Module Pool Programming with

System Application Product (SAP)

A Major Project Submitted in Partial Fulfilment of the requirements

for award of the degree of

MASTER OF ENGINEERING IN ELECTRONICS AND COMMUNICATION ENGINEERING

DELHI UNIVERSITY, DELHI

Submitted By

Sunil Khinchi (11 / E&C / 03)

Delhi University Roll No. 2851

Under The Guidance of

Mr. O.P. Verma

Assistant Professor

Department of Electronics & Communication Engineering

Delhi College of Engineering, Delhi.

Department of Electronics & Communication Engineering

Delhi College of Engineering,

Delhi - 110042.

June – 2007

CERTIFICATE

This is citify that this thesis work entitled **"MODULE POOL PROGRAMING WITH SYSTEM APPLIATION PRODUCT (SAP)"** is bonafide work carried out by **SUNIL KHINCHI (11 / E&C / 03 DU Roll NO. 2851)** who carried out the Major Project work under my supervision and submitted in partial fulfilment of the requirements for the award of the Degree Master of Engineering.

Mr. O. P. Verma,

Assistant Professor,

Department of Electronics & Communication,

Delhi College of Engineering – DELHI.

ACKNOWLEDGEMENTS

Behind every achievement of a student lie the unflinching efforts of his teachers without whom, as student we could never know the liveliness of hard work and to this day we certainly feel it as our worldly pleasure to get living to this day of thanking them.

I would like to thank our Had of Department **Professor Ashok Bhattachary and My Project Guide Mr. O. P. Verma** for their kind co-operation in bringing out this Project.

Sunil Khinchi

11 / E&C / 03

DU Roll No. 2851

ABSTRACT

The Work in this thesis **"MODULE POOL PROGRAMING WITH SYSTEM APPLICATION (SAP)"** present an approach to design and customize system application product (SAP) an Enterprise Resource Planning (ERP) Package as per the requirement of organization for the sales and distribution, material management and finance and control.

This report is prepared in advance Business Application Programming (ABAP) Language and screens are generated with help of Module Pool Programming.

CONTANTS

Chapter 1
INTRODUCTION

SAP was founded in 1972 in Walldorf, Germany. It stands for Systems, Applications and Products in Data Processing. Over the years, it has grown and evolved to become the world premier provider of client/server business solutions for which it is so well known today. The SAP R/3 enterprise application suite for open client/server systems has established a new standards for providing business information management solutions.

SAP product are consider excellent but not perfect. The main problems with software product is that it can never be perfect.

The main advantage of using SAP as your company ERP system is that SAP have a very high level of integration among its individual applications which guarantee consistency of data throughout the system and the company itself.

In a standard SAP project system, it is divided into three environments, **Development, Quality Assurance and Production**.

The development system is where most of the implementation work takes place. The quality assurance system is where all the final testing is conducted before moving the transports to the production environment. The production system is where all the daily business activities occur. It is also the client that all the end users use to perform their daily job functions.

To all company, the production system should only contains transport that have passed all the tests.

SAP is a table drive customization software. It allows businesses to make rapid changes in their business requirements with a common set of programs. User-exits are provided for business to add in additional source code. Tools such as screen variants are provided to let you set fields attributes whether to hide, display and make them mandatory fields.

Chapter 2

Types of Projects

The project types delivered with the SAP Solution Manager allow you to differentiate between different types of projects. You can use the SAP Solution Manager to create the following projects:

- **Implementation Project**

  Project to implement business processes in an SAP landscape.

  Create a project structure from the business processes. You can either create a new project structure, or base it on one of the following:

  - One or more user or partner templates
  - An existing project
  - Scenarios and configuration structures delivered by SAP
  - An existing production solution landscape

- **Template Project**

  A project to **create a template**.

  A template makes your project structure, or parts of it, with its assigned objects (documentation, test cases, IMG activities), available to other projects.

  You can lock templates, completely or partially, against changes when they are used in other projects. To use templates in other systems, transport them.

  Template projects are especially suited to SAP partner solutions or global rollout.

- **Upgrade Project**

  A project to upgrade existing systems.

  In an upgrade project you can:

  - Upgrade customizing: Upgrade existing functions
  - and/or
  - Delta customizing: Copy additional functions

- Optimization project

  A project to optimize the flow of business processes, or the use of a software solution.

  You can use optimization projects, for example, in SAP Services.

- **Safeguarding project**

  A project to resolve a critical situation in the implementation or use of an SAP solution.

  Safeguarding projects show the reasons for a critical situation and coordinate the steps required to resolve the problems.

- **Maintenance project**

  A project to maintain a solution

Chapter 3

SAP R/3 Architecture

History of SAP R/3

The first version of SAP's flagship enterprise software was a financial Accounting system named R/1. This was replaced by R/2 at the end of the 1970s. SAP R/2 was in a mainframe based business application software suite that was very successful in the 1980s and early 1990s. It was particularly popular with large multinational European companies who required soft-real-time business applications, with multi-currency and multi-language capabilities built in. With the advent of distributed client-server computing SAP AG brought out a client-server version of the software called SAP R/3 (The "R" was for "Real-time data processing" and 3 was for 3-tier). This new architecture is compatible with multiple platforms and operating systems, such as Microsoft Windows or UNIX. This opened up SAP to a whole new customer base.

SAP R/3 was officially launched on 6 July 1992. It was renamed SAP ERP and later again renamed ECC (ERP Central Component). SAP came to dominate the large business applications market over the next 10 years. SAP ECC 5.0 ERP is the successor of SAP R/3 4.70. The newest version of the suite is MySAP 2005 or SAP ECC 6.0.

SAP R/3 Release 4.0B Release Date June 1998

SAP R/3 Release 4.5B Release Date March 1999

SAP R/3 Release 4.6B Release Date Dec 1999

SAP R/3 Release 4.6C Release Date April 2001

SAP R/3 Enterprise Release 4.70 Release Date March- Dec 2003

SAP ECC 5.0 ERP (mySAP ERP 2004) Release Year 2004

SAP ECC 6.0 ERP (mySAP ERP 2005) Release Year 2005[2]

[edit] Organization

SAP R/3 is arranged into distinct functional modules, covering the typical functions in place in an organization. The most widely used modules are Financials and Controlling (FICO), Human Resources (HR), Materials Management (MM), Sales & Distribution (SD), and Production Planning (PP). Those modules, as well as the additional components of SAP R/3, are detailed in the next section.

Each module handles specific business tasks on its own, but is linked to the others where applicable. For instance, an invoice from the Billing transaction of Sales & Distribution will pass through to accounting, where it will appear in accounts receivable and cost of goods sold.

SAP has typically focused on best practice methodologies for driving its software processes, but has more recently expanded into vertical markets. In these situations, SAP produces specialized modules (referred to as IS or Industry Specific) geared toward a particular market segment, such as utilities or retail.

Using SAP often requires the payment of hefty license fees, as the customers have effectively outsourced various business software development tasks to SAP. By specializing in software development, SAP hopes to provide a better value to corporations than they could if they attempted to develop and maintain their own applications.

[edit] Technology

SAP based the architecture of R/3 on a three-tier client/server model.

1. Presentation Server 2. Application Server 3. Database Server

[edit] Presentation Server

The presentation server is actually a program named sapgui.exe. It is usually installed on a user's workstation. To start it, the user double-clicks on an icon on the desktop or chooses a menu path. When started, the presentation server displays the R/3 menus within a window. This window is commonly known as the SAPGUI, or the user interface (or simply, the interface). The interface accepts input from the user in the form of keystrokes, mouse-clicks, and function keys, and sends these requests to the application server to be processed. The application server sends the results back to the SAPGUI which then formats the output for display to the user.

[edit] Application Server

An application server is a set of executables that collectively interpret the ABAP/4 programs and manage the input and output for them. When an application server is started, these executables all start at the same time. When an application server is stopped, they all shut down together. The number of processes that start up when you bring up the application server is defined in a single configuration file called the application server profile. Each application server has a profile that specifies its characteristics when it starts up and while it is running. For example, an application sever profile specifies: • Number of processes and their types • Amount of memory each process may use • Length of time a user is inactive before being automatically logged off. The application server exists to interpret ABAP/4 programs, and they only run there-the programs do not run on the presentation server. An ABAP/4 program can start an executable on the presentation server, but an ABAP/4 program cannot execute there. If your ABAP/4 program requests information from the database, the application server will format the request and send it to the database server.

[edit] Discovering the Database Server

The database server is a set of executables that accept database requests from the application server. These requests are passed on to the RDBMS (Relation Database Management System). The RDBMS sends the data back to the database server, which then passes the

information back to the application server. The application server in turn passes that information to your ABAP/4 program.

There is usually a separate computer dedicated to house the database server, and the RDBMS may run on that computer also, or may be installed on its own computer.

SAP R/3 functionality is structured using its own proprietary language called ABAP (Advanced Business Application Programming). ABAP, or ABAP/4 is a fourth generation language (4GL), geared towards the creation of simple, yet powerful programs. R/3 also offers a complete development environment where developers can either modify existing SAP code to modify existing functionality or develop their own functions, whether reports or complete transactional systems within the SAP framework.

ABAP's main interaction with the database system is via Open SQL statements. These statements allow a developer to query, update, or delete information from the database. Advanced topics include GUI development and advanced integration with other systems. With the introduction of ABAP Objects, ABAP provides the opportunity to develop applications with object-oriented programming.

The most difficult part of SAP R/3 is its implementation, since SAP R/3 is never used the same way in any two places. For instance, Atlas Copco can have a different implementation of SAP R/3 from Procter & Gamble. Some companies may run multiple productive clients/systems or even multiple *instances* of SAP R/3. This is seen, for example, when a company running R/3 acquires a new business already running R/3. They may elect to keep both systems separate, migrate one into the other, or migrate both onto a completely new instance.

The system landscape is ultimately the customer's decision. There are definite pros and cons on the continuum from single global instance / productive client (master data, impact of configuration changes on multiple business units) to separate instances per business unit (hardware costs and communication between instances/clients)

Two primary issues are the root of the complexity and of the differences:

- Customization configuration - Within R/3, there are tens of thousands of database tables that may be used to control how the application behaves. For instance, each company will have its own accounting "Chart of Accounts" which reflects how its transactions flow together to represent its activity. That will be specific to a given company. In general, the behavior (and appearance) of virtually every screen and transaction is controlled by configuration tables. This gives the implementor great power to make the application behave differently for different environments. With that power comes considerable complexity.
- Extensions, Bolt-Ons - In any company, there will be a need to develop interface programs to communicate with other corporate information systems. This generally involves developing ABAP/4 code, and considerable "systems integration" effort to either determine what data is to be drawn out of R/3 or to interface into R/3 to load data into the system.

Due to the complexity of implementation, these companies recruit highly skilled SAP consultants to do the job. The implementation must consider the company's needs and

resources. Some companies implement only a few modules of SAP while others may want numerous modules.

SAP has several layers. The **Basis System** (BC) includes the ABAP programming language, and is the heart (i.e. the base) of operations and should not be visible to higher level or managerial users. Other customizing and implementation tools exist also. The heart of the system (from a manager's viewpoint) are the application modules.

Chapter 4

Link Between SAP Modules

Link Between SAP SD, MM & FI

1. In SAP you will always get integration with other modules. SD will interact with FI, MM will interact with SD :-

1a. Looking at MM and SD interaction first, take the scenario of a third party order process. This process uses a purchase order (which is sent to your vendor). Also invoice verification is used further along the process to check that the invoice you send to your customer is the same material and quantity as that which the vendor sends to you (but obviously shipped directly to your customer).

1b. Billing is an SD function. But SAP needs to know, when processing a customer's payment, to which GL account the payment has to be processed. For instance payment of a UK based material would be placed in a different GL account to that of a non-UK based material. Furthermore, a UK based customer may have a different GL account to that of an Export customer. This is configured in Account Determination.

2. ABAPers are there to essential do some bespoke development. Your integration, or interaction, with them would be when specifying the tables, fields, input fields, a simple process explanation, data mapping (if doing an interface from one system to another) etc. ***-- Shahee**

The link between SD and MM :-

1. When you create sales order in SD, all the details of the items are copied from Material master of MM.

2. MRP and availibility check related data is also taken from MM although you control this data in SD also.

3. While you create inbound/outbound delivery with reference to a sales order,the shipping point determination takes place with the help of the loading group, plant data, shipping conditions etc. This also refers to Material Master.

4. The material which you are entering in a sales order must be extended to the sales area of your sales order/customer otherwise you cannot transact with this material.

