGTH = 12
       QUERY_LENGTH = 12
       BASE_TABLE = OFF
       MANDATORY = ON
       PAGE = 1
       LINE = 11
       COLUMN = 41
       INPUT_MASK = 999,999,999
       OUTPUT_MASK = 999,999,999
       DEFAULT = 0
       AUTOHELP = ON

    ENDDEFINE FIELD

    DEFINE FIELD

       NAME = OV_FOUND
       DATATYPE = RINT
       LENGTH = 12
       DISPLAY_LENGTH = 12
       QUERY_LENGTH = 12
       BASE_TABLE = OFF
       PAGE = 1
       LINE = 11
       COLUMN = 54
```

```
      HELP = Enter value for : OV_FOUND
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = OV_CLBAL
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 11
      COLUMN = 67
      HELP = Enter value for : OV_CLBAL
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = ON-VALIDATE-FIELD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         Begin
           if ((:datablk.ov_opbal + :datablk.ov_recpt) -
               (:datablk.ov_issue - :datablk.ov_found))
             <> :datablk.ov_clbal
             then
               if :GLOBAL.verify = 'NO'
                 then
                   :GLOBAL.verify := 'YES';
                   message('Closing Balance does not add up -- CHECK VALUE!'
                   bell;
                   raise FORM_TRIGGER_FAILURE;
               end if;
           end if;
           if :GLOBAL.verify <> 'OFF'
             then :GLOBAL.verify := 'NO';
           end if;
         End;
         >>>

      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = OV_UNOPN
      DATATYPE = RINT
```

```
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 18
      COLUMN = 15
      HELP = Enter value for : OV_UNOPN
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

  ENDDEFINE FIELD

  DEFINE FIELD

     NAME = OV_DSTRY
     DATATYPE = RINT
     LENGTH = 12
     DISPLAY_LENGTH = 12
     QUERY_LENGTH = 12
     BASE_TABLE = OFF
     PAGE = 1
     LINE = 18
     COLUMN = 28
     HELP = Enter value for : OV_DSTRY
     INPUT_MASK = 999,999,999
     OUTPUT_MASK = 999,999,999
     DEFAULT = 0

     DEFINE TRIGGER

        NAME = KEY-NXTFLD
        TRIGGER_TYPE = V3
        SHOW_KEY = OFF
        TEXT = <<<
        :datablk.ov_uncls := :datablk.ov_unopn - :datablk.ov_dstry;
        NEXT_FIELD;
        >>>

     ENDDEFINE TRIGGER

  ENDDEFINE FIELD

  DEFINE FIELD

     NAME = CT_OPBAL
     DATATYPE = RINT
     LENGTH = 12
     DISPLAY_LENGTH = 12
     QUERY_LENGTH = 12
     BASE_TABLE = OFF
     PAGE = 1
     LINE = 12
     COLUMN = 15
     HELP = Enter value for : CT_OPBAL
```

```
        INPUT_MASK = 999,999,999
        OUTPUT_MASK = 999,999,999
        DEFAULT = 0

        DEFINE TRIGGER

           NAME = ON-VALIDATE-FIELD
           TRIGGER_TYPE = V3
           SHOW_KEY = OFF
           TEXT = <<<
           Declare
             prevbal    number(10);
           Begin
             select dispense.clos_bal
               into prevbal
               from lmis.dispense
              where dispense.fac_code = :datablk.fac_code and
                    dispense.yymm = :GLOBAL.prevdate and
                    dispense.prod_code = 4;
             if prevbal <> :datablk.ct_opbal
               then
                 if :GLOBAL.verify = 'NO'
                   then
                     :GLOBAL.verify := 'YES';
                     message
           ('Opening Balance not equal to previous Closing Balance -- CHECK
                     bell;
                     raise FORM_TRIGGER_FAILURE;
                 end if;
             end if;
             if :GLOBAL.verify <> 'OFF'
               then :GLOBAL.verify := 'NO';
             end if;
           Exception
             when NO_DATA_FOUND then
               :GLOBAL.no_bal := '1';
           End;
           >>>

        ENDDEFINE TRIGGER

     ENDDEFINE FIELD

     DEFINE FIELD

        NAME = CT_RECPT
        DATATYPE = RINT
        LENGTH = 12
        DISPLAY_LENGTH = 12
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 12
        COLUMN = 28
        HELP = Enter value for : CT_RECPT
        INPUT_MASK = 999,999,999
```

```
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = CT_ISSUE
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      MANDATORY = ON
      PAGE = 1
      LINE = 12
      COLUMN = 41
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0
      AUTOHELP = ON

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = CT_FOUND
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      MANDATORY = ON
      PAGE = 1
      LINE = 12
      COLUMN = 54
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0
      QUERY = OFF
      AUTOHELP = ON

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = CT_CLBAL
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 12
      COLUMN = 67
      HELP = Enter value for : CT_CLBAL
```

```
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = ON-VALIDATE-FIELD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         Begin
           if ((:datablk.ct_opbal + :datablk.ct_recpt) -
               (:datablk.ct_issue - :datablk.ct_found))
             <> :datablk.ct_clbal
             then
               if :GLOBAL.verify = 'NO'
                 then
                   :GLOBAL.verify := 'YES';
                   message('Closing Balance does not add up -- CHECK VALUE!'
                   bell;
                   raise FORM_TRIGGER_FAILURE;
               end if;
           end if;
           if :GLOBAL.verify <> 'OFF'
             then :GLOBAL.verify := 'NO';
           end if;
         End;
         >>>

      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = CT_UNOPN
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 19
      COLUMN = 15
      HELP = Enter value for : CT_UNOPN
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = CT_DSTRY
      DATATYPE = RINT
      LENGTH = 12
```

```
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 19
      COLUMN = 28
      HELP = Enter value for : CT_DSTRY
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = KEY-NXTFLD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         :datablk.ct_uncls := :datablk.ct_unopn - :datablk.ct_dstry;
         NEXT_FIELD;
         >>>

      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = DP_OPBAL
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 13
      COLUMN = 15
      HELP = Enter value for : DP_OPBAL
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = ON-VALIDATE-FIELD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         Declare
           prevbal    number(10);
         Begin
           select dispense.clos_bal
             into prevbal
             from lmis.dispense
            where dispense.fac_code = :datablk.fac_code and
                  dispense.yymm = :GLOBAL.prevdate and
                  dispense.prod_code = 5;
```

```
            if prevbal <> :datablk.dp_opbal
              then
                if :GLOBAL.verify = 'NO'
                  then
                    :GLOBAL.verify := 'YES';
                    message
        ('Opening Balance not equal to previous Closing Balance -- CHECK VA
                    bell;
                    raise FORM_TRIGGER_FAILURE;
                end if;
            end if;
            if :GLOBAL.verify <> 'OFF'
              then :GLOBAL.verify := 'NO';
            end if;
          Exception
            when NO_DATA_FOUND then
              :GLOBAL.no_bal := '1';
          End;
          >>>

      ENDDEFINE TRIGGER

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = DP_RECPT
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 13
      COLUMN = 28
      HELP = Enter value for : DP_RECPT
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = DP_ISSUE
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      MANDATORY = ON
      PAGE = 1
      LINE = 13
      COLUMN = 41
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
```

```
      DEFAULT = 0
      AUTOHELP = ON

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = DP_FOUND
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 13
      COLUMN = 54
      HELP = Enter value for : DP_FOUND
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = DP_CLBAL
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 13
      COLUMN = 67
      HELP = Enter value for : DP_CLBAL
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = ON-VALIDATE-FIELD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         Begin
           if ((:datablk.dp_opbal + :datablk.dp_recpt) -
               (:datablk.dp_issue - :datablk.dp_found))
             <> :datablk.dp_clbal
             then
               if :GLOBAL.verify = 'NO'
                 then
                   :GLOBAL.verify := 'YES';
                   message('Closing Balance does not add up -- CHECK VALUE!')
                   bell;
                   raise FORM_TRIGGER_FAILURE;
```

```
                end if;
            end if;
            if :GLOBAL.verify <> 'OFF'
              then :GLOBAL.verify := 'NO';
            end if;
          End;
          >>>

      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = DP_UNOPN
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 20
      COLUMN = 15
      HELP = Enter value for : DP_UNOPN
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = DP_DSTRY
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 20
      COLUMN = 28
      HELP = Enter value for : DP_DSTRY
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = KEY-NXTFLD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         :datablk.dp_uncls := :datablk.dp_unopn - :datablk.dp_dstry;
         NEXT_FIELD;
         >>>
```

```
      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = NS_OPBAL
      DATATYPE = RINT
      LENGTH = 12
      DISPLAY_LENGTH = 12
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 14
      COLUMN = 15
      HELP = Enter value for : NS_OPBAL
      INPUT_MASK = 999,999,999
      OUTPUT_MASK = 999,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = ON-VALIDATE-FIELD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         Declare
           prevbal    number(10);
         Begin
           select dispense.clos_bal
             into prevbal
             from lmis.dispense
            where dispense.fac_code = :datablk.fac_code and
                  dispense.yymm = :GLOBAL.prevdate and
                  dispense.prod_code = 6;
           if prevbal <> :datablk.ns_opbal
             then
               if :GLOBAL.verify = 'NO'
                 then
                   :GLOBAL.verify := 'YES';
                   message
         ('Opening Balance not equal to previous Closing Balance -- CHECK VA
                   bell;
                   raise FORM_TRIGGER_FAILURE;
               end if;
           end if;
           if :GLOBAL.verify <> 'OFF'
             then :GLOBAL.verify := 'NO';
           end if;
         Exception
           when NO_DATA_FOUND then
             :GLOBAL.no_bal := '1';
         End;
         >>>

      ENDDEFINE TRIGGER
```

```
ENDDEFINE FIELD

DEFINE FIELD

   NAME = NS_RECPT
   DATATYPE = RINT
   LENGTH = 12
   DISPLAY_LENGTH = 12
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 14
   COLUMN = 28
   HELP = Enter value for : NS_RECPT
   INPUT_MASK = 999,999,999
   OUTPUT_MASK = 999,999,999
   DEFAULT = 0

ENDDEFINE FIELD

DEFINE FIELD

   NAME = NS_ISSUE
   DATATYPE = RINT
   LENGTH = 12
   DISPLAY_LENGTH = 12
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   MANDATORY = ON
   PAGE = 1
   LINE = 14
   COLUMN = 41
   INPUT_MASK = 999,999,999
   OUTPUT_MASK = 999,999,999
   DEFAULT = 0
   AUTOHELP = ON

ENDDEFINE FIELD

DEFINE FIELD

   NAME = NS_FOUND
   DATATYPE = RINT
   LENGTH = 12
   DISPLAY_LENGTH = 12
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 14
   COLUMN = 54
   HELP = Enter value for : NS_FOUND
   INPUT_MASK = 999,999,999
   OUTPUT_MASK = 999,999,999
   DEFAULT = 0
```

```
ENDDEFINE FIELD

DEFINE FIELD

   NAME = NS_CLBAL
   DATATYPE = RINT
   LENGTH = 12
   DISPLAY_LENGTH = 12
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 14
   COLUMN = 67
   HELP = Enter value for : NS_CLBAL
   INPUT_MASK = 999,999,999
   OUTPUT_MASK = 999,999,999
   DEFAULT = 0

   DEFINE TRIGGER

      NAME = ON-VALIDATE-FIELD
      TRIGGER_TYPE = V3
      SHOW_KEY = OFF
      TEXT = <<<
      Begin
        if ((:datablk.ns_opbal + :datablk.ns_recpt) -
            (:datablk.ns_issue - :datablk.ns_found))
          <> :datablk.ns_clbal
          then
            if :GLOBAL.verify = 'NO'
              then
                :GLOBAL.verify := 'YES';
                message('Closing Balance does not add up -- CHECK VALUE!');
                bell;
                raise FORM_TRIGGER_FAILURE;
            end if;
        end if;
        if :GLOBAL.verify <> 'OFF'
          then :GLOBAL.verify := 'NO';
        end if;
      End;
      >>>