There are many such links between SD and MM.

Now the link between SD and FI :-

1. Whenever you create a delivery with reference to a sales order, goods movement takes place in the bacgground. eg. In case of standard sales order, you create an outbound goods delivery to the customer.

Here movement 601 takes place. This movement is configured in MM. Also, this movement hits some G/L account in FI. Every such movement of good s hits some G/L account.

2. The accounts posting in FI is done with reference to the billing documents (invoice, debit note, credit note etc) created in SD. Thus this is a link between SD and FI

3. Tax determination: In case of a tax determination also, there is a direct link between SD and MM

SD Integration points with other modules

SD module is highly integrated with the other modules in SAP.

```
Sales Order –
Integration Points                      Module
•Availability Check             -       MM
•Credit Check                   -       FI
•Costing                        -       CO/ MM
•Tax Determination              -       FI
•Transfer of Requirements       -       PP/ MM

Delivery & Goods Issue –
Integration Points                      Module
•Availability Check             -       MM
•Credit Check                   -       FI
•Reduces stock                  -       MM
•Reduces Inventory $            -       FI/ CO
•Requirement Eliminated         -       PP/ MM

Billing -
Integration Points                      Module
•Debit A/R                      -       FI/ CO
•Credit Revenue                 -       FI/ CO
•Updates G/ L                   -       FI/ CO
  (Tax, discounts, surcharges, etc.)
•Milestone Billing              -       PS

Return Delivery & Credit Memo -
Integration Points                      Module
•Increases Inventory            -       MM
•Updates G/ L                   -       FI
•Credit Memo                    -       FI
•Adjustment to A/R              -       FI
•Reduces Revenue                -       FI
```

Tips by: Subha

SD Transaction Code Flow:

Inquiry / Document type IN
Tcode for creation VA11,VA12,VA13. tables VBAK,VBAP

Quotation / QT
Tcode for creation VA21,VA22,VA23. tables VBAK,VBAP

Purchase Order PO
Tcode for creation ME21,ME22,ME23. tables EKKO,EKPO.

Sales Order OR
Tcode for creation VA01,VA02,VA03. tables VBAK,VBAP

Delivery LF
Tcode for creation VL01,VL02,VL03. tables LIKP,LIPS

Billing MN
Tcode for creation VF01,VF02,VF03. tables VBRK,VBRP

To create a sales order we need purchase order number and custmer number. Before that, to create a purchase order we need to have material no, vendor no.

To create vendor tcode is
xk01(create), xk02(change) , xk03(display)
Tables are lfa1.

To create custmer tcode is xd01, xd02, xd03.
Table is kna1.

Chapter 8
Reports

```
REPORT  ZDEMO_001                       .

TYPE-POOLS SLIS.

DATA I_FIELDCAT TYPE  SLIS_T_FIELDCAT_ALV.
DATA WA_FIELDCAT TYPE SLIS_FIELDCAT_ALV.

DATA IT_VBAK LIKE VBAK OCCURS 0 WITH HEADER LINE.

START-OF-SELECTION.
  SELECT * FROM VBAK INTO TABLE IT_VBAK UP TO 20 ROWS.

END-OF-SELECTION.

*  CALL FUNCTION 'REUSE_ALV_FIELDCATALOG_MERGE'
*   EXPORTING
**   I_PROGRAM_NAME            =
*    I_INTERNAL_TABNAME         = 'IT_VBAK'.
*    I_STRUCTURE_NAME           = 'VBAK'
**   I_CLIENT_NEVER_DISPLAY      = 'X'
**   I_INCLNAME                 =
**   I_BYPASSING_BUFFER         =
**   I_BUFFER_ACTIVE            =
```

```
*    CHANGING
*     CT_FIELDCAT               = I_FIELDCAT
** EXCEPTIONS
**   INCONSISTENT_INTERFACE      = 1
**   PROGRAM_ERROR               = 2
**   OTHERS                      = 3
*          .
*  IF SY-SUBRC <> 0.
** MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
**        WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
*  ENDIF.

 WA_FIELDCAT-FIELDNAME = 'VBELN'.
 WA_FIELDCAT-SELTEXT_L = 'Sal doc no'.
 WA_FIELDCAT-KEY = 'X'.

 APPEND WA_FIELDCAT TO I_FIELDCAT.
 CLEAR WA_FIELDCAT.

 WA_FIELDCAT-FIELDNAME = 'ERDAT'.
 WA_FIELDCAT-SELTEXT_L = 'Date'.

 APPEND WA_FIELDCAT TO I_FIELDCAT.
 CLEAR WA_FIELDCAT.

 WA_FIELDCAT-FIELDNAME = 'ERNAM'.
 WA_FIELDCAT-SELTEXT_L = 'Name'.
```

```
 APPEND WA_FIELDCAT TO I_FIELDCAT.
 CLEAR WA_FIELDCAT.

 CALL FUNCTION 'REUSE_ALV_LIST_DISPLAY'
  EXPORTING
   IT_FIELDCAT                = I_FIELDCAT
*    IT_EXCLUDING               =
*    IT_SPECIAL_GROUPS          =
*    IT_SORT                    =
*    IT_FILTER                  =
*    IS_SEL_HIDE                =
*    I_DEFAULT                  = 'X'
*    I_SAVE                     = ' '
*    IS_VARIANT                 =
*    IT_EVENTS                  =
*    IT_EVENT_EXIT              =
*    IS_PRINT                   =
*    IS_REPREP_ID               =
*    I_SCREEN_START_COLUMN      = 0
*    I_SCREEN_START_LINE        = 0
*    I_SCREEN_END_COLUMN        = 0
*    I_SCREEN_END_LINE          = 0
*  IMPORTING
*    E_EXIT_CAUSED_BY_CALLER    =
*    ES_EXIT_CAUSED_BY_USER     =
   TABLES
```

```
     T_OUTTAB                  = IT_VBAK
*   EXCEPTIONS
*     PROGRAM_ERROR             = 1
*     OTHERS                    = 2
          .
 IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*        WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
 ENDIF.
```

```
REPORT  ZDEMO_002                    .
TYPE-POOLS SLIS.

DATA IT_VBAP LIKE VBAP OCCURS 0 WITH HEADER LINE.

DATA I_FIELDCAT TYPE SLIS_T_FIELDCAT_ALV.
DATA WA_FIELDCAT TYPE SLIS_FIELDCAT_ALV.

START-OF-SELECTION.
 SELECT * FROM VBAP INTO TABLE IT_VBAP UP TO 20 ROWS.

END-OF-SELECTION.
```

```
*--1
WA_FIELDCAT-FIELDNAME = 'VBELN'.
WA_FIELDCAT-SELTEXT_L = 'Sal doc no'.
WA_FIELDCAT-key = 'X'.

APPEND WA_FIELDCAT TO I_FIELDCAT.
CLEAR WA_FIELDCAT.

*-- 2
WA_FIELDCAT-FIELDNAME = 'POSNR'.
WA_FIELDCAT-SELTEXT_L = 'Item no'.

APPEND WA_FIELDCAT TO I_FIELDCAT.
CLEAR WA_FIELDCAT.

WA_FIELDCAT-FIELDNAME = 'MATNR'.
WA_FIELDCAT-SELTEXT_L = 'Material no'.

APPEND WA_FIELDCAT TO I_FIELDCAT.
CLEAR WA_FIELDCAT.

WA_FIELDCAT-FIELDNAME = 'NETWR'.
WA_FIELDCAT-SELTEXT_L = 'Amount'.

APPEND WA_FIELDCAT TO I_FIELDCAT.
CLEAR WA_FIELDCAT.
```

```
*-- GRID
CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
 EXPORTING
*   I_INTERFACE_CHECK              = ' '
*   I_BYPASSING_BUFFER             = ' '
*   I_BUFFER_ACTIVE                = ' '
  I_CALLBACK_PROGRAM              = SY-REPID
*   I_CALLBACK_PF_STATUS_SET       = ' '
*   I_CALLBACK_USER_COMMAND        = ' '
  I_CALLBACK_TOP_OF_PAGE          = 'TOP_OF_PAGE'
*   I_CALLBACK_HTML_TOP_OF_PAGE    = ' '
*   I_CALLBACK_HTML_END_OF_LIST    = ' '
*   I_STRUCTURE_NAME               =
*   I_BACKGROUND_ID                = ' '
*   I_GRID_TITLE                   =
*   I_GRID_SETTINGS                =
*   IS_LAYOUT                      =
  IT_FIELDCAT                     = I_FIELDCAT
*   IT_EXCLUDING                   =
*   IT_SPECIAL_GROUPS              =
*   IT_SORT                        =
*   IT_FILTER                      =
*   IS_SEL_HIDE                    =
*   I_DEFAULT                      = 'X'
*   I_SAVE                         = ' '
*   IS_VARIANT                     =
*   IT_EVENTS                      =
```

```
*   IT_EVENT_EXIT               =
*   IS_PRINT                    =
*   IS_REPREP_ID                =
*   I_SCREEN_START_COLUMN       = 0
*   I_SCREEN_START_LINE         = 0
*   I_SCREEN_END_COLUMN         = 0
*   I_SCREEN_END_LINE           = 0
*   IT_ALV_GRAPHICS             =
*   IT_HYPERLINK                =
*   IT_ADD_FIELDCAT             =
*   IT_EXCEPT_QINFO             =
*   I_HTML_HEIGHT_TOP           =
*   I_HTML_HEIGHT_END           =
* IMPORTING
*   E_EXIT_CAUSED_BY_CALLER     =
*   ES_EXIT_CAUSED_BY_USER      =
 TABLES
   T_OUTTAB                    = IT_VBAP
* EXCEPTIONS
*   PROGRAM_ERROR               = 1
*   OTHERS                      = 2
        .
IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*        WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.
```

```
FORM TOP_OF_PAGE.

DATA I_HEADER TYPE  SLIS_T_LISTHEADER.

DATA WA_HEADER TYPE  SLIS_LISTHEADER.


WA_HEADER-typ = 'H'.

WA_HEADER-info = 'ALV Header'.


APPEND WA_HEADER TO I_HEADER.


*WA_HEADER-typ = 'S'.

*WA_HEADER-info = 'ALV Header'.

*

*APPEND WA_HEADER TO I_HEADER.

*

*WA_HEADER-typ = 'A'.

*WA_HEADER-info = 'ALV Header'.

*

*APPEND WA_HEADER TO I_HEADER.


CALL FUNCTION 'REUSE_ALV_COMMENTARY_WRITE'

 EXPORTING

   IT_LIST_COMMENTARY       = I_HEADER

  I_LOGO                 = 'ENJOYSAP_LOGO'

*   I_END_OF_LIST_GRID       =

        .


ENDFORM.
```

```
REPORT  ZNACLASSICAL.


TABLES: MARA,MARD,MARC,MAKT.



DATA: BEGIN OF SN_ITAB OCCURS 0,
    SN_MATNR LIKE MARD-MATNR,
    SN_WERKS LIKE MARD-WERKS,
    SN_LGORT LIKE MARD-LGORT,
```

```
      SN_ERSDA LIKE MARA-ERSDA,
      SN_ERNAM LIKE MARA-ERNAM,
      SN_MAKTX LIKE MAKT-MAKTX,
      END OF SN_ITAB.
data : begin of itab3 occurs 0,
        werks like t001l-werks,
        lgort like t001l-lgort,
        lgobe like t001l-lgobe,
       end of itab3.
data : begin of itab4 occurs 0,
        werks like t001w-werks,
        name1 like t001w-name1,
       end of itab4.


SELECT-OPTIONS: SN_MATNR FOR MARA-MATNR,
                SN_LGORT FOR MARD-LGORT,
                SN_WERKS FOR MARD-WERKS,
                SN_ERSDA FOR MARA-ERSDA.


SELECT MARA~MATNR
       MARD~WERKS
       MARD~LGORT
       MARA~ERSDA
       MARA~ERNAM
       MAKT~MAKTX into table SN_itab from ( ( MARD inner join MARA on
```

```
MARA~MATNR eq MARD~MATNR ) inner join MAKT on MARA~MATNR eq MAKT~MATNR
) WHERE MARA~MATNR IN SN_MATNR AND MARD~LGORT IN SN_LGORT AND MARD~WERKS
 IN SN_WERKS AND MARA~ERSDA IN SN_ERSDA.




IF NOT SN_ITAB[] IS INITIAL.
SELECT WERKS
     NAME1 FROM T001W INTO TABLE ITAB3 FOR ALL ENTRIES IN SN_ITAB
WHERE WERKS EQ SN_ITAB-SN_WERKS.

SELECT WERKS
     LGOBE FROM T001L INTO TABLE ITAB4 FOR ALL ENTRIES IN SN_ITAB
WHERE WERKS EQ SN_ITAB-SN_WERKS .
ENDIF.
SORT: SN_ITAB,ITAB3,ITAB4.

 LOOP AT SN_ITAB.
 READ TABLE ITAB3 WITH KEY  WERKS = SN_ITAB-SN_WERKS.
 READ TABLE ITAB4 WITH KEY WERKS = SN_ITAB-SN_WERKS.
 WRITE:/ SN_ITAB-SN_MATNR,SN_ITAB-SN_MAKTX,SN_ITAB-SN_ERSDA,
SN_ITAB-SN_ERNAM,SN_ITAB-SN_LGORT,
SN_ITAB-SN_WERKS,ITAB4-NAME1,ITAB3-LGOBE.
ENDLOOP.
*&---------------------------------------------------------------------*
*& Report  ZSD_VATREPORT
```

```
*&
*&---------------------------------------------------------------------*
*&*      PROGRAM DESCRIPTION:
*       DEVELOPER: PRIYA RANJAN
*       Functional Consultant : AMOL BIDKAR /Amit Saxena
*       CREATION DATE: 27-06-2006.
*       Program Description : <VAT REPORT>
*&---------------------------------------------------------------------*

REPORT  ZSD_VATREPORT.


TABLES: VBRK , KNA1 ,  J_1IEXCHDR , J_1IMOCUST ,   VBRP .


DATA: W_KUNAG            LIKE VBRK-KUNAG.
DATA: W_NAME1            LIKE KNA1-NAME1.
DATA: W_J_1ICSTNO        LIKE J_1IMOCUST-J_1ICSTNO.
DATA: W_ORT01            LIKE KNA1-ORT01.
DATA: W_REGIO            LIKE KNA1-REGIO.
DATA: W_EXNUM            LIKE J_1IEXCHDR-EXNUM.
DATA: W_EXDAT            LIKE J_1IEXCHDR-EXDAT.
DATA: W_FKIMG            LIKE VBRP-FKIMG.
DATA: SUM                LIKE VBRP-FKLMG.
DATA: W_KZWI1            LIKE VBRP-KZWI1.
DATA: W_KZWI2            LIKE VBRP-KZWI2.
DATA: W_KZWI3            LIKE VBRP-KZWI3.
DATA: W_KZWI4            LIKE VBRP-KZWI4.
DATA: W_KZWI4_SUM        LIKE VBRP-KZWI4.
DATA: W_KZWI4A           LIKE VBRP-KZWI4.
DATA: W_KZWI4B           LIKE VBRP-KZWI4.
DATA: W_KZWI4C           LIKE VBRP-KZWI4.
```

```
DATA: W_KZWI5            LIKE VBRP-KZWI5.

DATA: W_KZWI6            LIKE VBRP-KZWI6.

DATA: W_KNUMV1           LIKE KONV-KNUMV.

DATA: W_KNUMV2           LIKE KONV-KNUMV.

DATA: W_KNUMV3           LIKE KONV-KNUMV.

DATA: W_KNUMV4           LIKE KONV-KNUMV.




DATA: BEGIN OF I_VBRK OCCURS 0,


      VBELN              LIKE VBRK-VBELN,

      FKDAT              LIKE VBRK-FKDAT,

      KUNAG              LIKE VBRK-KUNAG,

      NAME1              LIKE KNA1-NAME1,

      FKART              LIKE VBRK-FKART,

      KNUMV              LIKE VBRK-KNUMV,


END OF I_VBRK.



DATA: BEGIN OF I_HEADER OCCURS 0,


      VBELN              LIKE VBRK-VBELN,

      KUNAG              LIKE VBRK-KUNAG,

      NAME1              LIKE KNA1-NAME1,

      J_1ICSTNO          LIKE J_1IMOCUST-J_1ICSTNO,

      ORT01              LIKE KNA1-ORT01,

      REGIO              LIKE KNA1-REGIO,

      EXNUM              LIKE J_1IEXCHDR-EXNUM,

      EXDAT              LIKE J_1IEXCHDR-EXDAT,

      FKIMG              LIKE VBRP-FKIMG,
```

```
      KZWI1             LIKE VBRP-KZWI1,
      KZWI2             LIKE VBRP-KZWI2,
      KZWI3             LIKE VBRP-KZWI3,
      KZWI4             LIKE VBRP-KZWI4,
      KZWI4A            LIKE VBRP-KZWI4,
      KZWI4B            LIKE VBRP-KZWI4,
      KZWI4C            LIKE VBRP-KZWI4,
      KZWI5             LIKE VBRP-KZWI5,
      KZWI6             LIKE VBRP-KZWI6,
      KNUMV             LIKE VBRK-KNUMV,