   ENDDEFINE TRIGGER

ENDDEFINE FIELD

DEFINE FIELD

   NAME = NS_UNOPN
   DATATYPE = RINT
   LENGTH = 12
   DISPLAY_LENGTH = 12
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
```

```
        LINE = 21
        COLUMN = 15
        HELP = Enter value for : NS_UNOPN
        INPUT_MASK = 999,999,999
        OUTPUT_MASK = 999,999,999
        DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

        NAME = NS_DSTRY
        DATATYPE = RINT
        LENGTH = 12
        DISPLAY_LENGTH = 12
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 21
        COLUMN = 28
        HELP = Enter value for : NS_DSTRY
        INPUT_MASK = 999,999,999
        OUTPUT_MASK = 999,999,999
        DEFAULT = 0

        DEFINE TRIGGER

            NAME = KEY-NXTFLD
            TRIGGER_TYPE = V3
            SHOW_KEY = OFF
            TEXT = <<<
            :datablk.ns_uncls := :datablk.ns_unopn - :datablk.ns_dstry;
            EXECUTE_TRIGGER('KEY-COMMIT');
            >>>

        ENDDEFINE TRIGGER

    ENDDEFINE FIELD

    DEFINE FIELD

        NAME = FAC_NAME
        DATATYPE = CHAR
        LENGTH = 20
        DISPLAY_LENGTH = 20
        QUERY_LENGTH = 20
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 4
        COLUMN = 50
        INPUT = OFF
        UPDATE = OFF
        QUERY = OFF

    ENDDEFINE FIELD
```

```
DEFINE FIELD

   NAME = TYPE_NAME
   DATATYPE = CHAR
   LENGTH = 20
   DISPLAY_LENGTH = 20
   QUERY_LENGTH = 20
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 5
   COLUMN = 50
   INPUT = OFF
   UPDATE = OFF
   QUERY = OFF

ENDDEFINE FIELD

DEFINE FIELD

   NAME = CD_UNCLS
   DATATYPE = RINT
   LENGTH = 12
   DISPLAY_LENGTH = 12
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 16
   COLUMN = 41
   HELP = Enter value for : CD_UNCLS
   INPUT_MASK = 999,999,999
   OUTPUT_MASK = 999,999,999
   DEFAULT = 0
   INPUT = OFF
   UPDATE = OFF
   QUERY = OFF

ENDDEFINE FIELD

DEFINE FIELD

   NAME = C5_UNCLS
   DATATYPE = RINT
   LENGTH = 12
   DISPLAY_LENGTH = 12
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 17
   COLUMN = 41
   HELP = Enter value for : C5_UNCLS
   INPUT_MASK = 999,999,999
   OUTPUT_MASK = 999,999,999
   DEFAULT = 0
   INPUT = OFF
   UPDATE = OFF
   QUERY = OFF
```

```
ENDDEFINE FIELD

DEFINE FIELD

   NAME = OV_UNCLS
   DATATYPE = RINT
   LENGTH = 12
   DISPLAY_LENGTH = 12
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 18
   COLUMN = 41
   HELP = Enter value for : OV_UNCLS
   INPUT_MASK = 999,999,999
   OUTPUT_MASK = 999,999,999
   DEFAULT = 0
   INPUT = OFF
   UPDATE = OFF
   QUERY = OFF

ENDDEFINE FIELD

DEFINE FIELD

   NAME = CT_UNCLS
   DATATYPE = RINT
   LENGTH = 12
   DISPLAY_LENGTH = 12
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 19
   COLUMN = 41
   HELP = Enter value for : CT_UNCLS
   INPUT_MASK = 999,999,999
   OUTPUT_MASK = 999,999,999
   DEFAULT = 0
   INPUT = OFF
   UPDATE = OFF
   QUERY = OFF

ENDDEFINE FIELD

DEFINE FIELD

   NAME = DP_UNCLS
   DATATYPE = RINT
   LENGTH = 12
   DISPLAY_LENGTH = 12
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 20
   COLUMN = 41
```

```
        HELP = Enter value for : DP_UNCLS
        INPUT_MASK = 999,999,999
        OUTPUT_MASK = 999,999,999
        DEFAULT = 0
        INPUT = OFF
        UPDATE = OFF
        QUERY = OFF

    ENDDEFINE FIELD

    DEFINE FIELD

        NAME = NS_UNCLS
        DATATYPE = RINT
        LENGTH = 12
        DISPLAY_LENGTH = 12
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 21
        COLUMN = 41
        HELP = Enter value for : NS_UNCLS
        INPUT_MASK = 999,999,999
        OUTPUT_MASK = 999,999,999
        DEFAULT = 0
        INPUT = OFF
        UPDATE = OFF
        QUERY = OFF

    ENDDEFINE FIELD

ENDDEFINE BLOCK


DEFINE SCREEN

    DEFINE PAGE

        PAGE = 1
        PAGE_XS = 0
        PAGE_YS = 0
        PAGE_PX0 = 0
        PAGE_PY0 = 0
        PAGE_PXS = 0
        PAGE_PYS = 0
        PAGE_SX0 = 0
        PAGE_SY0 = 0
        MODE = TEXT
        LINE = 2
        BOILER = <<<
                                        Form 7 Data Entry Screen
        >>>
        LINE = 4
        BOILER = <<<
                  Facility Code:                    Name:
          Year and Month (YYMM):                    Type:
```

```
>>>
LINE = 7
BOILER = <<<
                    Opening                                  Found        C
                    Balance     Recieved      Issued       Unusable      B
 1. Condoms
 2. OPC-5
 3. Ovral
 4. Copper-T
 5. Depo-Prov
 6. Norist.
                 Unus. Open   Unus. Destr. Unus. Clos.
 1. Condoms
 2. OPC-5
 3. Ovral
 4. Copper-T
 5. Depo-Prov
 6. Norist.
>>>
MODE = BOX
BOILER = <<<
p-------------------------------------------------------------------
|
(-------------------------------------------------------------------
|
|
(-------------------------------------------------------------------
|
|
|
|
|
|
|
|
|
|
|
|
|
|
b-------------------------------------------------------------------
>>>

ENDDEFINE PAGE

DEFINE PAGE

   PAGE = 2
   PAGE_XS = 0
   PAGE_YS = 0
   PAGE_PX0 = 0
   PAGE_PY0 = 0
   PAGE_PXS = 0
   PAGE_PYS = 0
```

```
        PAGE_SX0 = 0
        PAGE_SY0 = 0

    ENDDEFINE PAGE

    DEFINE PAGE

        PAGE = 3
        PAGE_XS = 0
        PAGE_YS = 0
        PAGE_PX0 = 0
        PAGE_PY0 = 0
        PAGE_PXS = 0
        PAGE_PYS = 0
        PAGE_SX0 = 0
        PAGE_SY0 = 0

    ENDDEFINE PAGE

    DEFINE PAGE

        PAGE = 4
        PAGE_XS = 0
        PAGE_YS = 0
        PAGE_PX0 = 0
        PAGE_PY0 = 0
        PAGE_PXS = 0
        PAGE_PYS = 0
        PAGE_SX0 = 0
        PAGE_SY0 = 0

    ENDDEFINE PAGE

ENDDEFINE SCREEN

)DEFINE FORM
```

ANNEX 3

Form 7b Source Code

```
 Copyright (c) 1988 by the Oracle Corporation */


L*FORMS_VERSION = 03.00.16.08.01
RSE = ON

FINE FORM

 COMMENT = <<<
 This version uses the table F7B_BUFFER to store entered values before being
  written to the DISPENSE table.
 >>>

 NAME = form7b
 TITLE = Form 7b Data Entry Screen
 DEFAULT_MENU_APPLICATION = DEFAULT

 DEFINE TRIGGER

    NAME = ON-ERROR
    TRIGGER_TYPE = V3
    SHOW_KEY = OFF
    TEXT = <<<
    if ERROR_CODE = 40401
      then message(:GLOBAL.status || ' - Form 7b Data Saved (V1.4)');
      else message('FRM-' || to_char(ERROR_CODE) || ': ' || ERROR_TEXT);
    end if;
    >>>

 ENDDEFINE TRIGGER

 DEFINE BLOCK

    NAME = datablk
    DESCRIPTION = datablk
    ROWS_DISPLAYED = 1
    ROWS_BUFFERED = 1
    BASE_LINE = 1
    LINES_PER_ROW = 14
    ARRAY_SIZE = 1

    DEFINE TRIGGER

       NAME = KEY-CLRFRM
       TRIGGER_TYPE = V3
       TEXT = <<<
       CLEAR_FORM(no_validate);
       >>>

    ENDDEFINE TRIGGER

    DEFINE TRIGGER

       NAME = KEY-COMMIT
       TRIGGER_TYPE = V3
```

```
SHOW_KEY = OFF
TEXT = <<<
Declare
  srch_code   char(4);
  srch_yymm   char(4);
Begin
/**-------------------------------------------------------------------**
 ** User has committed SAVE THE DATA! Store in dispense file          **
 **-------------------------------------------------------------------**/
  :GLOBAL.verify := 'OFF';              -- ** Turn off Validation
  :datablk.cd_recpt := NVL(:datablk.cd_recpt,0);
  :datablk.cd_issue := NVL(:datablk.cd_issue,0);
  :datablk.cd_retrn := NVL(:datablk.cd_retrn,0);
  :datablk.cd_opbal := NVL(:datablk.cd_opbal,0);
  :datablk.cd_clbal := NVL(:datablk.cd_clbal,0);
  :datablk.cd_field := NVL(:datablk.cd_field,0);
  :datablk.fcd_recpt := NVL(:datablk.fcd_recpt,0);
  :datablk.fcd_issue := NVL(:datablk.fcd_issue,0);
  :datablk.fcd_retrn := NVL(:datablk.fcd_retrn,0);
  :datablk.fcd_opbal := NVL(:datablk.fcd_opbal,0);
  :datablk.fcd_clbal := NVL(:datablk.fcd_clbal,0);
  :datablk.c5_recpt := NVL(:datablk.c5_recpt,0);
  :datablk.c5_issue := NVL(:datablk.c5_issue,0);
  :datablk.c5_retrn := NVL(:datablk.c5_retrn,0);
  :datablk.c5_opbal := NVL(:datablk.c5_opbal,0);
  :datablk.c5_clbal := NVL(:datablk.c5_clbal,0);
  :datablk.c5_field := NVL(:datablk.c5_field,0);
  :datablk.fc5_recpt := NVL(:datablk.fc5_recpt,0);
  :datablk.fc5_issue := NVL(:datablk.fc5_issue,0);
  :datablk.fc5_retrn := NVL(:datablk.fc5_retrn,0);
  :datablk.fc5_opbal := NVL(:datablk.fc5_opbal,0);
  :datablk.fc5_clbal := NVL(:datablk.fc5_clbal,0);
  :datablk.ov_recpt := NVL(:datablk.ov_recpt,0);
  :datablk.ov_issue := NVL(:datablk.ov_issue,0);
  :datablk.ov_retrn := NVL(:datablk.ov_retrn,0);
  :datablk.ov_opbal := NVL(:datablk.ov_opbal,0);
  :datablk.ov_clbal := NVL(:datablk.ov_clbal,0);
  :datablk.ov_field := NVL(:datablk.ov_field,0);
  :datablk.fov_recpt := NVL(:datablk.fov_recpt,0);
  :datablk.fov_issue := NVL(:datablk.fov_issue,0);
  :datablk.fov_retrn := NVL(:datablk.fov_retrn,0);
  :datablk.fov_opbal := NVL(:datablk.fov_opbal,0);
  :datablk.fov_clbal := NVL(:datablk.fov_clbal,0);
  :datablk.ct_recpt := NVL(:datablk.ct_recpt,0);
  :datablk.ct_issue := NVL(:datablk.ct_issue,0);
  :datablk.ct_retrn := NVL(:datablk.ct_retrn,0);
  :datablk.ct_opbal := NVL(:datablk.ct_opbal,0);
  :datablk.ct_clbal := NVL(:datablk.ct_clbal,0);
  :datablk.ct_field := NVL(:datablk.ct_field,0);
  :datablk.fct_recpt := NVL(:datablk.fct_recpt,0);
  :datablk.fct_issue := NVL(:datablk.fct_issue,0);
  :datablk.fct_retrn := NVL(:datablk.fct_retrn,0);
  :datablk.fct_opbal := NVL(:datablk.fct_opbal,0);
  :datablk.fct_clbal := NVL(:datablk.fct_clbal,0);
  :datablk.dp_recpt := NVL(:datablk.dp_recpt,0);
  :datablk.dp_issue := NVL(:datablk.dp_issue,0);
```

```
:datablk.dp_retrn := NVL(:datablk.dp_retrn,0);
:datablk.dp_opbal := NVL(:datablk.dp_opbal,0);
:datablk.dp_clbal := NVL(:datablk.dp_clbal,0);
:datablk.dp_field := NVL(:datablk.dp_field,0);
:datablk.fdp_recpt := NVL(:datablk.fdp_recpt,0);
:datablk.fdp_issue := NVL(:datablk.fdp_issue,0);
:datablk.fdp_retrn := NVL(:datablk.fdp_retrn,0);
:datablk.fdp_opbal := NVL(:datablk.fdp_opbal,0);
:datablk.fdp_clbal := NVL(:datablk.fdp_clbal,0);
:datablk.ns_recpt := NVL(:datablk.ns_recpt,0);
:datablk.ns_issue := NVL(:datablk.ns_issue,0);
:datablk.ns_retrn := NVL(:datablk.ns_retrn,0);
:datablk.ns_opbal := NVL(:datablk.ns_opbal,0);
:datablk.ns_clbal := NVL(:datablk.ns_clbal,0);
:datablk.ns_field := NVL(:datablk.ns_field,0);
:datablk.fns_recpt := NVL(:datablk.fns_recpt,0);
:datablk.fns_issue := NVL(:datablk.fns_issue,0);
:datablk.fns_retrn := NVL(:datablk.fns_retrn,0);
:datablk.fns_opbal := NVL(:datablk.fns_opbal,0);
:datablk.fns_clbal := NVL(:datablk.fns_clbal,0);
srch_code   := :datablk.fac_code;
srch_yymm   := :datablk.yymm;
if :GLOBAL.status = 'NEW'
  then
    insert into lmis.dispense
    values (srch_code,srch_yymm,1,
            :datablk.cd_recpt,:datablk.cd_issue,:datablk.cd_retrn,
            :datablk.cd_opbal,:datablk.cd_clbal,:datablk.cd_field,
            0,0,0);
    insert into lmis.dispense
    values (srch_code,srch_yymm,2,
            :datablk.c5_recpt,:datablk.c5_issue,:datablk.c5_retrn,
            :datablk.c5_opbal,:datablk.c5_clbal,:datablk.c5_field,
            0,0,0);
    insert into lmis.dispense
    values (srch_code,srch_yymm,3,
            :datablk.ov_recpt,:datablk.ov_issue,:datablk.ov_retrn,
            :datablk.ov_opbal,:datablk.ov_clbal,:datablk.ov_field,
            0,0,0);
    insert into lmis.dispense
    values (srch_code,srch_yymm,4,
            :datablk.ct_recpt,:datablk.ct_issue,:datablk.ct_retrn,
            :datablk.ct_opbal,:datablk.ct_clbal,:datablk.ct_field,
            0,0,0);
    insert into lmis.dispense
    values (srch_code,srch_yymm,5,
            :datablk.dp_recpt,:datablk.dp_issue,:datablk.dp_retrn,
            :datablk.dp_opbal,:datablk.dp_clbal,:datablk.dp_field,
            0,0,0);
    insert into lmis.dispense
    values (srch_code,srch_yymm,6,
            :datablk.ns_recpt,:datablk.ns_issue,:datablk.ns_retrn,
            :datablk.ns_opbal,:datablk.ns_clbal,:datablk.ns_field,
            0,0,0);
    srch_code := 'F'||substr(srch_code,2,3);  -- ** Save the FIELD
    insert into lmis.dispense
```

```
        values (srch_code,srch_yymm,1,
                :datablk.fcd_recpt,:datablk.fcd_issue,:datablk.fcd_retr
                :datablk.fcd_opbal,:datablk.fcd_clbal,0,0,0,0);
        insert into lmis.dispense
        values (srch_code,srch_yymm,2,
                :datablk.fc5_recpt,:datablk.fc5_issue,:datablk.fc5_retr
                :datablk.fc5_opbal,:datablk.fc5_clbal,0,0,0,0);
        insert into lmis.dispense
        values (srch_code,srch_yymm,3,
                :datablk.fov_recpt,:datablk.fov_issue,:datablk.fov_retr
                :datablk.fov_opbal,:datablk.fov_clbal,0,0,0,0);
        insert into lmis.dispense
        values (srch_code,srch_yymm,4,
                :datablk.fct_recpt,:datablk.fct_issue,:datablk.fct_retr
                :datablk.fct_opbal,:datablk.fct_clbal,0,0,0,0);
        insert into lmis.dispense
        values (srch_code,srch_yymm,5,
                :datablk.fdp_recpt,:datablk.fdp_issue,:datablk.fdp_retr
                :datablk.fdp_opbal,:datablk.fdp_clbal,0,0,0,0);
        insert into lmis.dispense
        values (srch_code,srch_yymm,6,
                :datablk.fns_recpt,:datablk.fns_issue,:datablk.fns_retr
                :datablk.fns_opbal,:datablk.fns_clbal,0,0,0,0);
      else
        update lmis.dispense
          set open_bal = :datablk.cd_opbal,
              receipts = :datablk.cd_recpt,
              issues   = :datablk.cd_issue,
              returns  = :datablk.cd_retrn,
              fieldret = :datablk.cd_field,
              clos_bal = :datablk.cd_clbal
          where dispense.fac_code  = srch_code and
                dispense.yymm      = srch_yymm and
                dispense.prod_code = 1;                -- ** Condoms
        update lmis.dispense
          set open_bal = :datablk.c5_opbal,
              receipts = :datablk.c5_recpt,
              issues   = :datablk.c5_issue,
              returns  = :datablk.c5_retrn,
              fieldret = :datablk.c5_field,
              clos_bal = :datablk.c5_clbal
          where dispense.fac_code  = srch_code and
                dispense.yymm      = srch_yymm and
                dispense.prod_code = 2;                -- ** OPC-5
        update lmis.dispense
          set open_bal = :datablk.ov_opbal,
              receipts = :datablk.ov_recpt,
              issues   = :datablk.ov_issue,
              returns  = :datablk.ov_retrn,
              fieldret = :datablk.ov_field,
              clos_bal = :datablk.ov_clbal
          where dispense.fac_code  = srch_code and
                dispense.yymm      = srch_yymm and
                dispense.prod_code = 3;                -- ** Ovral
        update lmis.dispense
          set open_bal = :datablk.ct_opbal,
```

```
            receipts = :datablk.ct_recpt,
            issues   = :datablk.ct_issue,
            returns  = :datablk.ct_retrn,
            fieldret = :datablk.ct_field,
            clos_bal = :datablk.ct_clbal
        where dispense.fac_code  = srch_code and
              dispense.yymm      = srch_yymm and
              dispense.prod_code = 4;                -- ** Copper-T
      update lmis.dispense
        set open_bal = :datablk.dp_opbal,
            receipts = :datablk.dp_recpt,
            issues   = :datablk.dp_issue,
            returns  = :datablk.dp_retrn,
            fieldret = :datablk.dp_field,
            clos_bal = :datablk.dp_clbal
        where dispense.fac_code  = srch_code and
              dispense.yymm      = srch_yymm and
              dispense.prod_code = 5;                -- ** Deprovera
      update lmis.dispense
        set open_bal = :datablk.ns_opbal,
            receipts = :datablk.ns_recpt,
            issues   = :datablk.ns_issue,
            returns  = :datablk.ns_retrn,
            fieldret = :datablk.ns_field,
            clos_bal = :datablk.ns_clbal
        where dispense.fac_code  = srch_code and
              dispense.yymm      = srch_yymm and
              dispense.prod_code = 6;                -- ** Noristerat
      srch_code := 'F'||substr(srch_code,2,3);
      update lmis.dispense
        set open_bal = :datablk.fcd_opbal,
            receipts = :datablk.fcd_recpt,
            issues   = :datablk.fcd_issue,
            returns  = :datablk.fcd_retrn,
            clos_bal = :datablk.fcd_clbal
        where dispense.fac_code  = srch_code and
              dispense.yymm      = srch_yymm and
              dispense.prod_code = 1;                -- ** Condoms
      update lmis.dispense
        set open_bal = :datablk.fc5_opbal,
            receipts = :datablk.fc5_recpt,
            issues   = :datablk.fc5_issue,
            returns  = :datablk.fc5_retrn,
            clos_bal = :datablk.fc5_clbal
        where dispense.fac_code  = srch_code and
              dispense.yymm      = srch_yymm and
              dispense.prod_code = 2;                -- ** OPC-5
      update lmis.dispense
        set open_bal = :datablk.fov_opbal,
            receipts = :datablk.fov_recpt,
            issues   = :datablk.fov_issue,
            returns  = :datablk.fov_retrn,
            clos_bal = :datablk.fov_clbal
        where dispense.fac_code  = srch_code and
              dispense.yymm      = srch_yymm and
              dispense.prod_code = 3;                -- ** Ovral
```

```
            update lmis.dispense
              set open_bal = :datablk.fct_opbal,
                  receipts = :datablk.fct_recpt,
                  issues   = :datablk.fct_issue,
                  returns  = :datablk.fct_retrn,
                  clos_bal = :datablk.fct_clbal
              where dispense.fac_code  = srch_code and
                    dispense.yymm      = srch_yymm and
                    dispense.prod_code = 4;              -- ** Copper-T
            update lmis.dispense
              set open_bal = :datablk.fdp_opbal,
                  receipts = :datablk.fdp_recpt,
                  issues   = :datablk.fdp_issue,
                  returns  = :datablk.fdp_retrn,
                  clos_bal = :datablk.fdp_clbal
              where dispense.fac_code  = srch_code and
                    dispense.yymm      = srch_yymm and
                    dispense.prod_code = 5;              -- ** Deprovera
            update lmis.dispense
              set open_bal = :datablk.fns_opbal,
                  receipts = :datablk.fns_recpt,
                  issues   = :datablk.fns_issue,
                  returns  = :datablk.fns_retrn,
                  clos_bal = :datablk.fns_clbal
              where dispense.fac_code  = srch_code and
                    dispense.yymm      = srch_yymm and
                    dispense.prod_code = 6;             -- ** Noristerat
          end if;
          if :GLOBAL.status = 'NEW'                     -- ** Add Audit Inf.
            then
              insert into lmis.f7_audit
              values (:datablk.fac_code, :datablk.yymm,
                      USER, SYSDATE, USER, SYSDATE);
              insert into lmis.f7_audit
              values (srch_code, :datablk.yymm,
                      USER, SYSDATE, USER, SYSDATE);
            else
              update lmis.f7_audit
                 set mdop_code = USER, md_date = SYSDATE
               where f7_audit.fac_code = :datablk.fac_code and
                     f7_audit.yymm = :datablk.yymm;
              update lmis.f7_audit
                 set mdop_code = USER, md_date = SYSDATE
               where f7_audit.fac_code = srch_code and
                     f7_audit.yymm = :datablk.yymm;
          end if;
          COMMIT;
          CLEAR_FORM(NO_VALIDATE);
        END;
        >>>