      KNUMV1            LIKE VBRK-KNUMV,
      KNUMV2            LIKE VBRK-KNUMV,
      KNUMV3            LIKE VBRK-KNUMV,
      KNUMV4            LIKE VBRK-KNUMV,

END OF I_HEADER.

DATA: BEGIN OF I_FINAL OCCURS 0,

      VBELN             LIKE VBRK-VBELN,
      KUNAG             LIKE VBRK-KUNAG,
      NAME1             LIKE KNA1-NAME1,
      J_1ICSTNO         LIKE J_1IMOCUST-J_1ICSTNO,
      ORT01             LIKE KNA1-ORT01,
      REGIO             LIKE KNA1-REGIO,
      EXNUM             LIKE J_1IEXCHDR-EXNUM,
      EXDAT             LIKE J_1IEXCHDR-EXDAT,
      FKIMG             LIKE VBRP-FKIMG,
      KZWI1             LIKE VBRP-KZWI1,
      KZWI2             LIKE VBRP-KZWI2,
      KZWI3             LIKE VBRP-KZWI3,
      KZWI4             LIKE VBRP-KZWI4,
```

```
      KZWI4A             LIKE VBRP-KZWI4,
      KZWI4B             LIKE VBRP-KZWI4,
      KZWI4C             LIKE VBRP-KZWI4,
      KZWI5              LIKE VBRP-KZWI5,
      KZWI6              LIKE VBRP-KZWI6,
      MATNR              LIKE VBRP-MATNR,
      CHAPID             LIKE J_1IMTCHID-J_1ICHID,
      VGBEL              LIKE VBRP-VGBEL,
      ZGP                LIKE YWBOD-ZGPNO,
      ZTRNO              LIKE ZGPAS-TRNO,

END OF I_FINAL.

DATA : I_KNOV1   LIKE KONV OCCURS 0 WITH HEADER LINE.
DATA : I_KNOV2   LIKE KONV OCCURS 0 WITH HEADER LINE.
DATA : I_KNOVC2  LIKE KONV OCCURS 0 WITH HEADER LINE.
DATA : I_KNOV    LIKE KONV OCCURS 0 WITH HEADER LINE.
DATA : W_MATNR LIKE VBRP-MATNR.
DATA : W_CHAPID LIKE J_1IMTCHID-J_1ICHID.
DATA : W_VGBEL LIKE VBRP-VGBEL.
DATA : W_ZGP  LIKE YWBOD-ZGPNO.
DATA : W_TRNO LIKE ZGPAS-TRNO.



**=======================ALV DECLARATION======================

TYPE-POOLS : SLIS.

TYPES: SLIS_T_LISTHEADER TYPE SLIS_LISTHEADER OCCURS 1.

DATA : HEADING TYPE SLIS_T_LISTHEADER WITH HEADER LINE.
```

```
DATA: GT_FIELDCAT TYPE SLIS_T_FIELDCAT_ALV, "ALV Catalog Table
      GS_FIELDCAT TYPE SLIS_FIELDCAT_ALV.   "ALV Catalog Structure

DATA : EVENTSTAB      TYPE   SLIS_T_EVENT WITH HEADER LINE,
       GD_TAB_GROUP   TYPE   SLIS_T_SP_GROUP_ALV,
       GD_LAYOUT      TYPE   SLIS_LAYOUT_ALV,
       GD_REPID       LIKE   SY-REPID.



*---------------------------------------------------------------*
*                      SELECTION SCREEN                         *
*---------------------------------------------------------------*

SELECTION-SCREEN BEGIN OF BLOCK B1 .
SELECT-OPTIONS : S_FKDAT FOR VBRK-FKDAT OBLIGATORY.
SELECT-OPTIONS : S_FKRAT FOR VBRK-FKART OBLIGATORY.
SELECT-OPTIONS : S_KUNAG FOR VBRK-KUNAG .
SELECT-OPTIONS : S_VBELN FOR VBRK-VBELN .
SELECTION-SCREEN END OF BLOCK B1.



**---------------------------------------------------------------*
*                   DATA FETCHING                               *
**---------------------------------------------------------------*



AT SELECTION-SCREEN ON S_FKDAT.

START-OF-SELECTION.

  SELECT VBELN KNUMV FKDAT FKART KUNAG  FROM VBRK INTO CORRESPONDING FIELDS OF TABLE I_VBRK
```

```
  WHERE FKDAT IN S_FKDAT AND FKART IN S_FKRAT AND KUNAG IN S_KUNAG AND VBELN IN S_VBELN AND
FKSTO = ' '.



  LOOP AT I_VBRK .


    I_HEADER-VBELN = I_VBRK-VBELN.

    I_HEADER-KNUMV = I_VBRK-KNUMV.

    I_HEADER-KUNAG = I_VBRK-KUNAG.

    APPEND I_HEADER.


  ENDLOOP.


  LOOP AT I_HEADER.

    CLEAR : W_EXNUM , W_EXDAT .


    SELECT SINGLE KUNAG FROM VBRK INTO W_KUNAG WHERE VBELN = I_HEADER-VBELN.


    SELECT SINGLE NAME1 FROM KNA1 INTO W_NAME1 WHERE KUNNR = I_HEADER-KUNAG.


    SELECT SINGLE J_1ICSTNO  FROM J_1IMOCUST INTO W_J_1ICSTNO  WHERE KUNNR = I_HEADER-
KUNAG.


    SELECT SINGLE ORT01  FROM KNA1 INTO W_ORT01  WHERE KUNNR = I_HEADER-KUNAG.


    SELECT SINGLE REGIO  FROM KNA1 INTO W_REGIO  WHERE KUNNR = I_HEADER-KUNAG.


    SELECT SINGLE EXNUM EXDAT  FROM J_1IEXCHDR INTO (W_EXNUM , W_EXDAT)  WHERE RDOC =
I_HEADER-VBELN.

*Added on 28/08/2006 by Jyoti

    IF W_EXNUM  = SPACE.

      W_EXNUM = 'N/Invoiced'.

    ENDIF.
```

```
*****QUANTITY

    CLEAR : W_FKIMG , SUM.

    SELECT  FKIMG   FROM  VBRP INTO SUM   WHERE VBELN = I_HEADER-VBELN.

      W_FKIMG = SUM + W_FKIMG.

    ENDSELECT.

*****BASE VALUE * BED *  ECess

    CLEAR : W_KZWI1 , W_KZWI2 , W_KZWI3 ,  I_HEADER-KZWI1 , I_HEADER-KZWI2 , I_HEADER-KZWI3
.

    SELECT  KWERT FROM  KONV INTO W_KZWI1  WHERE KNUMV = I_HEADER-KNUMV AND KSCHL = 'ZASS'
.
      I_HEADER-KZWI1 = I_HEADER-KZWI1 + W_KZWI1.
    ENDSELECT.



    READ TABLE S_FKRAT WITH KEY LOW = 'ZDEM'  .
    IF SY-SUBRC  = 0.

      I_HEADER-KZWI2 = 0 .
      I_HEADER-KZWI3 = 0.

    ELSE.

      READ TABLE S_FKRAT WITH KEY HIGH = 'ZDEM'  .
      IF SY-SUBRC NE 0 .
```

```
        SELECT  KWERT FROM  KONV INTO W_KZWI2  WHERE KNUMV = I_HEADER-KNUMV AND KSCHL =
'JEXP' .

          I_HEADER-KZWI2 = I_HEADER-KZWI2 + W_KZWI2.

        ENDSELECT.


        SELECT  KWERT FROM  KONV INTO W_KZWI3  WHERE KNUMV = I_HEADER-KNUMV AND KSCHL =
'JECS' .

          I_HEADER-KZWI3 = I_HEADER-KZWI3 + W_KZWI3.

        ENDSELECT.


      ELSEIF SY-SUBRC NE 0.


        I_HEADER-KZWI2 = 0 .

        I_HEADER-KZWI3 = 0.


      ENDIF.

    ENDIF.




***VAT 4%

****LOGIC CHANGE BY AMIT SAXENA ON 02/08/2006


    DATA  : SUM_KZWI4 TYPE KZWI4.


    CLEAR  : I_KNOV1-KWERT , SUM_KZWI4.


    SELECT  * FROM KONV INTO CORRESPONDING FIELDS OF  TABLE I_KNOV1 WHERE KNUMV = I_HEADER-
KNUMV AND KSCHL = 'JIVP' AND KBETR = '40'.


    LOOP AT I_KNOV1 .


      SUM_KZWI4 =  I_KNOV1-KWERT + SUM_KZWI4 .
```

```
    ENDLOOP.


    I_HEADER-KZWI4 = SUM_KZWI4.



****VAT 12.5 %

****LOGIC CHANGE BY AMIT SAXENA ON 02/08/2006

    CLEAR  : I_KNOV2-KWERT , SUM_KZWI4.


    SELECT  * FROM KONV INTO CORRESPONDING FIELDS OF  TABLE I_KNOV2 WHERE KNUMV = I_HEADER-
KNUMV AND KSCHL = 'JIVP' AND KBETR = '125'.



    LOOP AT I_KNOV2 .


      SUM_KZWI4 =  I_KNOV2-KWERT + SUM_KZWI4 .


    ENDLOOP.


    I_HEADER-KZWI4A = SUM_KZWI4.


****CST 2%

****LOGIC CHANGE BY AMIT SAXENA ON 02/08/2006


    CLEAR  : I_KNOVC2-KWERT , SUM_KZWI4.


    SELECT  * FROM KONV INTO CORRESPONDING FIELDS OF  TABLE I_KNOVC2 WHERE KNUMV =
I_HEADER-KNUMV AND KSCHL = 'JIVC' AND KBETR = '20'.


    LOOP AT I_KNOVC2 .


      SUM_KZWI4 =  I_KNOVC2-KWERT + SUM_KZWI4 .
```

```
    ENDLOOP.


    I_HEADER-KZWI4B = SUM_KZWI4.


****CST 4%

****LOGIC CHANGE BY AMIT SAXENA ON 02/08/2006

    CLEAR  : I_KNOV-KWERT , SUM_KZWI4.


    SELECT  * FROM KONV INTO CORRESPONDING FIELDS OF  TABLE I_KNOV WHERE KNUMV = I_HEADER-
KNUMV AND KSCHL = 'JIVC' AND KBETR = '40'.


    LOOP AT I_KNOV .


      SUM_KZWI4 =  I_KNOV-KWERT + SUM_KZWI4 .


    ENDLOOP.


    I_HEADER-KZWI4C = SUM_KZWI4.




****FREIGHT


    DATA  : SUM_KZWI5 TYPE KZWI5.

    CLEAR : W_KZWI5 ,SUM_KZWI5 , I_HEADER-KZWI5 .


    SELECT  KWERT FROM  KONV INTO W_KZWI5  WHERE KNUMV = I_HEADER-KNUMV AND KSCHL = 'ZF00'
AND  KPOSN NE 0.

      SUM_KZWI5 = SUM_KZWI5 + W_KZWI5.

    ENDSELECT.


    I_HEADER-KZWI5 = SUM_KZWI5.
```

```
***TOTAL BILL AMOUNT


    DATA  : SUM_KZWI6 TYPE KZWI6.

    DATA  : W_KWERT TYPE KZWI6.

    CLEAR : W_KZWI6 ,SUM_KZWI6 .


****LOGIC CHANGE BY AMIT SAXENA ON 02/08/2006


    DATA : BEGIN OF I_TOTAL OCCURS 0 ,


           KNUMV LIKE KONV-KNUMV,

           KWERT LIKE KONV-KWERT,

           KPOSN LIKE KONV-KPOSN,

           END OF I_TOTAL.




    READ TABLE S_FKRAT WITH KEY LOW = 'ZDEM' .

    IF SY-SUBRC = 0 .

      CLEAR : SUM_KZWI6 .

      SELECT    *  FROM  KONV INTO  CORRESPONDING FIELDS OF TABLE  I_TOTAL    WHERE KNUMV =
I_HEADER-KNUMV AND

      ( KSCHL = 'JIVC' OR   KSCHL = 'ZASS' OR    KSCHL = 'JIVP' OR KSCHL = 'ZF00' ).




      LOOP AT I_TOTAL WHERE KPOSN NE 0 .

        SUM_KZWI6 =  I_TOTAL-KWERT + SUM_KZWI6 .

      ENDLOOP.

      I_HEADER-KZWI6 = SUM_KZWI6.


    ELSEIF SY-SUBRC NE 0.
```

```
      READ TABLE S_FKRAT WITH KEY HIGH = 'ZDEM' .

      IF SY-SUBRC NE 0.

        CLEAR : SUM_KZWI6 .

        SELECT    *  FROM  KONV INTO  CORRESPONDING FIELDS OF TABLE  I_TOTAL    WHERE KNUMV
= I_HEADER-KNUMV AND

        ( KSCHL = 'JIVC' OR   KSCHL = 'ZASS' OR KSCHL = 'JEXP'   OR  KSCHL = 'JECS'    OR
KSCHL = 'JIVP' OR KSCHL = 'ZF00' ).




        LOOP AT I_TOTAL WHERE KPOSN NE 0 .

          SUM_KZWI6 =  I_TOTAL-KWERT + SUM_KZWI6 .

        ENDLOOP.

        I_HEADER-KZWI6 = SUM_KZWI6.


      ELSEIF SY-SUBRC EQ 0.

        CLEAR : SUM_KZWI6 .


        SELECT    *  FROM  KONV INTO  CORRESPONDING FIELDS OF TABLE  I_TOTAL    WHERE KNUMV
= I_HEADER-KNUMV AND

     ( KSCHL = 'JIVC' OR   KSCHL = 'ZASS' OR    KSCHL = 'JIVP' OR KSCHL = 'ZF00' ).




        LOOP AT I_TOTAL WHERE KPOSN NE 0 .

          SUM_KZWI6 =  I_TOTAL-KWERT + SUM_KZWI6 .

        ENDLOOP.

        I_HEADER-KZWI6 = SUM_KZWI6.


      ENDIF.

    ENDIF.


***VALUE TRANSFER FROM VARIABLE INTO I_HEADER


    I_HEADER-NAME1       =    W_NAME1.
```

```
    I_HEADER-J_1ICSTNO   =    W_J_1ICSTNO.
    I_HEADER-ORT01       =    W_ORT01.
    I_HEADER-REGIO       =    W_REGIO .
    I_HEADER-EXNUM       =    W_EXNUM.
    I_HEADER-EXDAT       =    W_EXDAT.
    I_HEADER-FKIMG       =    W_FKIMG.

    MODIFY I_HEADER.

  ENDLOOP.

END-OF-SELECTION.

***TABLE I_FINAL FOR FINAL OUTPUT ALV VALUE

*IF I_HEADER-EXNUM EQ ' '.

*ENDIF.

  LOOP AT I_HEADER.

    MOVE I_HEADER-VBELN       TO I_FINAL-VBELN.
    MOVE I_HEADER-KUNAG       TO I_FINAL-KUNAG.
    MOVE I_HEADER-NAME1       TO I_FINAL-NAME1.
    MOVE I_HEADER-J_1ICSTNO   TO I_FINAL-J_1ICSTNO.
    MOVE I_HEADER-ORT01       TO I_FINAL-ORT01.
    MOVE I_HEADER-REGIO       TO I_FINAL-REGIO.
    MOVE I_HEADER-EXNUM       TO I_FINAL-EXNUM.
    MOVE I_HEADER-EXDAT       TO I_FINAL-EXDAT.
    MOVE I_HEADER-FKIMG       TO I_FINAL-FKIMG.
    MOVE I_HEADER-KZWI1       TO I_FINAL-KZWI1.
    MOVE I_HEADER-KZWI2       TO I_FINAL-KZWI2.
```

```
  MOVE I_HEADER-KZWI3       TO I_FINAL-KZWI3.
  MOVE I_HEADER-KZWI4       TO I_FINAL-KZWI4.
  MOVE I_HEADER-KZWI4A      TO I_FINAL-KZWI4A.
  MOVE I_HEADER-KZWI4B      TO I_FINAL-KZWI4B.
  MOVE I_HEADER-KZWI4C      TO I_FINAL-KZWI4C.
  MOVE I_HEADER-KZWI5       TO I_FINAL-KZWI5.
  MOVE I_HEADER-KZWI6       TO I_FINAL-KZWI6.

  APPEND I_FINAL.

ENDLOOP.



DELETE I_FINAL WHERE EXNUM = SPACE.

LOOP AT I_FINAL.
  SELECT SINGLE MATNR VGBEL FROM VBRP INTO (W_MATNR,W_VGBEL) WHERE VBELN = I_FINAL-VBELN.
  MOVE W_MATNR TO I_FINAL-MATNR.
  MOVE W_VGBEL TO I_FINAL-VGBEL.
  MODIFY I_FINAL.
ENDLOOP.

LOOP AT I_FINAL.
  SELECT SINGLE J_1ICHID FROM J_1IMTCHID INTO W_CHAPID WHERE MATNR = I_FINAL-MATNR.
  MOVE W_CHAPID TO I_FINAL-CHAPID.
  MODIFY I_FINAL.
ENDLOOP.

LOOP AT I_FINAL.
  SELECT SINGLE ZGPNO FROM YWBOD INTO W_ZGP WHERE VBELN = I_FINAL-VGBEL.
  MOVE W_ZGP TO I_FINAL-ZGP.
  MODIFY I_FINAL.
```

```
  ENDLOOP.



  LOOP AT I_FINAL.
    SELECT SINGLE TRNO FROM ZGPAS INTO W_TRNO WHERE ZGPNO = I_FINAL-ZGP.
    MOVE W_TRNO TO I_FINAL-ZTRNO.
    MODIFY I_FINAL.
  ENDLOOP.

**=====================ALV GRID DISPLAY=======================*"change on 08/08/06
*
  PERFORM BUILT_FIELD_CATALOG.
  GD_REPID = SY-REPID.
  CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
    EXPORTING
      I_CALLBACK_PROGRAM = GD_REPID
      I_BACKGROUND_ID    = 'ALV_BACKGROUND'
      I_GRID_TITLE       = 'SD VAT REPORT'
      IT_FIELDCAT        = GT_FIELDCAT[]
      I_SAVE             = 'X'
    TABLES
      T_OUTTAB           = I_FINAL.


*&---------------------------------------------------------------------
*
*&      Form  BUILT_FIELD_CATALOG
*&---------------------------------------------------------------------
*
*       text
*----------------------------------------------------------------------
*
FORM BUILT_FIELD_CATALOG.
```

```
GS_FIELDCAT-COL_POS      = '1'.
GS_FIELDCAT-FIELDNAME    = 'VBELN'.
GS_FIELDCAT-OUTPUTLEN    = 18.
GS_FIELDCAT-SELTEXT_L    = 'BILL NO'.
APPEND GS_FIELDCAT TO GT_FIELDCAT.


GS_FIELDCAT-COL_POS      = '1'.
GS_FIELDCAT-FIELDNAME    = 'MATNR'.
GS_FIELDCAT-OUTPUTLEN    = 18.
GS_FIELDCAT-SELTEXT_L    = 'Material Number'.
APPEND GS_FIELDCAT TO GT_FIELDCAT.


GS_FIELDCAT-COL_POS      = '1'.
GS_FIELDCAT-FIELDNAME    = 'CHAPID'.
GS_FIELDCAT-OUTPUTLEN    = 18.
GS_FIELDCAT-SELTEXT_L    = 'Chapter ID'.
APPEND GS_FIELDCAT TO GT_FIELDCAT.



GS_FIELDCAT-COL_POS      = '2'.
GS_FIELDCAT-FIELDNAME    = 'KUNAG'.
GS_FIELDCAT-OUTPUTLEN    = 18.
GS_FIELDCAT-SELTEXT_L    = 'CUSTOMER NO'.
APPEND GS_FIELDCAT TO GT_FIELDCAT.


GS_FIELDCAT-COL_POS      = '3'.
GS_FIELDCAT-FIELDNAME    = 'NAME1'.
```

```
GS_FIELDCAT-OUTPUTLEN    = 18.
GS_FIELDCAT-SELTEXT_L    = 'CUSTOMER NAME'.
APPEND GS_FIELDCAT TO GT_FIELDCAT.


GS_FIELDCAT-COL_POS      = '4'.
GS_FIELDCAT-FIELDNAME    = 'J_1ICSTNO'.
GS_FIELDCAT-OUTPUTLEN    = 18.
GS_FIELDCAT-SELTEXT_L    = 'TIN / CST / RC'.
APPEND GS_FIELDCAT TO GT_FIELDCAT.


GS_FIELDCAT-COL_POS      = '5'.
GS_FIELDCAT-FIELDNAME    = 'ORT01'.
GS_FIELDCAT-OUTPUTLEN    = 18.
GS_FIELDCAT-SELTEXT_L    = 'CITY'.
APPEND GS_FIELDCAT TO GT_FIELDCAT.


GS_FIELDCAT-COL_POS      = '6'.
GS_FIELDCAT-FIELDNAME    = 'REGIO'.
GS_FIELDCAT-OUTPUTLEN    = 18.
GS_FIELDCAT-SELTEXT_L    = 'STATE'.
APPEND GS_FIELDCAT TO GT_FIELDCAT.


GS_FIELDCAT-COL_POS      = '7'.
GS_FIELDCAT-FIELDNAME    = 'EXNUM'.
GS_FIELDCAT-OUTPUTLEN    = 18.
GS_FIELDCAT-SELTEXT_L    = 'EXCISE INVOICE NO'.
APPEND GS_FIELDCAT TO GT_FIELDCAT.
```

```
  GS_FIELDCAT-COL_POS      = '8'.
  GS_FIELDCAT-FIELDNAME    = 'EXDAT'.
  GS_FIELDCAT-OUTPUTLEN    = 18.
*  GS_FIELDCAT-SELTEXT_L    = 'BILL DATE'.
  GS_FIELDCAT-SELTEXT_L    = 'INVOICE DATE'.
  APPEND GS_FIELDCAT TO GT_FIELDCAT.


  GS_FIELDCAT-COL_POS      = '9'.
  GS_FIELDCAT-FIELDNAME    = 'FKIMG'.
  GS_FIELDCAT-OUTPUTLEN    = 18.
  GS_FIELDCAT-SELTEXT_L    = 'QUANTITY'.
  APPEND GS_FIELDCAT TO GT_FIELDCAT.


  GS_FIELDCAT-COL_POS      = '10'.
  GS_FIELDCAT-FIELDNAME    = 'KZWI1'.
  GS_FIELDCAT-OUTPUTLEN    = 18.
  GS_FIELDCAT-SELTEXT_L    = 'BASE VALUE'.
  APPEND GS_FIELDCAT TO GT_FIELDCAT.


  GS_FIELDCAT-COL_POS      = '11'.
  GS_FIELDCAT-FIELDNAME    = 'KZWI2'.
  GS_FIELDCAT-OUTPUTLEN    = 18.
  GS_FIELDCAT-SELTEXT_L    = 'BED'.
  APPEND GS_FIELDCAT TO GT_FIELDCAT.


  GS_FIELDCAT-COL_POS      = '12'.
  GS_FIELDCAT-FIELDNAME    = 'KZWI3'.
```

```
GS_FIELDCAT-OUTPUTLEN    = 18.
GS_FIELDCAT-SELTEXT_L    = 'ECESS'.
APPEND GS_FIELDCAT TO GT_FIELDCAT.


GS_FIELDCAT-COL_POS      = '13'.
GS_FIELDCAT-FIELDNAME    = 'KZWI4'.
GS_FIELDCAT-OUTPUTLEN    = 18.
GS_FIELDCAT-SELTEXT_L    = 'VAT 4%'.
APPEND GS_FIELDCAT TO GT_FIELDCAT.


GS_FIELDCAT-COL_POS      = '14'.
GS_FIELDCAT-FIELDNAME    = 'KZWI4A'.
GS_FIELDCAT-OUTPUTLEN    = 18.
GS_FIELDCAT-SELTEXT_L    = 'VAT 12.5%'.
APPEND GS_FIELDCAT TO GT_FIELDCAT.


GS_FIELDCAT-COL_POS      = '15'.
GS_FIELDCAT-FIELDNAME    = 'KZWI4B'.
GS_FIELDCAT-OUTPUTLEN    = 18.
GS_FIELDCAT-SELTEXT_L    = 'CST 2%'.
APPEND GS_FIELDCAT TO GT_FIELDCAT.


GS_FIELDCAT-COL_POS      = '16'.
GS_FIELDCAT-FIELDNAME    = 'KZWI4C'.
GS_FIELDCAT-OUTPUTLEN    = 18.
GS_FIELDCAT-SELTEXT_L    = 'CST 4%'.
APPEND GS_FIELDCAT TO GT_FIELDCAT.
```

```
  GS_FIELDCAT-COL_POS      = '17'.
  GS_FIELDCAT-FIELDNAME    = 'KZWI5'.
  GS_FIELDCAT-OUTPUTLEN    = 18.
  GS_FIELDCAT-SELTEXT_L    = 'FREIGHT'.
  APPEND GS_FIELDCAT TO GT_FIELDCAT.


  GS_FIELDCAT-COL_POS      = '18'.
  GS_FIELDCAT-FIELDNAME    = 'KZWI6'.
  GS_FIELDCAT-OUTPUTLEN    = 18.
  GS_FIELDCAT-SELTEXT_L    = 'TOTAL BILL AMOUNT'.
  APPEND GS_FIELDCAT TO GT_FIELDCAT.


  GS_FIELDCAT-COL_POS      = '18'.
  GS_FIELDCAT-FIELDNAME    = 'ZTRNO'.
  GS_FIELDCAT-OUTPUTLEN    = 18.
  GS_FIELDCAT-SELTEXT_L    = 'Truck No.'.
  APPEND GS_FIELDCAT TO GT_FIELDCAT.



ENDFORM.                    "BUILT_FIELD_CATALOG



*&---------------------------------------------------------------------*
*& REPORT  ZSDREP.                                                     *
*& Created On : 19/10/2006                                             *
*& Company : Crystal Phosphates Ltd.                                   *
*& Author : Navneet Kumar                                              *
```

```
*&---------------------------------------------------------------------*
*& Description : SD Cycle Report                                       *
*&---------------------------------------------------------------------*

REPORT  zsdrep  LINE-SIZE 700 NO STANDARD PAGE HEADING .    "#EC *


*&---------------------------------------------------------------------*
*& Table Decleration                                                   *
*&---------------------------------------------------------------------*

TABLES : vbak,        " Sales Document: Header Data
         zsdtab,
         vbap,        " Sales Document: Item Data
         vbkd,        " Sales Document: Business Data
         vbpa,        " Sales Document: Partner
         konp,        " Conditions (Item)
         konv,        " Conditions (Transaction Data)
         likp,        " SD Document: Delivery Header Data
         lips,        " SD document: Delivery: Item data
         vbrk,        " Billing Document: Header Data
         vbrp,        " Billing Document: Item Data
         vbfa,        " Sales Document Flow
         makt,        " Material Descriptions
         tvko,        " Organizational Unit: Sales Organizations
         kna1,        " General Data in Customer Master
         adrc,        " Addresses (Business Address Services)
         j_1iexchdr,  " Excise invoice header detail
         j_1isrgrps,  " Excise Document Series
         j_1iexgrps,  " Excise Groups
         tpart,       " Partner function
         t005t.       " Country Names
```

```
*&---------------------------------------------------------------------*
*& Selection Screen                                                    *
*&---------------------------------------------------------------------*

SELECTION-SCREEN : BEGIN OF BLOCK b3 WITH FRAME TITLE text-003.
SELECT-OPTIONS : s_kunnr1 FOR kna1-kunnr,      " Sales Person Code
                 s_AUDAT  FOR vbak-AUDAT,      " Document Date.
                 s_FKART  FOR vbrk-FKART.      " Billing Type.
SELECTION-SCREEN : END OF BLOCK b3.

SELECTION-SCREEN : BEGIN OF BLOCK b1 WITH FRAME TITLE text-001.
SELECT-OPTIONS : s_bukrs FOR vbak-bukrs_vf,    " Company Code
                 s_vkorg FOR vbak-vkorg,       " Sales Organization
                 s_vtweg FOR vbak-vtweg,       " Distribution Channel
                 s_spart FOR vbak-spart,       " Division
                 s_matnr FOR vbap-matnr,       " Material
                 s_kunnr FOR vbak-kunnr,       " Sold to party
                 s_taxk1 FOR vbak-taxk1,       " Form Type Used
                 s_werks FOR vbap-werks,       " Plant
                 s_gsber FOR vbrp-gsber.       " Business Area
SELECTION-SCREEN : END OF BLOCK b1.




*&---------------------------------------------------------------------*
*& Internal Table Decleration                                          *
*&---------------------------------------------------------------------*



DATA : BEGIN OF i_vbak OCCURS 0,                            "#EC *
```

```
        kunnr     LIKE vbak-kunnr,         " Sold to party
        vbtyp     LIKE vbak-vbtyp,
        vkorg     LIKE vbak-vkorg,         " Sales Organization
        vtweg     LIKE vbak-vtweg,         " Distribution Channel
        spart     LIKE vbak-spart,         " Division
        bukrs     LIKE vbak-bukrs_vf,      " Company Code
        vkbur     LIKE vbak-vkbur,         " Sales office
        vkgrp     LIKE vbak-vkgrp,         " Sales group
        vbeln     LIKE vbak-vbeln,         " Sales Order No
        auart     LIKE vbak-auart,         " Sales Order Type
        taxk1     LIKE vbak-taxk1,         " Form Type Used
        erdat     LIKE vbak-erdat,         " Sales Order Date
        augru     LIKE vbak-augru,         " Order Reason  ########
        vgbel     LIKE vbak-vgbel,          " ####
        gsber     LIKE vbrp-gsber,         " Business Area ########
        matnr     LIKE vbap-matnr,         " Material Number
        posnr     LIKE vbap-posnr,         " Sales Document Item
        kwmeng    LIKE vbap-kwmeng,        " Sales Order Matrial Quantity
        zieme     LIKE vbap-zieme,         " UoM
        route     LIKE vbap-route,         " Route
        werks     LIKE vbap-werks,         " Plant