      ENDDEFINE TRIGGER

      DEFINE TRIGGER

        NAME = KEY-DOWN
```

```
    TRIGGER_TYPE = V3
    SHOW_KEY = OFF
    TEXT = <<<
    Declare
      src_fld char(3);
      nxt_fld char(4);
      tar_fld char(10);
      bas_fld char(5);
    Begin
      nxt_fld := 'NULL';
      src_fld := substr(:SYSTEM.cursor_field,9,3);
      if substr(src_fld,1,1) = 'F'
        then bas_fld := substr(:SYSTEM.cursor_field,13,5);
        else bas_fld := substr(:SYSTEM.cursor_field,12,5);
      end if;
      if    src_fld = 'CD_' then nxt_fld := 'C5_';
      elsif src_fld = 'C5_' then nxt_fld := 'OV_';
      elsif src_fld = 'OV_' then nxt_fld := 'CT_';
      elsif src_fld = 'CT_' then nxt_fld := 'DP_';
      elsif src_fld = 'DP_' then nxt_fld := 'NS_';
      elsif src_fld = 'NS_' then nxt_fld := 'FCD_';
      elsif src_fld = 'FCD' then nxt_fld := 'FC5_';
      elsif src_fld = 'FC5' then nxt_fld := 'FOV_';
      elsif src_fld = 'FOV' then nxt_fld := 'FCT_';
      elsif src_fld = 'FCT' then nxt_fld := 'FDP_';
      elsif src_fld = 'FDP' then nxt_fld := 'FNS_';
      else nxt_fld := 'NULL';
      end if;
      if :SYSTEM.cursor_field = 'DATABLK.NS_FIELD'
        then nxt_fld := 'NULL';
      end if;
      if nxt_fld <> 'NULL'
        then
          tar_fld := nxt_fld || bas_fld;
          GO_FIELD(tar_fld);
        else
          next_field;
      end if;
    END;
    >>>

  ENDDEFINE TRIGGER

  DEFINE TRIGGER

    NAME = KEY-NXTREC
    TRIGGER_TYPE = V3
    TEXT = <<<
    EXECUTE_TRIGGER('KEY-DOWN');
    >>>

  ENDDEFINE TRIGGER

  DEFINE TRIGGER

    NAME = KEY-PRVREC
```

```
        TRIGGER_TYPE = V3
        SHOW_KEY = OFF
        TEXT = <<<
        EXECUTE_TRIGGER('KEY-UP');
        >>>

     ENDDEFINE TRIGGER

     DEFINE TRIGGER

        NAME = KEY-UP
        TRIGGER_TYPE = V3
        SHOW_KEY = OFF
        TEXT = <<<
        Declare
          src_fld char(3);
          nxt_fld char(4);
          tar_fld char(10);
          bas_fld char(5);
        Begin
          nxt_fld := 'NULL';
          src_fld := substr(:SYSTEM.cursor_field,9,3);
          if substr(src_fld,1,1) = 'F'
            then bas_fld := substr(:SYSTEM.cursor_field,13,5);
            else bas_fld := substr(:SYSTEM.cursor_field,12,5);
          end if;
          if    src_fld = 'C5_' then nxt_fld := 'CD_';
          elsif src_fld = 'OV_' then nxt_fld := 'C5_';
          elsif src_fld = 'CT_' then nxt_fld := 'OV_';
          elsif src_fld = 'DP_' then nxt_fld := 'CT_';
          elsif src_fld = 'NS_' then nxt_fld := 'DP_';
          elsif src_fld = 'FCD' then nxt_fld := 'NS_';
          elsif src_fld = 'FC5' then nxt_fld := 'FCD_';
          elsif src_fld = 'FOV' then nxt_fld := 'FC5_';
          elsif src_fld = 'FCT' then nxt_fld := 'FOV_';
          elsif src_fld = 'FDP' then nxt_fld := 'FCT_';
          elsif src_fld = 'FNS' then nxt_fld := 'FDP_';
          else nxt_fld := 'NULL';
          end if;
          if nxt_fld <> 'NULL'
            then
              tar_fld := nxt_fld || bas_fld;
              GO_FIELD(tar_fld);
            else
              next_field;
          end if;
        END;
        >>>

     ENDDEFINE TRIGGER

     DEFINE FIELD

        NAME = FAC_CODE
        DATATYPE = CHAR
        LENGTH = 4
```

```
DISPLAY_LENGTH = 4
QUERY_LENGTH = 4
BASE_TABLE = OFF
MANDATORY = ON
PAGE = 1
LINE = 4
COLUMN = 26
HELP = Enter the Facility Code (Must be a THANA Code i.e. T001)
LOV_TEXT = <<<
select fac_name, fac_code, type_name
  into :datablk.fac_name, :datablk.fac_code, :datablk.type_name
  from lmis.facility,lmis.fac_types
 where substr(facility.fac_code,1,1) = fac_types.type_code
       and
       substr(facility.fac_code,1,1) = 'T'
order by fac_name
>>>
LOV_TITLE = Facilities Listing
LOV_X = 15
LOV_Y = 6
AUTOSKIP = ON
AUTOHELP = ON

DEFINE TRIGGER

   NAME = ON-VALIDATE-FIELD
   TRIGGER_TYPE = V3
   SHOW_KEY = OFF
   TEXT = <<<
   Begin
     :GLOBAL.verify   := 'NO';
     :GLOBAL.status   := 'NEW';
     :GLOBAL.fld_code := 'F';
     :GLOBAL.no_bal := '1';
     :datablk.fac_code := upper(:datablk.fac_code);
     if substr(:datablk.fac_code,1,1) <> 'T'
       then
         message('Only Thanas may submit FORM 7Bs!');
         BELL;
         raise FORM_TRIGGER_FAILURE;
     end if;
     select facility.fac_name,
            fac_types.type_name
       into :datablk.fac_name,:datablk.type_name
       from lmis.facility,lmis.fac_types
      where facility.fac_code = :datablk.fac_code
            and
            fac_types.type_code = substr(facility.fac_code,1,1);
  Exception
    when NO_DATA_FOUND then
      BELL;
      message('Facility Code Not Found In Facility File - Press [LIST
      raise FORM_TRIGGER_FAILURE;
  End;
  >>>
```

```
      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = YYMM
      DATATYPE = CHAR
      LENGTH = 4
      DISPLAY_LENGTH = 4
      QUERY_LENGTH = 4
      BASE_TABLE = OFF
      MANDATORY = ON
      PAGE = 1
      LINE = 5
      COLUMN = 26
      HELP = Enter the Processing Year and Month. i.e. January 1993 = 9301
      AUTOHELP = ON

      DEFINE TRIGGER

         NAME = ON-VALIDATE-FIELD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         Begin
           if substr(:datablk.yymm,3,2) > '12' or
              substr(:datablk.yymm,3,2) < '01' or
              length(:datablk.yymm) < 4
            then
              message('Invalid Month Code - Try Again!');
              raise FORM_TRIGGER_FAILURE;
           end if;
         End;
         >>>

      ENDDEFINE TRIGGER

      DEFINE TRIGGER

         NAME = KEY-NXTFLD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         Declare
         /* */
           prevyy      number(2);
           prevmm      number(2);
           prevnum     number(4);
           srch_code    char(4);
           srch_yymm    char(4);
           update_flag number(1);
         /* */
           CURSOR disp is select dispense.prod_code,
                                 dispense.open_bal,
                                 dispense.receipts,
```

```
                                dispense.fieldret,
                                dispense.issues,
                                dispense.returns,
                                dispense.clos_bal
                           from lmis.dispense
                          where dispense.fac_code  = srch_code and
                                dispense.yymm      = srch_yymm;
/* */
Begin
  :GLOBAL.verify   := 'NO';
  :GLOBAL.status   := 'NEW';
  :GLOBAL.fld_code := 'F' || substr(:datablk.fac_code,2,3);
  prevyy := to_number(substr(:datablk.yymm,1,2));
  prevmm := to_number(substr(:datablk.yymm,3,2));
  if prevmm = 1
    then
      prevmm := 12;
      prevyy := prevyy - 1;
    else
      prevmm := prevmm - 1;
  end if;
  prevnum  := (prevyy*100)+prevmm;
  :GLOBAL.prevdate := substr(to_char(prevnum,'9999'),2,4);
/**-----------------------------------------------------------------------
 ** Search for Previous F7b Data if none found then we have a new F7
 **-----------------------------------------------------------------------
  srch_code   := :datablk.fac_code;
  srch_yymm   := :datablk.yymm;
  update_flag := 0;
  For disprec in disp LOOP
    update_flag := 1;              -- ** Got a previously stored form
    if disprec.prod_code = 1       -- ** Condoms
      then
        :datablk.cd_opbal := disprec.open_bal;
        :datablk.cd_recpt := disprec.receipts;
        :datablk.cd_field := NVL(disprec.fieldret,0);
        :datablk.cd_issue := disprec.issues;
        :datablk.cd_retrn := disprec.returns;
        :datablk.cd_clbal := disprec.clos_bal;
    elsif disprec.prod_code = 2  -- ** OPC-5
      then
        :datablk.c5_opbal := disprec.open_bal;
        :datablk.c5_recpt := disprec.receipts;
        :datablk.c5_field := NVL(disprec.fieldret,0);
        :datablk.c5_issue := disprec.issues;
        :datablk.c5_retrn := disprec.returns;
        :datablk.c5_clbal := disprec.clos_bal;
    elsif disprec.prod_code = 3  -- ** OVRAL
      then
        :datablk.ov_opbal := disprec.open_bal;
        :datablk.ov_recpt := disprec.receipts;
        :datablk.ov_field := NVL(disprec.fieldret,0);
        :datablk.ov_issue := disprec.issues;
        :datablk.ov_retrn := disprec.returns;
        :datablk.ov_clbal := disprec.clos_bal;
    elsif disprec.prod_code = 4  -- ** COPPER-T
```

```
          then
            :datablk.ct_opbal := disprec.open_bal;
            :datablk.ct_recpt := disprec.receipts;
            :datablk.ct_field := NVL(disprec.fieldret,0);
            :datablk.ct_issue := disprec.issues;
            :datablk.ct_retrn := disprec.returns;
            :datablk.ct_clbal := disprec.clos_bal;
        elsif disprec.prod_code = 5  -- ** DEPO-PROVERA
          then
            :datablk.dp_opbal := disprec.open_bal;
            :datablk.dp_recpt := disprec.receipts;
            :datablk.dp_field := NVL(disprec.fieldret,0);
            :datablk.dp_issue := disprec.issues;
            :datablk.dp_retrn := disprec.returns;
            :datablk.dp_clbal := disprec.clos_bal;
        elsif disprec.prod_code = 6  -- ** NORISTERAT
          then
            :datablk.ns_opbal := disprec.open_bal;
            :datablk.ns_recpt := disprec.receipts;
            :datablk.ns_field := NVL(disprec.fieldret,0);
            :datablk.ns_issue := disprec.issues;
            :datablk.ns_retrn := disprec.returns;
            :datablk.ns_clbal := disprec.clos_bal;
        END IF;
      END LOOP;
    /**-----------------------------------------------------------------
     ** If we found the THANA data there must be a FIELD
     **-----------------------------------------------------------------
      if update_flag = 1
        then
          :GLOBAL.status := 'OLD';
          srch_code   := 'F' || substr(:datablk.fac_code,2,3);
          srch_yymm   := :datablk.yymm;
          For disprec in disp LOOP
            if disprec.prod_code = 1     -- ** Condoms
              then
                :datablk.fcd_opbal := disprec.open_bal;
                :datablk.fcd_recpt := disprec.receipts;
                :datablk.fcd_issue := disprec.issues;
                :datablk.fcd_retrn := disprec.returns;
                :datablk.fcd_clbal := disprec.clos_bal;
            elsif disprec.prod_code = 2  -- ** OPC-5
              then
                :datablk.fc5_opbal := disprec.open_bal;
                :datablk.fc5_recpt := disprec.receipts;
                :datablk.fc5_issue := disprec.issues;
                :datablk.fc5_retrn := disprec.returns;
                :datablk.fc5_clbal := disprec.clos_bal;
            elsif disprec.prod_code = 3  -- ** OVRAL
              then
                :datablk.fov_opbal := disprec.open_bal;
                :datablk.fov_recpt := disprec.receipts;
                :datablk.fov_issue := disprec.issues;
                :datablk.fov_retrn := disprec.returns;
                :datablk.fov_clbal := disprec.clos_bal;
            elsif disprec.prod_code = 4  -- ** COPPER-T
```

```
                then
                  :datablk.fct_opbal := disprec.open_bal;
                  :datablk.fct_recpt := disprec.receipts;
                  :datablk.fct_issue := disprec.issues;
                  :datablk.fct_retrn := disprec.returns;
                  :datablk.fct_clbal := disprec.clos_bal;
              elsif disprec.prod_code = 5  -- ** DEPO-PROVERA
                then
                  :datablk.fdp_opbal := disprec.open_bal;
                  :datablk.fdp_recpt := disprec.receipts;
                  :datablk.fdp_issue := disprec.issues;
                  :datablk.fdp_retrn := disprec.returns;
                  :datablk.fdp_clbal := disprec.clos_bal;
              elsif disprec.prod_code = 6  -- ** NORISTERAT
                then
                  :datablk.fns_opbal := disprec.open_bal;
                  :datablk.fns_recpt := disprec.receipts;
                  :datablk.fns_issue := disprec.issues;
                  :datablk.fns_retrn := disprec.returns;
                  :datablk.fns_clbal := disprec.clos_bal;
              END IF;
            END LOOP;
          END IF;
          GO_FIELD('CD_OPBAL');
        END;
        >>>

    ENDDEFINE TRIGGER

  ENDDEFINE FIELD

  DEFINE FIELD

    NAME = CD_OPBAL
    DATATYPE = RINT
    LENGTH = 9
    DISPLAY_LENGTH = 9
    QUERY_LENGTH = 12
    BASE_TABLE = OFF
    PAGE = 1
    LINE = 9
    COLUMN = 15
    HELP = Enter value for : CD_OPBAL
    INPUT_MASK = 9,999,999
    OUTPUT_MASK = 9,999,999
    DEFAULT = 0

    DEFINE TRIGGER

      NAME = ON-VALIDATE-FIELD
      TRIGGER_TYPE = V3
      SHOW_KEY = OFF
      TEXT = <<<
      Declare
        prevbal    number(10);
      Begin
```

```
          select dispense.clos_bal
            into prevbal
            from lmis.dispense
           where dispense.fac_code = :datablk.fac_code and
                 dispense.yymm = :GLOBAL.prevdate and
                 dispense.prod_code = 1;
          if prevbal <> :datablk.cd_opbal
            then
              if :GLOBAL.verify = 'NO'
                then
                  :GLOBAL.verify := 'YES';
                  message
       ('Opening Balance not equal to previous Closing Balance -- CHECK VA
                  bell;
                  raise FORM_TRIGGER_FAILURE;
              end if;
            end if;
            if :GLOBAL.verify <> 'OFF'
              then :GLOBAL.verify := 'NO';
            end if;
       Exception
            when NO_DATA_FOUND then
              :GLOBAL.no_bal := '1';
       End;
       >>>

    ENDDEFINE TRIGGER

  ENDDEFINE FIELD

  DEFINE FIELD

    NAME = CD_RECPT
    DATATYPE = RINT
    LENGTH = 9
    DISPLAY_LENGTH = 9
    QUERY_LENGTH = 12
    BASE_TABLE = OFF
    PAGE = 1
    LINE = 9
    COLUMN = 26
    HELP = Enter value for : CD_RECPT
    INPUT_MASK = 9,999,999
    OUTPUT_MASK = 9,999,999
    DEFAULT = 0

  ENDDEFINE FIELD

  DEFINE FIELD

    NAME = CD_FIELD
    DATATYPE = RINT
    LENGTH = 9
    DISPLAY_LENGTH = 9
    QUERY_LENGTH = 12
    BASE_TABLE = OFF
```

```
      PAGE = 1
      LINE = 9
      COLUMN = 37
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0
      AUTOHELP = ON

      DEFINE TRIGGER

         NAME = KEY-NXTFLD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         if :GLOBAL.status = 'NEW'
           then :datablk.fcd_retrn := :datablk.cd_field;
         end if;
         next_field;
         >>>

      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = CD_ISSUE
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 9
      COLUMN = 48
      HELP = Enter value for : CD_ISSUE
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = KEY-NXTFLD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         if :GLOBAL.status = 'NEW'
           then :datablk.fcd_recpt := :datablk.cd_issue;
         end if;
         next_field;
         >>>

      ENDDEFINE TRIGGER

   ENDDEFINE FIELD
```

```
DEFINE FIELD

   NAME = CD_RETRN
   DATATYPE = RINT
   LENGTH = 9
   DISPLAY_LENGTH = 9
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 9
   COLUMN = 59
   HELP = Enter value for : CD_RETRN
   INPUT_MASK = 9,999,999
   OUTPUT_MASK = 9,999,999
   DEFAULT = 0

ENDDEFINE FIELD

DEFINE FIELD

   NAME = CD_CLBAL
   DATATYPE = RINT
   LENGTH = 9
   DISPLAY_LENGTH = 9
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 9
   COLUMN = 70
   HELP = Enter value for : CD_CLBAL
   INPUT_MASK = 9,999,999
   OUTPUT_MASK = 9,999,999
   DEFAULT = 0

   DEFINE TRIGGER

      NAME = ON-VALIDATE-FIELD
      TRIGGER_TYPE = V3
      SHOW_KEY = OFF
      TEXT = <<<
      Begin
        if ((:datablk.cd_opbal + :datablk.cd_recpt + :datablk.cd_field) -
            (:datablk.cd_retrn + :datablk.cd_issue)) <> :datablk.cd_clbal
          then
            if :GLOBAL.verify = 'NO'
              then
                :GLOBAL.verify := 'YES';
                message('Closing Balance does not add up -- CHECK VALUE!'
                bell;
                raise FORM_TRIGGER_FAILURE;
            end if;
        end if;
        if :GLOBAL.verify <> 'OFF'
          then :GLOBAL.verify := 'NO';
        end if;
      End;
```

```
        >>>

      ENDDEFINE TRIGGER

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = C5_OPBAL
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 10
      COLUMN = 15
      HELP = Enter value for : C5_OPBAL
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

      DEFINE TRIGGER

        NAME = ON-VALIDATE-FIELD
        TRIGGER_TYPE = V3
        SHOW_KEY = OFF
        TEXT = <<<
        Declare
          prevbal    number(10);
        Begin
          select dispense.clos_bal
            into prevbal
            from lmis.dispense
           where dispense.fac_code = :datablk.fac_code and
                 dispense.yymm = :GLOBAL.prevdate and
                 dispense.prod_code = 2;
          if prevbal <> :datablk.c5_opbal
            then
              if :GLOBAL.verify = 'NO'
                then
                  :GLOBAL.verify := 'YES';
                  message
        ('Opening Balance not equal to previous Closing Balance -- CHECK VAL
                  bell;
                  raise FORM_TRIGGER_FAILURE;
              end if;
          end if;
          if :GLOBAL.verify <> 'OFF'
            then :GLOBAL.verify := 'NO';
          end if;
        Exception
          when NO_DATA_FOUND then
            :GLOBAL.no_bal := '1';
        End;
        >>>
```

```
      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = C5_RECPT
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 10
      COLUMN = 26
      HELP = Enter value for : C5_RECPT
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = C5_FIELD
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 10
      COLUMN = 37
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0
      AUTOHELP = ON

      DEFINE TRIGGER

         NAME = KEY-NXTFLD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         if :GLOBAL.status = 'NEW'
           then :datablk.fc5_retrn := :datablk.c5_field;
         end if;
         next_field;
         >>>

      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD
```

```
      NAME = C5_ISSUE
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 10
      COLUMN = 48
      HELP = Enter value for : C5_ISSUE
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = KEY-NXTFLD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         if :GLOBAL.status = 'NEW'
           then :datablk.fc5_recpt := :datablk.c5_issue;
         end if;
         next_field;
         >>>

      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = C5_RETRN
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 10
      COLUMN = 59
      HELP = Enter value for : C5_RETRN
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = C5_CLBAL
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
```

```
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 10
        COLUMN = 70
        HELP = Enter value for : C5_CLBAL
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

        DEFINE TRIGGER

           NAME = ON-VALIDATE-FIELD
           TRIGGER_TYPE = V3
           SHOW_KEY = OFF
           TEXT = <<<
           Begin
             if ((:datablk.c5_opbal + :datablk.c5_recpt + :datablk.c5_field) -
                 (:datablk.c5_retrn + :datablk.c5_issue)) <> :datablk.c5_clbal
               then
                 if :GLOBAL.verify = 'NO'
                   then
                     :GLOBAL.verify := 'YES';
                     message('Closing Balance does not add up -- CHECK VALUE!'
                     bell;
                     raise FORM_TRIGGER_FAILURE;
                 end if;
             end if;
             if :GLOBAL.verify <> 'OFF'
               then :GLOBAL.verify := 'NO';
             end if;
           End;
           >>>

        ENDDEFINE TRIGGER

     ENDDEFINE FIELD

     DEFINE FIELD

        NAME = OV_OPBAL
        DATATYPE = RINT
        LENGTH = 9
        DISPLAY_LENGTH = 9
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 11
        COLUMN = 15
        HELP = Enter value for : OV_OPBAL
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

        DEFINE TRIGGER

           NAME = ON-VALIDATE-FIELD
```

```
        TRIGGER_TYPE = V3
        SHOW_KEY = OFF
        TEXT = <<<
        Declare
          prevbal    number(10);
        Begin
          select dispense.clos_bal
            into prevbal
            from lmis.dispense
           where dispense.fac_code = :datablk.fac_code and
                 dispense.yymm = :GLOBAL.prevdate and
                 dispense.prod_code = 3;
          if prevbal <> :datablk.ov_opbal
            then
              if :GLOBAL.verify = 'NO'
                then
                  :GLOBAL.verify := 'YES';
                  message
       ('Opening Balance not equal to previous Closing Balance -- CHECK VA
                  bell;
                  raise FORM_TRIGGER_FAILURE;
              end if;
          end if;
          if :GLOBAL.verify <> 'OFF'
            then :GLOBAL.verify := 'NO';
          end if;
       Exception
          when NO_DATA_FOUND then
            :GLOBAL.no_bal := '1';
       End;
       >>>

    ENDDEFINE TRIGGER

  ENDDEFINE FIELD

  DEFINE FIELD

    NAME = OV_RECPT
    DATATYPE = RINT
    LENGTH = 9
    DISPLAY_LENGTH = 9
    QUERY_LENGTH = 12
    BASE_TABLE = OFF
    PAGE = 1
    LINE = 11
    COLUMN = 26
    HELP = Enter value for : OV_RECPT
    INPUT_MASK = 9,999,999
    OUTPUT_MASK = 9,999,999
    DEFAULT = 0

  ENDDEFINE FIELD

  DEFINE FIELD
```

```
        NAME = OV_FIELD
        DATATYPE = RINT
        LENGTH = 9
        DISPLAY_LENGTH = 9
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 11
        COLUMN = 37
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0
        AUTOHELP = ON

        DEFINE TRIGGER

           NAME = KEY-NXTFLD
           TRIGGER_TYPE = V3
           SHOW_KEY = OFF
           TEXT = <<<
           if :GLOBAL.status = 'NEW'
             then :datablk.fov_retrn := :datablk.ov_field;
           end if;
           next_field;
           >>>

        ENDDEFINE TRIGGER

     ENDDEFINE FIELD

     DEFINE FIELD

        NAME = OV_ISSUE
        DATATYPE = RINT
        LENGTH = 9
        DISPLAY_LENGTH = 9
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 11
        COLUMN = 48
        HELP = Enter value for : OV_ISSUE
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

        DEFINE TRIGGER

           NAME = KEY-NXTFLD
           TRIGGER_TYPE = V3
           SHOW_KEY = OFF
           TEXT = <<<
           if :GLOBAL.status = 'NEW'
             then :datablk.fov_recpt := :datablk.ov_issue;
           end if;
           next_field;
```

```
        >>>

      ENDDEFINE TRIGGER

    ENDDEFINE FIELD

    DEFINE FIELD

       NAME = OV_RETRN
       DATATYPE = RINT
       LENGTH = 9
       DISPLAY_LENGTH = 9
       QUERY_LENGTH = 12
       BASE_TABLE = OFF
       PAGE = 1
       LINE = 11
       COLUMN = 59
       HELP = Enter value for : OV_RETRN
       INPUT_MASK = 9,999,999
       OUTPUT_MASK = 9,999,999
       DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