        netwr     LIKE vbap-netwr,         " Net value of the order item in document
currency(cr. note Amt.) #####
        vstel     LIKE vbap-vstel,         " Shipping Point
        knumh     LIKE vbap-knumh,         " Number of condition record
        audat     LIKE vbak-audat,         " Dr note date
        c_vbeln   LIKE vbak-vbeln,         " Credit Sales Order Type ######
        d_vbeln   LIKE vbak-vbeln,         " Debit Sales Order Type  ######
        ex_vbeln  LIKE vbrp-vbeln,         " Ref Doc No.
        shkunnr   LIKE vbpa-kunnr,         " Ship to Party
        prkunnr   LIKE vbpa-kunnr,         "
        bstkd     LIKE vbkd-bstkd,         " Purchase Order Number
```

```
bstdk     LIKE vbkd-bstdk,         " Purchase Order Date
inco1     LIKE vbkd-inco1,         " Incoterms (part 1)
inco2     LIKE vbkd-inco2,         " Incoterms (part 2)
zterm     LIKE vbkd-zterm,         " Payment Terms
crfkdat   LIKE vbkd-fkdat,         " Cr Note Date ########
frno(4)   TYPE n,                  " Form Number
maktx     LIKE makt-maktx,         " Material Description
ntgew     LIKE likp-ntgew,         " Delivery Quantity(net weight)
dvbeln    LIKE lips-vbeln,         " Delivery Number
bldat     LIKE likp-bldat,         " Delivery Date
lfdat     LIKE likp-lfdat,         " Delivery Date ####
lfart     LIKE likp-lfart,         " Delivery Type
lfimg     LIKE lips-lfimg,
ivbeln    LIKE vbrk-vbeln,         " Invoice Number
knumv     LIKE vbrk-knumv,
fkdat     LIKE vbrk-fkdat,         " Billing Date
fkart     LIKE vbrk-fkart,         " Billing Type
exnum     LIKE j_1iexchdr-exnum,   " Official Excise Document Number ####
exdat     LIKE j_1iexchdr-exdat,   " Excise Document Date ####
exgrp     LIKE j_1iexchdr-exgrp,   " Excise Group ####
srgrp     LIKE j_1iexchdr-srgrp,   " Excise Document Series Group ####
chapid    LIKE j_1iexcdtl-chapid,  " Chapter ID ###
vvbeln    LIKE vbfa-vbeln,         " Subsequent sales and distribution document ####
vbeln_m   LIKE vbfa-vbeln,         " Subsequent sales and distribution document ####
name1     LIKE adrc-name1,         " Address of Sold-to-Party ###
name2     LIKE adrc-name2,         " Address of Sold-to-Party ###
city1     LIKE adrc-city1,         " Address of Sold-to-Party ###
street    LIKE adrc-street,        " Address of Sold-to-Party ###
post_code1  LIKE adrc-post_code1,  " Address of Sold-to-Party ###
country   LIKE t005t-landx,
cr_vbeln  LIKE vbak-vbeln,
cr_netwr  LIKE vbap-netwr,
```

```
vebln2    LIKE vbrp-vbeln,         "^^^^^^^^^
name_sales LIKE kna1-name1,
form     LIKE zsdtab-form,
 B001    LIKE KONV-KBETR,
 JEXQ    LIKE KONV-KBETR,
 JEXT    LIKE KONV-KBETR,
 JCES    LIKE KONV-KBETR,
 JIN7    LIKE KONV-KBETR,
 JIN8    LIKE KONV-KBETR,
 ZDCP    LIKE KONV-KBETR,
 ZDBP    LIKE KONV-KBETR,
 ZDFR    LIKE KONV-KBETR,
 ZDER    LIKE KONV-KBETR,
 ZTCS    LIKE KONV-KBETR,
 ZPAC    LIKE KONV-KBETR,
 ZFR1    LIKE KONV-KBETR,
 ZPR0    LIKE KONV-KBETR,
 ZCRT    LIKE KONV-KBETR,
 JEXP    LIKE KONV-KBETR,
 ZJD3    LIKE KONV-KBETR,
 JCEX    LIKE KONV-KBETR,
 ZSFR    LIKE KONV-KBETR,
 ZAIR    LIKE KONV-KBETR,
 ZINS    LIKE KONV-KBETR,
 ZCHA    LIKE KONV-KBETR,
 JTRD    LIKE KONV-KBETR,
 JCST    LIKE KONV-KBETR,
 JLST    LIKE KONV-KBETR,
 JIN6    LIKE KONV-KBETR,
 JFCG    LIKE KONV-KBETR,
 ZJDI    LIKE KONV-KBETR,
 ZJD2    LIKE KONV-KBETR,
```

```
         ZFRE    LIKE KONV-KBETR,
         ZFRO    LIKE KONV-KBETR,
       END OF i_vbak.

DATA : wa_fin  LIKE  i_vbak.

DATA : Begin of condition OCCURS 0,
         knumv   LIKE konv-knumv,
         kschl   LIKE konv-kschl,
         kbetr   LIKE konv-kbetr,
       END OF condition.

DATA : BEGIN OF i_tpart OCCURS 0,
         parvw  LIKE tpart-parvw,
         vtext  LIKE tpart-vtext,
        END OF i_tpart.

DATA : BEGIN OF x_tpart OCCURS 0,
         parvw  LIKE tpart-parvw,
         vtext  LIKE tpart-vtext,
        END OF x_tpart.

DATA : BEGIN OF i_kna1 OCCURS 0,                          "#EC *
          kunnr LIKE kna1-kunnr,
          name1 LIKE kna1-name1,
        END OF i_kna1.

DATA : BEGIN OF i_vbpa OCCURS 0,                          "#EC *
          kunnr LIKE vbpa-kunnr,
          vbeln LIKE vbpa-vbeln,
          parvw LIKE vbpa-parvw,
        END OF i_vbpa.
```

```
DATA : BEGIN OF x_vbpa OCCURS 0,                          "#EC *
         kunnr LIKE vbpa-kunnr,
         parvw LIKE vbpa-parvw,
         vbeln LIKE vbpa-vbeln,
       END OF x_vbpa.


DATA : BEGIN OF i_vbfa OCCURS 0,                          "#EC *
        vbeln LIKE vbfa-vbeln,
        vbelv LIKE vbfa-vbelv,
       END OF i_vbfa.


DATA : BEGIN OF i_makt OCCURS 0,                          "#EC *
        maktx   LIKE makt-maktx,
        matnr   LIKE makt-matnr,
       END OF i_makt.

DATA : BEGIN OF i_vbkd OCCURS 0,                          "#EC *
        bstkd   LIKE vbkd-bstkd,
        bstdk   LIKE vbkd-bstdk,
        inco1   LIKE vbkd-inco1,
        inco2   LIKE vbkd-inco2,
        zterm   LIKE vbkd-zterm,
        fkdat   LIKE vbkd-fkdat,
        vbeln   LIKE vbkd-vbeln,
       END OF i_vbkd.

DATA : BEGIN OF x_likp OCCURS 0,                          "#EC *
        lfdat   LIKE likp-lfdat,
        lfart   LIKE likp-lfart,
```

```
         lfimg   LIKE lips-lfimg,
         vgbel   LIKE lips-vgbel,
         matnr   LIKE  lips-matnr,
         vbeln   LIKE lips-vbeln,
        END OF x_likp.

DATA : BEGIN OF x_vbfa OCCURS 0,                          "#EC *
         vbeln LIKE vbfa-vbeln,
         vbelv LIKE vbfa-vbelv,
        END OF x_vbfa.

DATA : BEGIN OF i_lips OCCURS 0,                          "#EC *
         lfimg   LIKE lips-lfimg,
         matnr   LIKE lips-matnr,
         vbeln   LIKE lips-vbeln,
         vgbel   LIKE vbrp-vgbel,
         vbeln2  LIKE vbrp-vbeln,
         END OF i_lips.

DATA : BEGIN OF i_vbrk OCCURS 0,                          "#EC *
         vbeln  LIKE  vbrk-vbeln,
         fkdat  LIKE  vbrk-fkdat,
         fkart  LIKE  vbrk-fkart,
         knumv  LIKE  vbrk-knumv,
         vkorg  LIKE  vbrk-vkorg,
        END OF i_vbrk.

DATA : BEGIN OF i_vbrp OCCURS 0,                          "#EC *
         gsber   LIKE vbrp-gsber,
         spart   LIKE vbrp-spart,
         vbeln   LIKE vbrp-vbeln,
        END OF i_vbrp.
```

```
DATA : BEGIN OF x_vbak OCCURS 0,
         kunnr LIKE vbak-kunnr,                          "#EC *
         vbtyp LIKE vbak-vbtyp,
         vbeln LIKE vbak-vbeln,
         audat LIKE vbak-audat,
       END OF x_vbak.


DATA : BEGIN OF i_adrc OCCURS 0,                          "#EC *
       name1    LIKE adrc-name1,
       name2    LIKE adrc-name2,
       city1    LIKE adrc-city1,
       street    LIKE adrc-street,
       post_code1  LIKE adrc-post_code1,
       kunnr    LIKE kna1-kunnr,
       END OF i_adrc.

DATA : BEGIN OF i_j_1iexcdtl OCCURS 0,                    "#EC *
        chapid  LIKE j_1iexcdtl-chapid,
        matnr   LIKE j_1iexcdtl-matnr,
       END OF i_j_1iexcdtl.

DATA : BEGIN OF i_j_1iexchdr OCCURS 0,
       kunag   LIKE j_1iexchdr-kunag,
       werks   LIKE j_1iexchdr-werks,
       exnum   LIKE j_1iexchdr-exnum,
       exdat   LIKE j_1iexchdr-exdat,
       exgrp   LIKE j_1iexchdr-exgrp,
       srgrp   LIKE j_1iexchdr-srgrp,
       rdoc    LIKE j_1iexchdr-rdoc,
```

```
        END OF i_j_1iexchdr.

DATA : Begin of i_join OCCURS 0,
         vbeln   LIKE  vbrp-vbeln,
         gsber   LIKE  vbrp-gsber,
         vgbel  LIKE  lips-vgbel,
         end of i_join.

DATA : Begin of i_sdtab OCCURS 0.
         include structure zsdtab.
DATA : end of i_sdtab.
*&---------------------------------------------------------------------*
*& Work Area Decleration                                               *
*&---------------------------------------------------------------------*


DATA : wa_vbak LIKE i_vbak,
       wa_sdtab LIKE i_sdtab,
       wa_xvbak LIKE x_vbak,
       wa_vbpa LIKE i_vbpa,
       wa_xvbpa LIKE x_vbpa,
       wa_makt LIKE i_makt,
       wa_vbkd LIKE i_vbkd,
       wa_vbrk LIKE i_vbrk,
       wa_xlikp LIKE x_likp,
       wa_vbrp LIKE i_vbrp,
       wa_vbfa  LIKE i_vbfa,
       wa_xvbfa  LIKE x_vbfa,
       wa_adrc  LIKE i_adrc,
       wa_j_1iexcdtl LIKE i_j_1iexcdtl,
       wa_j_1iexchdr LIKE i_j_1iexchdr,
       wa_xtpart  LIKE x_tpart,
```

```
       wa_tpart  LIKE i_tpart,
       wa_kna1   LIKE i_kna1,
       wa_join   LIKE i_join,
       wa_cond  LIKE condition.


*&---------------------------------------------------------------------*
*& Global Data Decleration                                             *
*&---------------------------------------------------------------------*

DATA : cursorfield(20).

*&---------------------------------------------------------------------*
*& Constants                                                           *
*&---------------------------------------------------------------------*

DATA : c_c       TYPE c VALUE 'C',
       c_cc      TYPE c VALUE 'K',
       c_cd      TYPE c VALUE 'L',
       c_j       TYPE c VALUE 'J',
       c_m       TYPE c VALUE 'M',
       c_sh(2)   TYPE c VALUE 'WE',
       cr(4)     TYPE c VALUE 'G2',
       dr(4)     TYPE c VALUE 'L2',
       total     LIKE vbap-kwmeng VALUE 0,
       total2    LIKE lips-lfimg VALUE 0,
       total3    LIKE KONV-KBETR VALUE 0.
*&---------------------------------------------------------------------*
*& Initialization                                                      *
*&---------------------------------------------------------------------*

START-OF-SELECTION.
```

```
  PERFORM logic.

  PERFORM modify_table.

  PERFORM pricing_cond.

END-OF-SELECTION.

  PERFORM display.


*&---------------------------------------------------------------------*

*& Form  logic

*&---------------------------------------------------------------------*

*  Description : Logic for data Extraction

*---------------------------------------------------------------------*

FORM logic .

  SORT i_vbak BY vbeln.

  SELECT a~vbtyp a~vkorg a~vtweg a~spart a~bukrs_vf a~vkbur a~vkgrp a~vbeln

a~auart a~kunnr a~taxk1 a~erdat a~augru a~gsber a~vgbel b~matnr b~posnr b~kwmeng b~zieme
b~route b~werks b~netwr b~vstel b~knumh