       NAME = OV_CLBAL
       DATATYPE = RINT
       LENGTH = 9
       DISPLAY_LENGTH = 9
       QUERY_LENGTH = 12
       BASE_TABLE = OFF
       PAGE = 1
       LINE = 11
       COLUMN = 70
       HELP = Enter value for : OV_CLBAL
       INPUT_MASK = 9,999,999
       OUTPUT_MASK = 9,999,999
       DEFAULT = 0

       DEFINE TRIGGER

          NAME = ON-VALIDATE-FIELD
          TRIGGER_TYPE = V3
          SHOW_KEY = OFF
          TEXT = <<<
          Begin
            if ((:datablk.ov_opbal + :datablk.ov_recpt + :datablk.ov_field) -
                (:datablk.ov_retrn + :datablk.ov_issue)) <> :datablk.ov_clbal
              then
                if :GLOBAL.verify = 'NO'
                  then
                    :GLOBAL.verify := 'YES';
                    message('Closing Balance does not add up -- CHECK VALUE!')
                    bell;
                    raise FORM_TRIGGER_FAILURE;
```

```
                end if;
            end if;
            if :GLOBAL.verify <> 'OFF'
              then :GLOBAL.verify := 'NO';
            end if;
          End;
          >>>

      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = CT_OPBAL
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 12
      COLUMN = 15
      HELP = Enter value for : CT_OPBAL
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = ON-VALIDATE-FIELD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         Declare
           prevbal    number(10);
         Begin
           select dispense.clos_bal
             into prevbal
             from lmis.dispense
            where dispense.fac_code = :datablk.fac_code and
                  dispense.yymm = :GLOBAL.prevdate and
                  dispense.prod_code = 4;
           if prevbal <> :datablk.ct_opbal
             then
               if :GLOBAL.verify = 'NO'
                 then
                   :GLOBAL.verify := 'YES';
                   message
         ('Opening Balance not equal to previous Closing Balance -- CHECK VA
                   bell;
                   raise FORM_TRIGGER_FAILURE;
               end if;
           end if;
           if :GLOBAL.verify <> 'OFF'
             then :GLOBAL.verify := 'NO';
```

```
          end if;
        Exception
          when NO_DATA_FOUND then
            :GLOBAL.no_bal := '1';
        End;
        >>>

      ENDDEFINE TRIGGER

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = CT_RECPT
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 12
      COLUMN = 26
      HELP = Enter value for : CT_RECPT
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = CT_FIELD
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 12
      COLUMN = 37
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0
      AUTOHELP = ON

      DEFINE TRIGGER

        NAME = KEY-NXTFLD
        TRIGGER_TYPE = V3
        SHOW_KEY = OFF
        TEXT = <<<
        if :GLOBAL.status = 'NEW'
          then :datablk.fct_retrn := :datablk.ct_field;
        end if;
        next_field;
        >>>
```

```
        ENDDEFINE TRIGGER

     ENDDEFINE FIELD

     DEFINE FIELD

        NAME = CT_ISSUE
        DATATYPE = RINT
        LENGTH = 9
        DISPLAY_LENGTH = 9
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        MANDATORY = ON
        PAGE = 1
        LINE = 12
        COLUMN = 48
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0
        QUERY = OFF
        AUTOHELP = ON

        DEFINE TRIGGER

           NAME = KEY-NXTFLD
           TRIGGER_TYPE = V3
           SHOW_KEY = OFF
           TEXT = <<<
           if :GLOBAL.status = 'NEW'
             then :datablk.fct_recpt := :datablk.ct_issue;
           end if;
           next_field;
           >>>

        ENDDEFINE TRIGGER

     ENDDEFINE FIELD

     DEFINE FIELD

        NAME = CT_RETRN
        DATATYPE = RINT
        LENGTH = 9
        DISPLAY_LENGTH = 9
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 12
        COLUMN = 59
        HELP = Enter value for : CT_RETRN
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

     ENDDEFINE FIELD
```

```
DEFINE FIELD

   NAME = CT_CLBAL
   DATATYPE = RINT
   LENGTH = 9
   DISPLAY_LENGTH = 9
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 12
   COLUMN = 70
   HELP = Enter value for : CT_CLBAL
   INPUT_MASK = 9,999,999
   OUTPUT_MASK = 9,999,999
   DEFAULT = 0

   DEFINE TRIGGER

      NAME = ON-VALIDATE-FIELD
      TRIGGER_TYPE = V3
      SHOW_KEY = OFF
      TEXT = <<<
      Begin
        if ((:datablk.ct_opbal + :datablk.ct_recpt + :datablk.ct_field) -
            (:datablk.ct_retrn + :datablk.ct_issue)) <> :datablk.ct_clbal
          then
            if :GLOBAL.verify = 'NO'
              then
                :GLOBAL.verify := 'YES';
                message('Closing Balance does not add up -- CHECK VALUE!')
                bell;
                raise FORM_TRIGGER_FAILURE;
            end if;
        end if;
        if :GLOBAL.verify <> 'OFF'
          then :GLOBAL.verify := 'NO';
        end if;
      End;
      >>>

   ENDDEFINE TRIGGER

ENDDEFINE FIELD

DEFINE FIELD

   NAME = DP_OPBAL
   DATATYPE = RINT
   LENGTH = 9
   DISPLAY_LENGTH = 9
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 13
   COLUMN = 15
```

```
        HELP = Enter value for : DP_OPBAL
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

        DEFINE TRIGGER

           NAME = ON-VALIDATE-FIELD
           TRIGGER_TYPE = V3
           SHOW_KEY = OFF
           TEXT = <<<
           Declare
             prevbal    number(10);
           Begin
             select dispense.clos_bal
               into prevbal
               from lmis.dispense
              where dispense.fac_code = :datablk.fac_code and
                    dispense.yymm = :GLOBAL.prevdate and
                    dispense.prod_code = 5;
             if prevbal <> :datablk.dp_opbal
               then
                 if :GLOBAL.verify = 'NO'
                   then
                     :GLOBAL.verify := 'YES';
                     message
           ('Opening Balance not equal to previous Closing Balance -- CHECK VA
                     bell;
                     raise FORM_TRIGGER_FAILURE;
                 end if;
             end if;
             if :GLOBAL.verify <> 'OFF'
               then :GLOBAL.verify := 'NO';
             end if;
           Exception
             when NO_DATA_FOUND then
               :GLOBAL.no_bal := '1';
           End;
           >>>

        ENDDEFINE TRIGGER

     ENDDEFINE FIELD

     DEFINE FIELD

        NAME = DP_RECPT
        DATATYPE = RINT
        LENGTH = 9
        DISPLAY_LENGTH = 9
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 13
        COLUMN = 26
        HELP = Enter value for : DP_RECPT
```

```
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = DP_FIELD
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 13
      COLUMN = 37
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0
      AUTOHELP = ON

      DEFINE TRIGGER

         NAME = KEY-NXTFLD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         if :GLOBAL.status = 'NEW'
           then :datablk.fdp_retrn := :datablk.dp_field;
         end if;
         next_field;
         >>>

      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = DP_ISSUE
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 13
      COLUMN = 48
      HELP = Enter value for : DP_ISSUE
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

      DEFINE TRIGGER
```

```
        NAME = KEY-NXTFLD
        TRIGGER_TYPE = V3
        SHOW_KEY = OFF
        TEXT = <<<
        if :GLOBAL.status = 'NEW'
          then :datablk.fdp_recpt := :datablk.dp_issue;
        end if;
        next_field;
        >>>

    ENDDEFINE TRIGGER

  ENDDEFINE FIELD

  DEFINE FIELD

    NAME = DP_RETRN
    DATATYPE = RINT
    LENGTH = 9
    DISPLAY_LENGTH = 9
    QUERY_LENGTH = 12
    BASE_TABLE = OFF
    PAGE = 1
    LINE = 13
    COLUMN = 59
    HELP = Enter value for : DP_RETRN
    INPUT_MASK = 9,999,999
    OUTPUT_MASK = 9,999,999
    DEFAULT = 0

  ENDDEFINE FIELD

  DEFINE FIELD

    NAME = DP_CLBAL
    DATATYPE = RINT
    LENGTH = 9
    DISPLAY_LENGTH = 9
    QUERY_LENGTH = 12
    BASE_TABLE = OFF
    PAGE = 1
    LINE = 13
    COLUMN = 70
    HELP = Enter value for : DP_CLBAL
    INPUT_MASK = 9,999,999
    OUTPUT_MASK = 9,999,999
    DEFAULT = 0

    DEFINE TRIGGER

      NAME = ON-VALIDATE-FIELD
      TRIGGER_TYPE = V3
      SHOW_KEY = OFF
      TEXT = <<<
      Begin
        if ((:datablk.dp_opbal + :datablk.dp_recpt + :datablk.dp_field) -
```

```
            (:datablk.dp_retrn + :datablk.dp_issue)) <> :datablk.dp_clbal
          then
            if :GLOBAL.verify = 'NO'
              then
                :GLOBAL.verify := 'YES';
                message('Closing Balance does not add up -- CHECK VALUE!'
                bell;
                raise FORM_TRIGGER_FAILURE;
            end if;
        end if;
        if :GLOBAL.verify <> 'OFF'
          then :GLOBAL.verify := 'NO';
        end if;
      End;
      >>>

    ENDDEFINE TRIGGER

  ENDDEFINE FIELD

  DEFINE FIELD

    NAME = NS_OPBAL
    DATATYPE = RINT
    LENGTH = 9
    DISPLAY_LENGTH = 9
    QUERY_LENGTH = 12
    BASE_TABLE = OFF
    PAGE = 1
    LINE = 14
    COLUMN = 15
    HELP = Enter value for : NS_OPBAL
    INPUT_MASK = 9,999,999
    OUTPUT_MASK = 9,999,999
    DEFAULT = 0

    DEFINE TRIGGER

      NAME = ON-VALIDATE-FIELD
      TRIGGER_TYPE = V3
      SHOW_KEY = OFF
      TEXT = <<<
      Declare
        prevbal    number(10);
      Begin
        select dispense.clos_bal
          into prevbal
          from lmis.dispense
         where dispense.fac_code = :datablk.fac_code and
               dispense.yymm = :GLOBAL.prevdate and
               dispense.prod_code = 6;
        if prevbal <> :datablk.ns_opbal
          then
            if :GLOBAL.verify = 'NO'
              then
                :GLOBAL.verify := 'YES';
```

```
                    message
          ('Opening Balance not equal to previous Closing Balance -- CHECK V
                    bell;
                    raise FORM_TRIGGER_FAILURE;
                end if;
            end if;
            if :GLOBAL.verify <> 'OFF'
              then :GLOBAL.verify := 'NO';
            end if;
          Exception
            when NO_DATA_FOUND then
              :GLOBAL.no_bal := '1';
          End;
          >>>

      ENDDEFINE TRIGGER

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = NS_RECPT
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 14
      COLUMN = 26
      HELP = Enter value for : NS_RECPT
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = NS_FIELD
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 14
      COLUMN = 37
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0
      AUTOHELP = ON

      DEFINE TRIGGER

        NAME = KEY-NXTFLD
```

```
        TRIGGER_TYPE = V3
        SHOW_KEY = OFF
        TEXT = <<<
        if :GLOBAL.status = 'NEW'
          then :datablk.fns_retrn := :datablk.ns_field;
        end if;
        next_field;
        >>>

    ENDDEFINE TRIGGER

  ENDDEFINE FIELD

  DEFINE FIELD

    NAME = NS_ISSUE
    DATATYPE = RINT
    LENGTH = 9
    DISPLAY_LENGTH = 9
    QUERY_LENGTH = 12
    BASE_TABLE = OFF
    PAGE = 1
    LINE = 14
    COLUMN = 48
    HELP = Enter value for : NS_ISSUE
    INPUT_MASK = 9,999,999
    OUTPUT_MASK = 9,999,999
    DEFAULT = 0