INTO corresponding fields of i_vbak FROM vbak AS a


  INNER JOIN

  vbap AS b ON a~vbeln = b~vbeln


  WHERE ( a~vbtyp    =   c_c    OR    a~vbtyp    =   c_cc     OR     a~vbtyp    =   c_cd )
AND

        a~bukrs_vf IN s_bukrs AND

        a~vkorg    IN s_vkorg AND

        a~vtweg    IN s_vtweg AND

        a~spart    IN s_spart AND

        a~kunnr    IN s_kunnr AND

        a~taxk1    IN s_taxk1 AND

        a~audat    IN s_audat AND

        b~matnr    IN s_matnr AND

        b~werks    IN s_werks .
```

```
append i_vbak.
clear i_vbak.
endselect.

*&---------------------------------------------------------------------*
*& VBFA Entry                                                          *
*&---------------------------------------------------------------------*
  SORT i_vbak BY vbeln.
  SELECT  vbeln vbelv FROM vbfa INTO TABLE i_vbfa FOR ALL ENTRIES IN i_vbak
  WHERE vbelv = i_vbak-vbeln AND
        posnn = i_vbak-posnr AND
        vbtyp_n = c_j.

  SELECT vbeln vbelv FROM vbfa INTO TABLE x_vbfa FOR ALL ENTRIES IN i_vbak
     WHERE vbelv = i_vbak-vbeln AND
           posnn = i_vbak-posnr AND
           vbtyp_n = c_m.
*&---------------------------------------------------------------------*
*& LIKP & LIPS Entry                                                   *
*&---------------------------------------------------------------------*

loop at i_vbfa.
  SELECT a~lfdat a~lfart b~lfimg b~vgbel b~matnr b~vbeln  " c~vbeln
  INTO corresponding fields of x_likp FROM likp AS a
  INNER JOIN
  lips AS b ON a~vbeln = b~vbeln

  WHERE
      a~vbeln  = i_vbfa-vbeln.
 append x_likp.
 clear x_likp.
 endselect.
```

```
 endloop.

*&---------------------------------------------------------------------*
*& X_VBAK Entry                                                        *
*&---------------------------------------------------------------------*

  SELECT kunnr audat vbtyp vbeln FROM vbak INTO CORRESPONDING FIELDS OF x_vbak
       WHERE vbtyp = c_cc OR
             vbtyp = c_cd.

  APPEND X_VBAK.
  CLEAR X_VBAK.

  ENDSELECT.

*&---------------------------------------------------------------------*
*& VBPA Entry                                                          *
*&---------------------------------------------------------------------*
  SORT i_vbak BY vbeln.
  SELECT kunnr vbeln parvw FROM vbpa INTO TABLE i_vbpa  FOR ALL ENTRIES IN i_vbak
  WHERE vbeln = i_vbak-vbeln AND
        parvw = c_sh.

  SELECT kunnr parvw vbeln FROM vbpa INTO TABLE x_vbpa  FOR ALL ENTRIES IN i_vbak
  WHERE vbeln = i_vbak-vbeln AND
        KUNNR IN S_KUNNR1 AND
        ( parvw = 'ZA' OR
        parvw = 'ZB' OR
        parvw = 'ZC' ).


*&---------------------------------------------------------------------*
```

```
*& MAKT Entry                                                          *
*&---------------------------------------------------------------------*
  SORT i_vbak BY matnr.

  SELECT  maktx matnr FROM makt INTO TABLE i_makt FOR ALL ENTRIES IN i_vbak
  WHERE matnr = i_vbak-matnr AND
        spras = sy-langu.

*&---------------------------------------------------------------------*
*& VBKD Entry                                                          *
*&---------------------------------------------------------------------*
  SORT i_vbak BY vbeln.
  SELECT  bstkd bstdk inco1 inco2 zterm fkdat vbeln FROM vbkd INTO TABLE i_vbkd FOR ALL
ENTRIES IN i_vbak
  WHERE vbeln = i_vbak-vbeln.



*&---------------------------------------------------------------------*
*& j_1iexchdr Entry                                                    *
*&---------------------------------------------------------------------*
  SORT i_vbak BY kunnr.
  SELECT   kunag werks exnum exdat exgrp srgrp rdoc FROM j_1iexchdr
  INTO TABLE i_j_1iexchdr FOR ALL ENTRIES IN i_vbak
  WHERE kunag = i_vbak-kunnr AND
       ( TRNTYP = 'DLFC'  OR  TRNTYP = 'ARE1'  OR  TRNTYP = 'ARE3' ).

*&---------------------------------------------------------------------*
*& j_1iexcdtl Entry                                                    *
*&---------------------------------------------------------------------*
  SORT i_vbak BY werks.
  SELECT  chapid matnr FROM j_1iexcdtl INTO TABLE i_j_1iexcdtl FOR ALL ENTRIES IN i_vbak
  WHERE matnr = i_vbak-matnr.
```

```
*&---------------------------------------------------------------------*
*& VBRP(i_join) Entry                                                  *
*&---------------------------------------------------------------------*
loop at i_vbfa.
SELECT a~vbeln a~gsber b~vgbel into corresponding fields of i_join from
  vbrp as a inner join
  lips as b
  on a~vgbel = b~vbeln
    WHERE
      b~vbeln  = i_vbfa-vbeln
      AND a~gsber IN s_gsber.
  append i_join.
  clear i_join.
  endselect.
  endloop.


*&---------------------------------------------------------------------*
*& VBRK Entry                                                          *
*&---------------------------------------------------------------------*
  SORT i_vbak BY vbeln.
  SELECT  vbeln fkdat fkart knumv vkorg FROM vbrk INTO TABLE i_vbrk FOR ALL ENTRIES IN
x_vbfa
     WHERE vbeln = x_vbfa-vbeln AND
           fkart IN s_fkart.


*&---------------------------------------------------------------------*
*& ADRC Entry                                                          *
*&---------------------------------------------------------------------*


  SELECT  a~name1 a~name2 a~city1 a~street a~post_code1 b~kunnr INTO TABLE i_adrc
  FROM adrc AS a
```

```
  INNER JOIN
  kna1 AS b
  ON a~addrnumber = b~adrnr
  WHERE b~kunnr IN s_kunnr.



*&---------------------------------------------------------------------*
*& i_kna1 Entry                                                        *
*&---------------------------------------------------------------------*

  SELECT kunnr name1 FROM kna1 INTO table i_kna1
       WHERE kunnr IN s_kunnr1.



ENDFORM.                    " logic

*&---------------------------------------------------------------------*
*& Form  modify_table
*&---------------------------------------------------------------------*
*  Description : Modify final table Entry
*----------------------------------------------------------------------*
FORM modify_table.
  IF NOT i_vbak[] IS INITIAL.
  sort i_vbak by vbeln.
    CLEAR wa_vbak.
    LOOP AT i_vbak INTO wa_vbak.


      CLEAR wa_vbpa.
      READ TABLE i_vbpa INTO wa_vbpa WITH KEY vbeln = wa_vbak-vbeln.
      IF sy-subrc = 0.
*       Ship To Party
        wa_vbak-shkunnr = wa_vbpa-kunnr.
```

```
        CLEAR wa_vbpa.
      ENDIF.

      CLEAR wa_xvbpa.
      READ TABLE x_vbpa INTO wa_xvbpa WITH KEY vbeln = wa_vbak-vbeln.
      IF sy-subrc = 0.
*        Partner Function : Code
*        wa_vbak-iparvw = wa_xvbpa-parvw.
         wa_vbak-prkunnr = wa_xvbpa-kunnr.
        CLEAR wa_xvbpa.
      ENDIF.



      CLEAR wa_makt.

      READ TABLE i_makt INTO wa_makt WITH KEY matnr = wa_vbak-matnr.
      IF sy-subrc = 0.
*       Material Description
        wa_vbak-maktx = wa_makt-maktx.
        CLEAR wa_makt.
      ENDIF.

      CLEAR wa_vbrp.
      READ TABLE i_join INTO wa_join WITH KEY vgbel = wa_vbak-vbeln.
      IF sy-subrc = 0.
**       Business Area
        wa_vbak-gsber = wa_join-gsber.
        wa_vbak-ex_vbeln = wa_join-vbeln.
        CLEAR wa_join.
      ENDIF.

      CLEAR wa_j_1iexcdtl.
```

```
      READ TABLE i_j_1iexcdtl INTO wa_j_1iexcdtl WITH KEY matnr = wa_vbak-matnr.
      IF sy-subrc = 0.
**        Chapter ID
        wa_vbak-chapid = wa_j_1iexcdtl-chapid.
        CLEAR wa_j_1iexcdtl.
      ENDIF.




      CLEAR wa_vbkd.
      READ TABLE i_vbkd INTO wa_vbkd WITH KEY vbeln = wa_vbak-vbeln.
      IF sy-subrc = 0.
*       Purchase Order Number
        wa_vbak-bstkd = wa_vbkd-bstkd.
*       Purchase Order Date
        wa_vbak-bstdk = wa_vbkd-bstdk.
*       Inco terms 1
        wa_vbak-inco1 = wa_vbkd-inco1.
*       Inco terms 2
        wa_vbak-inco2 = wa_vbkd-inco2.
*       Payment Terms
        wa_vbak-zterm = wa_vbkd-zterm.

        CLEAR wa_vbkd.
      ENDIF.




      CLEAR wa_xlikp.

      READ TABLE x_likp INTO wa_xlikp WITH KEY matnr = wa_vbak-matnr.
      IF sy-subrc = 0.
```

```
*      Delivery Number
         wa_vbak-dvbeln = wa_xlikp-vbeln.
*        Delivery Date
         wa_vbak-lfdat = wa_xlikp-lfdat.
*        Delivery Type
         wa_vbak-lfart = wa_xlikp-lfart.
*        Delivery Quantity
         wa_vbak-lfimg = wa_xlikp-lfimg.
         CLEAR wa_xlikp.
       ENDIF.




       CLEAR wa_xvbfa.
       READ TABLE x_vbfa INTO wa_xvbfa WITH KEY vbelv = wa_vbak-vbeln.
       IF sy-subrc = 0.
*        Sales Order Number
         wa_vbak-vbeln_m = wa_xvbfa-vbeln.
         CLEAR wa_xvbfa.
       ENDIF.


       CLEAR wa_vbrk.


       READ TABLE i_vbrk INTO wa_vbrk WITH KEY vbeln = wa_vbak-vbeln_m. "vkorg = wa_vbak-
vkorg.
       IF sy-subrc = 0.
*        Invoice Number
         wa_vbak-ivbeln  = wa_vbrk-vbeln.
*        Invoice Date
         wa_vbak-fkdat  = wa_vbrk-fkdat.
*        Invoice Type
         wa_vbak-fkart  = wa_vbrk-fkart.
```

```
        wa_vbak-knumv  = wa_vbrk-knumv.
        CLEAR wa_vbrk.
      ENDIF.




      CLEAR wa_adrc.

      READ TABLE i_adrc INTO wa_adrc WITH KEY kunnr = wa_vbak-kunnr.
      IF sy-subrc = 0.
*       1st Name of Sold-to-Party
        wa_vbak-name1 = wa_adrc-name1.
*       2nd Name of Sold-to-Party
        wa_vbak-name2 = wa_adrc-name2.
*       City of Sold-to-Party
        wa_vbak-city1 = wa_adrc-city1.
        wa_vbak-street = wa_adrc-street.
*       PIN code of Sold-to-Party
        wa_vbak-post_code1 = wa_adrc-post_code1.

        CLEAR wa_adrc.
      ENDIF.

      CLEAR wa_adrc.

      CLEAR wa_xvbak.
      READ TABLE x_vbak INTO wa_xvbak WITH KEY vbeln = wa_vbak-vbeln.
      IF wa_xvbak-vbtyp = c_cc.
*    Credit No.
        wa_vbak-c_vbeln = wa_xvbak-vbeln.
```

```
      ENDIF.


      IF wa_xvbak-vbtyp = c_cd.
*    Debit No.
        wa_vbak-audat = wa_xvbak-audat.
        wa_vbak-d_vbeln = wa_xvbak-vbeln.
        CLEAR wa_xvbak.
      ENDIF.



      MODIFY i_vbak FROM wa_vbak.
      CLEAR : wa_vbak, wa_xlikp.