    DEFINE TRIGGER

      NAME = KEY-NXTFLD
      TRIGGER_TYPE = V3
      SHOW_KEY = OFF
      TEXT = <<<
      if :GLOBAL.status = 'NEW'
        then :datablk.fns_recpt := :datablk.ns_issue;
      end if;
      next_field;
      >>>

    ENDDEFINE TRIGGER

  ENDDEFINE FIELD

  DEFINE FIELD

    NAME = NS_RETRN
    DATATYPE = RINT
    LENGTH = 9
    DISPLAY_LENGTH = 9
    QUERY_LENGTH = 12
    BASE_TABLE = OFF
    PAGE = 1
    LINE = 14
    COLUMN = 59
```

```
        HELP = Enter value for : NS_RETRN
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

     ENDDEFINE FIELD

     DEFINE FIELD

        NAME = NS_CLBAL
        DATATYPE = RINT
        LENGTH = 9
        DISPLAY_LENGTH = 9
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 14
        COLUMN = 70
        HELP = Enter value for : NS_CLBAL
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

        DEFINE TRIGGER

           NAME = ON-VALIDATE-FIELD
           TRIGGER_TYPE = V3
           SHOW_KEY = OFF
           TEXT = <<<
           Begin
             if ((:datablk.ns_opbal + :datablk.ns_recpt + :datablk.ns_field) -
                 (:datablk.ns_retrn + :datablk.ns_issue)) <> :datablk.ns_clbal
               then
                 if :GLOBAL.verify = 'NO'
                   then
                     :GLOBAL.verify := 'YES';
                     message('Closing Balance does not add up -- CHECK VALUE!'
                     bell;
                     raise FORM_TRIGGER_FAILURE;
                 end if;
             end if;
             if :GLOBAL.verify <> 'OFF'
               then :GLOBAL.verify := 'NO';
             end if;
           End;
           >>>

        ENDDEFINE TRIGGER

     ENDDEFINE FIELD

     DEFINE FIELD

        NAME = FCD_OPBAL
        DATATYPE = RINT
        LENGTH = 9
```

```
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 16
      COLUMN = 15
      HELP = Enter value for : FCD_OPBAL
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = ON-VALIDATE-FIELD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         Declare
           prevbal    number(10);
         Begin
           select dispense.clos_bal
             into prevbal
             from lmis.dispense
            where dispense.fac_code = :GLOBAL.fld_code and
                  dispense.yymm = :GLOBAL.prevdate and
                  dispense.prod_code = 1;
           if prevbal <> :datablk.fcd_opbal
             then
               if :GLOBAL.verify = 'NO'
                 then
                   :GLOBAL.verify := 'YES';
                   message
         ('Opening Balance not equal to previous Closing Balance -- CHECK VA
                   bell;
                   raise FORM_TRIGGER_FAILURE;
               end if;
           end if;
           if :GLOBAL.verify <> 'OFF'
             then :GLOBAL.verify := 'NO';
           end if;
         Exception
           when NO_DATA_FOUND then
             :GLOBAL.no_bal := '1';
         End;
         >>>

      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = FCD_RECPT
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
```

```
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 16
        COLUMN = 26
        HELP = Enter value for : FCD_RECPT
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

        NAME = FCD_ISSUE
        DATATYPE = RINT
        LENGTH = 9
        DISPLAY_LENGTH = 9
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 16
        COLUMN = 48
        HELP = Enter value for : FCD_ISSUE
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

        NAME = FCD_RETRN
        DATATYPE = RINT
        LENGTH = 9
        DISPLAY_LENGTH = 9
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 16
        COLUMN = 59
        HELP = Enter value for : FCD_RETRN
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

        NAME = FCD_CLBAL
        DATATYPE = RINT
        LENGTH = 9
        DISPLAY_LENGTH = 9
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
```

```
      PAGE = 1
      LINE = 16
      COLUMN = 70
      HELP = Enter value for : FCD_CLBAL
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = ON-VALIDATE-FIELD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         Begin
           if ((:datablk.fcd_opbal + :datablk.fcd_recpt) -
               (:datablk.fcd_retrn + :datablk.fcd_issue))
              <> :datablk.fcd_clbal
             then
               if :GLOBAL.verify = 'NO'
                 then
                   :GLOBAL.verify := 'YES';
                   message('Closing Balance does not add up -- CHECK VALUE!')
                   bell;
                   raise FORM_TRIGGER_FAILURE;
               end if;
           end if;
           if :GLOBAL.verify <> 'OFF'
             then :GLOBAL.verify := 'NO';
           end if;
         End;
         >>>

      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = FC5_OPBAL
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 17
      COLUMN = 15
      HELP = Enter value for : FC5_OPBAL
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = ON-VALIDATE-FIELD
```

```
        TRIGGER_TYPE = V3
        SHOW_KEY = OFF
        TEXT = <<<
        Declare
          prevbal    number(10);
        Begin
          select dispense.clos_bal
            into prevbal
            from lmis.dispense
           where dispense.fac_code = :GLOBAL.fld_code and
                 dispense.yymm = :GLOBAL.prevdate and
                 dispense.prod_code = 2;
          if prevbal <> :datablk.fc5_opbal
            then
              if :GLOBAL.verify = 'NO'
                then
                  :GLOBAL.verify := 'YES';
                  message
        ('Opening Balance not equal to previous Closing Balance -- CHECK VA
                  bell;
                  raise FORM_TRIGGER_FAILURE;
              end if;
          end if;
          if :GLOBAL.verify <> 'OFF'
            then :GLOBAL.verify := 'NO';
          end if;
        Exception
          when NO_DATA_FOUND then
            :GLOBAL.no_bal := '1';
        End;
        >>>

      ENDDEFINE TRIGGER

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = FC5_RECPT
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 17
      COLUMN = 26
      HELP = Enter value for : FC5_RECPT
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD
```

```
      NAME = FC5_ISSUE
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 17
      COLUMN = 48
      HELP = Enter value for : FC5_ISSUE
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = FC5_RETRN
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 17
      COLUMN = 59
      HELP = Enter value for : FC5_RETRN
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = FC5_CLBAL
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 17
      COLUMN = 70
      HELP = Enter value for : FC5_CLBAL
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = ON-VALIDATE-FIELD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
```

```
        Begin
          if ((:datablk.fc5_opbal + :datablk.fc5_recpt) -
              (:datablk.fc5_retrn + :datablk.fc5_issue))
            <> :datablk.fc5_clbal
            then
              if :GLOBAL.verify = 'NO'
                then
                  :GLOBAL.verify := 'YES';
                  message('Closing Balance does not add up -- CHECK VALUE!'
                  bell;
                  raise FORM_TRIGGER_FAILURE;
              end if;
          end if;
          if :GLOBAL.verify <> 'OFF'
            then :GLOBAL.verify := 'NO';
          end if;
        End;
        >>>

    ENDDEFINE TRIGGER

  ENDDEFINE FIELD

  DEFINE FIELD

    NAME = FOV_OPBAL
    DATATYPE = RINT
    LENGTH = 9
    DISPLAY_LENGTH = 9
    QUERY_LENGTH = 12
    BASE_TABLE = OFF
    PAGE = 1
    LINE = 18
    COLUMN = 15
    HELP = Enter value for : FOV_OPBAL
    INPUT_MASK = 9,999,999
    OUTPUT_MASK = 9,999,999
    DEFAULT = 0

    DEFINE TRIGGER

      NAME = ON-VALIDATE-FIELD
      TRIGGER_TYPE = V3
      SHOW_KEY = OFF
      TEXT = <<<
      Declare
        prevbal    number(10);
      Begin
        select dispense.clos_bal
          into prevbal
          from lmis.dispense
         where dispense.fac_code = :GLOBAL.fld_code and
               dispense.yymm = :GLOBAL.prevdate and
               dispense.prod_code = 3;
        if prevbal <> :datablk.fov_opbal
          then
```

```
            if :GLOBAL.verify = 'NO'
              then
                :GLOBAL.verify := 'YES';
                message
      ('Opening Balance not equal to previous Closing Balance -- CHECK VA
                bell;
                raise FORM_TRIGGER_FAILURE;
            end if;
        end if;
        if :GLOBAL.verify <> 'OFF'
          then :GLOBAL.verify := 'NO';
        end if;
      Exception
        when NO_DATA_FOUND then
          :GLOBAL.no_bal := '1';
      End;
      >>>

    ENDDEFINE TRIGGER

  ENDDEFINE FIELD

  DEFINE FIELD

    NAME = FOV_RECPT
    DATATYPE = RINT
    LENGTH = 9
    DISPLAY_LENGTH = 9
    QUERY_LENGTH = 12
    BASE_TABLE = OFF
    PAGE = 1
    LINE = 18
    COLUMN = 26
    HELP = Enter value for : FOV_RECPT
    INPUT_MASK = 9,999,999
    OUTPUT_MASK = 9,999,999
    DEFAULT = 0

  ENDDEFINE FIELD

  DEFINE FIELD

    NAME = FOV_ISSUE
    DATATYPE = RINT
    LENGTH = 9
    DISPLAY_LENGTH = 9
    QUERY_LENGTH = 12
    BASE_TABLE = OFF
    PAGE = 1
    LINE = 18
    COLUMN = 48
    HELP = Enter value for : FOV_ISSUE
    INPUT_MASK = 9,999,999
    OUTPUT_MASK = 9,999,999
    DEFAULT = 0
```

```
ENDDEFINE FIELD

DEFINE FIELD

   NAME = FOV_RETRN
   DATATYPE = RINT
   LENGTH = 9
   DISPLAY_LENGTH = 9
   QUERY_LENGTH = 12
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 18
   COLUMN = 59
   HELP = Enter value for : FOV_RETRN
   INPUT_MASK = 9,999,999
   OUTPUT_MASK = 9,999,999
   DEFAULT = 0

ENDDEFINE FIELD

DEFINE FIELD

   NAME = FOV_CLBAL
   DATATYPE = RINT
   LENGTH = 9
   DISPLAY_LENGTH =
   QUERY_LENGTH =
   BASE_TABLE = OFF
   PAGE = 1
   LINE = 18
   COLUMN = 70
   HELP = Enter value for : FOV_CLBAL
   INPUT_MASK = 9,999,999
   OUTPUT_MASK = 9,999,999
   DEFAULT = 0

   DEFINE TRIGGER

      NAME = ON-VALIDATE-FIELD
      TRIGGER_TYPE = V3
      SHOW_KEY = OFF
      TEXT = <<<
      Begin
        if ((:datablk.fov_opbal + :datablk.fov_recpt) -
            (:datablk.fov_retrn + :datablk.fov_issue))
          <> :datablk.fov_clbal
          then
            if :GLOBAL.verify = 'NO'
              then
                :GLOBAL.verify := 'YES';
                message('Closing Balance does not add up -- CHECK VALUE!'
                bell;
                raise FORM_TRIGGER_FAILURE;
            end if;
        end if;
        if :GLOBAL.verify <> 'OFF'
```

```
            then :GLOBAL.verify := 'NO';
          end if;
        End;
        >>>

      ENDDEFINE TRIGGER

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = FCT_OPBAL
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 19
      COLUMN = 15
      HELP = Enter value for : FCT_OPBAL
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

      DEFINE TRIGGER

        NAME = ON-VALIDATE-FIELD
        TRIGGER_TYPE = V3
        SHOW_KEY = OFF
        TEXT = <<<
        Declare
          prevbal    number(10);
        Begin
          select dispense.clos_bal
            into prevbal
            from lmis.dispense
           where dispense.fac_code = :GLOBAL.fld_code and
                 dispense.yymm = :GLOBAL.prevdate and
                 dispense.prod_code = 4;
          if prevbal <> :datablk.fct_opbal
            then
              if :GLOBAL.verify = 'NO'
                then
                  :GLOBAL.verify := 'YES';
                  message
        ('Opening Balance not equal to previous Closing Balance -- CHECK VA
                  bell;
                  raise FORM_TRIGGER_FAILURE;
              end if;
          end if;
          if :GLOBAL.verify <> 'OFF'
            then :GLOBAL.verify := 'NO';
          end if;
        Exception
          when NO_DATA_FOUND then
```

116

```
            :GLOBAL.no_bal := '1';
        End;
        >>>

      ENDDEFINE TRIGGER

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = FCT_RECPT
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 19
      COLUMN = 26
      HELP = Enter value for : FCT_RECPT
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = FCT_ISSUE
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 19
      COLUMN = 48
      HELP = Enter value for : FCT_ISSUE
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

      NAME = FCT_RETRN
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 19
      COLUMN = 59
      HELP = Enter value for : FCT_RETRN
```

```
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = FCT_CLBAL
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 19
      COLUMN = 70
      HELP = Enter value for : FCT_CLBAL
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

      DEFINE TRIGGER

         NAME = ON-VALIDATE-FIELD
         TRIGGER_TYPE = V3
         SHOW_KEY = OFF
         TEXT = <<<
         Begin
           if ((:datablk.fct_opbal + :datablk.fct_recpt) -
               (:datablk.fct_retrn + :datablk.fct_issue))
             <> :datablk.fct_clbal
             then
               if :GLOBAL.verify = 'NO'
                 then
                   :GLOBAL.verify := 'YES';
                   message('Closing Balance does not add up -- CHECK VALUE!'
                   bell;
                   raise FORM_TRIGGER_FAILURE;
               end if;
           end if;
           if :GLOBAL.verify <> 'OFF'
             then :GLOBAL.verify := 'NO';
           end if;
         End;
         >>>

      ENDDEFINE TRIGGER

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = FDP_OPBAL
      DATATYPE = RINT
      LENGTH = 9
```

```
        DISPLAY_LENGTH = 9
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 20
        COLUMN = 15
        HELP = Enter value for : FDP_OPBAL
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

        DEFINE TRIGGER

           NAME = ON-VALIDATE-FIELD
           TRIGGER_TYPE = V3
           SHOW_KEY = OFF
           TEXT = <<<
           Declare
             prevbal    number(10);
           Begin
             select dispense.clos_bal
               into prevbal
               from lmis.dispense
              where dispense.fac_code = :GLOBAL.fld_code and
                    dispense.yymm = :GLOBAL.prevdate and
                    dispense.prod_code = 5;
             if prevbal <> :datablk.fdp_opbal
               then
                 if :GLOBAL.verify = 'NO'
                   then
                     :GLOBAL.verify := 'YES';
                     message
           ('Opening Balance not equal to previous Closing Balance -- CHECK VA
                     bell;
                     raise FORM_TRIGGER_FAILURE;
                 end if;
             end if;
             if :GLOBAL.verify <> 'OFF'
               then :GLOBAL.verify := 'NO';
             end if;
           Exception
             when NO_DATA_FOUND then
               :GLOBAL.no_bal := '1';
           End;
           >>>

        ENDDEFINE TRIGGER

     ENDDEFINE FIELD

     DEFINE FIELD

        NAME = FDP_RECPT
        DATATYPE = RINT
        LENGTH = 9
        DISPLAY_LENGTH = 9
```

119

```
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 20
      COLUMN = 26
      HELP = Enter value for : FDP_RECPT
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = FDP_ISSUE
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 20
      COLUMN = 48
      HELP = Enter value for : FDP_ISSUE
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = FDP_RETRN
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
      PAGE = 1
      LINE = 20
      COLUMN = 59
      HELP = Enter value for : FDP_RETRN
      INPUT_MASK = 9,999,999
      OUTPUT_MASK = 9,999,999
      DEFAULT = 0

   ENDDEFINE FIELD

   DEFINE FIELD

      NAME = FDP_CLBAL
      DATATYPE = RINT
      LENGTH = 9
      DISPLAY_LENGTH = 9
      QUERY_LENGTH = 12
      BASE_TABLE = OFF
```

```
        PAGE = 1
        LINE = 20
        COLUMN = 70
        HELP = Enter value for : FDP_CLBAL
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

        DEFINE TRIGGER

           NAME = ON-VALIDATE-FIELD
           TRIGGER_TYPE = V3
           SHOW_KEY = OFF
           TEXT = <<<
           Begin
             if ((:datablk.fdp_opbal + :datablk.fdp_recpt) -
                 (:datablk.fdp_retrn + :datablk.fdp_issue))
               <> :datablk.fdp_clbal
               then
                 if :GLOBAL.verify = 'NO'
                   then
                     :GLOBAL.verify := 'YES';
                     message('Closing Balance does not add up -- CHECK VALUE!'
                     bell;
                     raise FORM_TRIGGER_FAILURE;
                 end if;
             end if;
             if :GLOBAL.verify <> 'OFF'
               then :GLOBAL.verify := 'NO';
             end if;
           End;
           >>>

        ENDDEFINE TRIGGER

     ENDDEFINE FIELD

     DEFINE FIELD

        NAME = FNS_OPBAL
        DATATYPE = RINT
        LENGTH = 9
        DISPLAY_LENGTH = 9
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 21
        COLUMN = 15
        HELP = Enter value for : FNS_OPBAL
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

        DEFINE TRIGGER

           NAME = ON-VALIDATE-FIELD
```

```
        TRIGGER_TYPE = V3
        SHOW_KEY = OFF
        TEXT = <<<
        Declare
          prevbal    number(10);
        Begin
          select dispense.clos_bal
            into prevbal
            from lmis.dispense
           where dispense.fac_code = :GLOBAL.fld_code and
                 dispense.yymm = :GLOBAL.prevdate and
                 dispense.prod_code = 6;
          if prevbal <> :datablk.fns_opbal
            then
              if :GLOBAL.verify = 'NO'
                then
                  :GLOBAL.verify := 'YES';
                  message
        ('Opening Balance not equal to previous Closing Balance -- CHECK VA
                  bell;
                  raise FORM_TRIGGER_FAILURE;
              end if;
          end if;
          if :GLOBAL.verify <> 'OFF'
            then :GLOBAL.verify := 'NO';
          end if;
        Exception
          when NO_DATA_FOUND then
            :GLOBAL.no_bal := '1';
        End;
        >>>

    ENDDEFINE TRIGGER

ENDDEFINE FIELD

DEFINE FIELD

    NAME = FNS_RECPT
    DATATYPE = RINT
    LENGTH = 9
    DISPLAY_LENGTH = 9
    QUERY_LENGTH = 12
    BASE_TABLE = OFF
    PAGE = 1
    LINE = 21
    COLUMN = 26
    HELP = Enter value for : FNS_RECPT
    INPUT_MASK = 9,999,999
    OUTPUT_MASK = 9,999,999
    DEFAULT = 0

ENDDEFINE FIELD

DEFINE FIELD
```

```
        NAME = FNS_ISSUE
        DATATYPE = RINT
        LENGTH = 9
        DISPLAY_LENGTH = 9
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 21
        COLUMN = 48
        HELP = Enter value for : FNS_ISSUE
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

        NAME = FNS_RETRN
        DATATYPE = RINT
        LENGTH = 9
        DISPLAY_LENGTH = 9
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 21
        COLUMN = 59
        HELP = Enter value for : FNS_RETRN
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

    ENDDEFINE FIELD

    DEFINE FIELD

        NAME = FNS_CLBAL
        DATATYPE = RINT
        LENGTH = 9
        DISPLAY_LENGTH = 9
        QUERY_LENGTH = 12
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 21
        COLUMN = 70
        HELP = Enter value for : FNS_CLBAL
        INPUT_MASK = 9,999,999
        OUTPUT_MASK = 9,999,999
        DEFAULT = 0

        DEFINE TRIGGER

            NAME = ON-VALIDATE-FIELD
            TRIGGER_TYPE = V3
            SHOW_KEY = OFF
            TEXT = <<<
```

```
        Begin
          if ((:datablk.fns_opbal + :datablk.fns_recpt) -
              (:datablk.fns_retrn + :datablk.fns_issue))
            <> :datablk.fns_clbal
            then
              if :GLOBAL.verify = 'NO'
                then
                  :GLOBAL.verify := 'YES';
                  message('Closing Balance does not add up -- CHECK VALUE!'
                  bell;
                  raise FORM_TRIGGER_FAILURE;
              end if;
          end if;
          if :GLOBAL.verify <> 'OFF'
            then :GLOBAL.verify := 'NO';
          end if;
        End;
        >>>

    ENDDEFINE TRIGGER

    DEFINE TRIGGER

        NAME = KEY-NXTFLD
        TRIGGER_TYPE = V3
        SHOW_KEY = OFF
        TEXT = <<<
        EXECUTE_TRIGGER('KEY-COMMIT');
        >>>

    ENDDEFINE TRIGGER

  ENDDEFINE FIELD

  DEFINE FIELD

    NAME = FAC_NAME
    DATATYPE = CHAR
    LENGTH = 20
    DISPLAY_LENGTH = 20
    QUERY_LENGTH = 20
    BASE_TABLE = OFF
    PAGE = 1
    LINE = 4
    COLUMN = 50
    INPUT = OFF
    UPDATE = OFF
    QUERY = OFF

  ENDDEFINE FIELD

  DEFINE FIELD

    NAME = TYPE_NAME
    DATATYPE = CHAR
    LENGTH = 20
```

```
        DISPLAY_LENGTH = 20
        QUERY_LENGTH = 20
        BASE_TABLE = OFF
        PAGE = 1
        LINE = 5
        COLUMN = 50
        INPUT = OFF
        UPDATE = OFF
        QUERY = OFF

     ENDDEFINE FIELD

  ENDDEFINE BLOCK


  DEFINE SCREEN

     DEFINE PAGE

        PAGE = 1
        PAGE_XS = 0
        PAGE_YS = 0
        PAGE_PX0 = 0
        PAGE_PY0 = 0
        PAGE_PXS = 0
        PAGE_PYS = 0
        PAGE_SX0 = 0
        PAGE_SY0 = 0
        MODE = TEXT
        LINE = 2
        BOILER = <<<
                                    Form 7b Data Entry Screen
        >>>
        LINE = 4
        BOILER = <<<
                  Facility Code:                    Name:
          Year and Month (YYMM):                    Type:
        >>>
        LINE = 7
        BOILER = <<<
                        Opening          **Thana Storeroom**          Trans/
                        Balance   Received   Ret/Field    Issues   Returned
         1. Condoms
         2. OPC-5
         3. Ovral
         4. Copper-T
         5. Depo-Prov
         6. Norist.
                                          **Field Stock**
         1. Condoms
         2. OPC-5
         3. Ovral
         4. Copper-T
         5. Depo-Prov
         6. Norist.
        >>>
```

```
    MODE = BOX
    BOILER = <<<
    p-------------------------------------------------------------------
    |
    (-------------------------------------------------------------------
    |
    |
    (-------------------------------------------------------------------
    |
    |
    |
    |
    |
    |
    |
    |
    |
    |
    |
    |
    |
    b-------------------------------------------------------------------
    >>>

ENDDEFINE PAGE

DEFINE PAGE

    PAGE = 2
    PAGE_XS = 0
    PAGE_YS = 0
    PAGE_PX0 = 0
    PAGE_PY0 = 0
    PAGE_PXS = 0
    PAGE_PYS = 0
    PAGE_SX0 = 0
    PAGE_SY0 = 0

ENDDEFINE PAGE

DEFINE PAGE

    PAGE = 3
    PAGE_XS = 0
    PAGE_YS = 0
    PAGE_PX0 = 0
    PAGE_PY0 = 0
    PAGE_PXS = 0
    PAGE_PYS = 0
    PAGE_SX0 = 0
    PAGE_SY0 = 0

ENDDEFINE PAGE
```

```
      DEFINE PAGE

         PAGE = 4
         PAGE_XS = 0
         PAGE_YS = 0
         PAGE_PX0 = 0
         PAGE_PY0 = 0
         PAGE_PXS = 0
         PAGE_PYS = 0
         PAGE_SX0 = 0
         PAGE_SY0 = 0

      ENDDEFINE PAGE

   ENDDEFINE SCREEN

ENDDEFINE FORM
```