    ENDLOOP.


  ENDIF.


loop at i_vbak..
  SELECT * from zsdtab into corresponding fields of i_sdtab
  where VBELN = i_vbak-ivbeln.
  append i_sdtab.
  clear i_sdtab.
  ENDSELECT.
endloop.


*loop at i_vbak.
*  SELECT vbeln form from zsdtab into table i_sdtab "corresponding fields of i_sdtab
*  where VBELN = i_vbak-ivbeln.
*
**  append i_sdtab.
**  clear i_sdtab.
**  ENDSELECT.
```

```
*endloop.



IF NOT i_vbak[] IS INITIAL.
    sort i_j_1iexchdr by rdoc.
    sort i_vbak by ex_vbeln.
    CLEAR wa_vbak.
    LOOP AT i_vbak INTO wa_vbak.


      CLEAR wa_vbkd.
      READ TABLE i_vbkd INTO wa_vbkd WITH KEY vbeln = wa_vbak-vbeln.
      IF wa_vbak-c_vbeln <> ' '.
*       Credit Note Date
        wa_vbak-crfkdat = wa_vbkd-fkdat.
*       Credit Note Amount
        wa_vbak-netwr = wa_vbak-netwr.
        CLEAR wa_vbkd.
      ENDIF.
      IF wa_vbak-c_vbeln = ' '.
*       Credit Note Amount
      wa_vbak-netwr = ' '.
        CLEAR wa_vbkd.
      ENDIF.


      CLEAR wa_j_1iexchdr.
      if wa_vbak-ex_vbeln <> ' '.
      READ TABLE i_j_1iexchdr INTO wa_j_1iexchdr WITH KEY rdoc = wa_vbak-ex_vbeln.
      IF sy-subrc = 0.
      if wa_vbak-vbtyp = c_c.
*        Official Excise Document Number
        wa_vbak-exnum = wa_j_1iexchdr-exnum.
*        Excise Document Date
```

```
        wa_vbak-exdat = wa_j_1iexchdr-exdat.
*        Excise Group
        wa_vbak-srgrp = wa_j_1iexchdr-srgrp.
*        Excise Document Series Group
        wa_vbak-exgrp = wa_j_1iexchdr-exgrp.
*        Chapter ID
*        wa_vbak-chapid = wa_j_1iexchdr-chapid.
        CLEAR wa_j_1iexchdr.
        endif.
      ENDIF.
     endif.

     CLEAR wa_sdtab.
     READ TABLE i_sdtab INTO wa_sdtab WITH KEY vbeln = wa_vbak-ivbeln.
     IF SY-SUBRC = 0.
     wa_vbak-form = wa_sdtab-form.
     ENDIF.
     CLEAR wa_sdtab.




      CLEAR wa_kna1.
      READ TABLE i_kna1 INTO wa_kna1 WITH KEY kunnr = wa_vbak-prkunnr.
       IF SY-SUBRC = 0.
*    Sales Person.
        wa_vbak-name_sales = wa_kna1-name1.
        CLEAR wa_kna1.
      ENDIF.
      MODIFY i_vbak FROM wa_vbak.
      CLEAR : wa_vbak, wa_xlikp.

    ENDLOOP.
```

```
  ENDIF.



ENDFORM.                    " modify_table

*&---------------------------------------------------------------------*
*& Form  display
*&---------------------------------------------------------------------*
*  Description : Display as output
*----------------------------------------------------------------------*
FORM display .



  IF  i_vbak[] IS INITIAL.
    MESSAGE i002(z2) WITH 'No Data Found' .                "#EC NOTEXT
  ELSE.

FORMAT COLOR 1 ON.
write :/001(45) sy-uline.
  write:/(20) ' Company Code', sy-vline, (20) s_bukrs+3(4), sy-vline.
write :/001(45) sy-uline.
  write:/(20) ' Sales Person', sy-vline, (20) i_vbak-name_sales, sy-vline.
write :/001(45) sy-uline.
  write:/(20) ' Business Area', sy-vline, (20) s_gsber+3(4), sy-vline.
write :/001(45) sy-uline.
  write:/(20) ' Sales Organization', sy-vline, (20) i_vbak-vkorg, sy-vline.
write :/001(45) sy-uline.

  FORMAT COLOR OFF.
```

```
  sort i_vbak by kunnr.
  DATA l_vbeln LIKE vbak-vbeln.


*  IF  i_vbak[] IS INITIAL.
*    MESSAGE i002(z2) WITH 'No Data Found' .                 "#EC NOTEXT
*  ELSE.


*    CLEAR wa_fin.
    LOOP AT i_vbak.                                          "#EC *
    Move-corresponding i_vbak to wa_fin.
    at new kunnr.
    FORMAT COLOR 7 ON.
     WRITE :/001(699) sy-uline.


*        write :/ 'SOLD-TO-PARTY :', wa_fin-kunnr, 'SOLD-TO-PARTY NAME :', i_vbak-name1,
'SOLD-TO-PARTY ADDRESS :', i_vbak-city1,
*             'Inco Term2 :', i_vbak-inco2,(501) ' ',sy-vline.
        write :/ 'SOLD-TO-PARTY :', wa_fin-kunnr, 'SOLD-TO-PARTY NAME :', wa_fin-name1,
'SOLD-TO-PARTY ADDRESS :', wa_fin-city1,
             'Inco Term2 :', wa_fin-inco2,(501) ' ',sy-vline.


        WRITE :/001(699) sy-uline.
        FORMAT COLOR OFF.
  FORMAT COLOR 1 ON.
  WRITE :/(6) 'Distr.',sy-vline,(4) 'Div.',sy-vline,"#EC *
          (6) 'Sales',sy-vline, "#EC *
          (10) 'Sales',sy-vline,(10) 'Sales',sy-vline,
          (10) 'Purchase',sy-vline,(10) 'Purchase',          "#EC *
           sy-vline,(5) 'Pymnt',"#EC *
           sy-vline,(4) 'Form',sy-vline,(15) 'Form',sy-vline,(15) 'Material',"#EC *
           sy-vline,(35) 'Material',sy-vline,(15) 'Chapter',sy-vline,"#EC *
```

```
           (5) 'Sales',sy-vline,"#EC *

           (5) 'Plant',sy-vline,            "#EC *

           (10) 'Delivery',sy-vline,(10) 'Delivery',sy-vline,"#EC *

           (10) 'Delivery',sy-vline,(10) 'Invoice',sy-vline,"#EC *

           (10) 'Invoice',sy-vline, "#EC *

           (10) 'Cr Note',sy-vline,"#EC *

           (10) 'Cr Note',sy-vline,(15) 'Cr Note',sy-vline, "#EC *

           (10) 'Dr Note',sy-vline,(10) 'Dr Note',sy-vline, "#EC *

           (12) 'Excise',sy-vline,"#EC *


           (11) 'Price' , sy-vline,(11) 'IN Trade' , sy-vline,(11) 'Cash' , sy-vline,(11)
'Special' , sy-vline,

           (11) 'IN A/R' , sy-vline,

           (11) 'IN A/R' , sy-vline,(11) 'IN A/R' , sy-vline,(11) 'A/R VAT' , sy-vline,(11)
'Local' , sy-vline,(11) 'Commission' , sy-vline,(11) 'Brokerage' , sy-vline

           ,(11) 'Incentive' , sy-vline,(11) 'Indian' , sy-vline,(11) 'DEPB' , sy-
vline,(11) 'TAX-TCS' , sy-vline

           ,(11) 'Repacking' , sy-vline,(11) 'Cartage' , sy-vline,(11) 'A/R Edu.' , sy-
vline,(11) 'Insurance' , sy-vline

           ,(11) 'Frieght' , sy-vline,(11) 'IN 100%' , sy-vline,(11) 'Repacking' , sy-
vline,(11) 'Sea' , sy-vline

           ,(11) 'Air' , sy-vline,(11) 'CHA' , sy-vline.




  WRITE :/(06) 'Chnnl.',sy-vline,(4) ' ',sy-vline,"#EC *

          (6) 'Office',sy-vline, "#EC *

          (10) 'Order No',sy-vline,(10) 'Order Type',sy-vline,

           (10) 'Order No',sy-vline,"#EC *

          (10) 'Ord. Date',sy-vline,"#EC *

          (5) 'Terms',sy-vline,(4) 'Type',sy-vline,(15) 'No.',sy-vline,"#EC *

          (15) 'Number',sy-vline,(35) 'Description',sy-vline,"#EC *

          (15) 'ID',sy-vline,(5) 'Unit',sy-vline,"#EC *

           (5) ' ',sy-vline,"#EC *
```

```
           (10) 'Number',sy-vline,(10) 'Date',sy-vline,"#EC *

           (10) 'Quantity',sy-vline,(10) 'Number',sy-vline,(10) 'Date',sy-vline,"#EC *

           (10) 'Number',sy-vline,(10) 'Date',sy-vline,"#EC *

           (15) 'Amount',sy-vline,(10) 'Number',sy-vline,(10) 'Date',sy-vline,"#EC *

            (12) 'Invoice No.',sy-vline,  "#EC *


           (11) 'Basic' , sy-vline,(11) 'Discount' , sy-vline,(11) 'Discount' , sy-
vline,(11) 'Discount' , sy-vline,

           (11) 'BED' , sy-vline,

           (11) 'CST' , sy-vline,(11) 'LST' , sy-vline,(11) 'Payble' , sy-vline,(11)
'Frieght' , sy-vline,(11) '     %' , sy-vline,(11) '     %' , sy-vline

           ,(11) ' ' , sy-vline,(11) 'Haulage' , sy-vline,(11) 'Recoverable' , sy-
vline,(11) ' ' , sy-vline

           ,(11) 'Charges' , sy-vline,(11) 'Charges' , sy-vline,(11) 'Less' , sy-vline,(11)
' ' , sy-vline

           ,(11) 'Collection' , sy-vline,(11) 'Discount' , sy-vline,(11) 'Insurance' , sy-
vline,(11) 'Frieght' , sy-vline

           ,(11) 'Frieght' , sy-vline,(11) 'Charges' , sy-vline.




  WRITE :/001(699) sy-uline.

     FORMAT COLOR OFF.

    endat.

    FORMAT COLOR 2 ON.

      IF l_vbeln NE i_vbak-vbeln.

        l_vbeln = i_vbak-vbeln.


        WRITE :/(06) i_vbak-vtweg,sy-vline,"#EC *

                (4) i_vbak-spart,sy-vline,"#EC *

                (6) i_vbak-vkbur,sy-vline, "#EC *

                (10) i_vbak-vbeln,sy-vline,(10) i_vbak-auart,sy-vline,"#EC *

                (10) i_vbak-bstkd,sy-vline,(10) i_vbak-bstdk,sy-vline,"#EC *

                (5) i_vbak-zterm,sy-vline,(4) i_vbak-taxk1,sy-vline,(15) i_vbak-form,"#EC *

                 sy-vline,(15) i_vbak-matnr,sy-vline,(35) i_vbak-maktx,"#EC *
```

```
                 sy-vline,(15) i_vbak-chapid,sy-vline,(5) i_vbak-zieme,sy-vline,"#EC *

                (5) i_vbak-werks,sy-vline,"#EC *

                (10) i_vbak-dvbeln,sy-vline,(10) i_vbak-lfdat,sy-vline,"#EC *

                (10) i_vbak-lfimg,sy-vline,(10) i_vbak-ivbeln,sy-vline,"#EC *

                (10) i_vbak-fkdat,sy-vline,   "#EC *

                (10) i_vbak-C_VBELN,sy-vline,"#EC *

                (10) i_vbak-crfkdat,sy-vline,(15) i_vbak-netwr,sy-vline,"#EC *

                (10) i_vbak-D_VBELN,sy-vline,(10) i_vbak-audat,sy-vline,"#EC *

                (12) i_vbak-exnum,sy-vline,


                 (11) i_vbak-zpr0 , sy-vline,(11) i_vbak-jtrd , sy-vline,(11) i_vbak-ZJDI ,
sy-vline,(11) i_vbak-zjd2 , sy-vline,

                 (11) i_vbak-jexp , sy-vline,(11) i_vbak-jcst , sy-vline,(11) i_vbak-jlst ,
sy-vline,(11) i_vbak-jin6 , sy-vline,

                 (11) i_vbak-zfro , sy-vline,(11) i_vbak-zdcp , sy-vline,

                 (11) i_vbak-zdbp , sy-vline,(11) i_vbak-zjd3 , sy-vline,(11) i_vbak-zdfr ,
sy-vline,(11) i_vbak-zder , sy-vline,

                 (11) i_vbak-ztcs , sy-vline,(11) i_vbak-zpac , sy-vline,(11) i_vbak-zcrt ,
sy-vline,(11) i_vbak-jcex , sy-vline,

                 (11) i_vbak-zins , sy-vline

                 ,(11) i_vbak-zfre , sy-vline,(11) i_vbak-jfcg , sy-vline,(11) i_vbak-zfr1 ,
sy-vline,(11) i_vbak-zsfr , sy-vline

                 ,(11) i_vbak-zair , sy-vline,(11) i_vbak-zcha , sy-vline.


      ELSE.

        WRITE :/(06) i_vbak-vtweg,sy-vline,"#EC *

                (4) i_vbak-spart,sy-vline,"#EC *

                (6) i_vbak-vkbur,sy-vline,"#EC *

                (10) i_vbak-vbeln,sy-vline,(10) i_vbak-auart,sy-vline,"#EC *

                (10) i_vbak-bstkd,sy-vline,(10) i_vbak-bstdk,sy-vline,"#EC *

                (5) i_vbak-zterm,sy-vline,(4) i_vbak-taxk1,sy-vline,(15) i_vbak-form,"#EC *

                sy-vline,(15) i_vbak-matnr,sy-vline,(35) i_vbak-maktx,"#EC *

                sy-vline,(15) i_vbak-chapid,sy-vline,(5) i_vbak-zieme,sy-vline,"#EC *

                (5) i_vbak-werks,sy-vline,"#EC *
```

```
                (10) i_vbak-dvbeln,sy-vline,(10) i_vbak-lfdat,sy-vline,"#EC *

                (10) i_vbak-lfimg,sy-vline,(10) i_vbak-ivbeln,sy-vline,"#EC *

                (10) i_vbak-fkdat,sy-vline, "#EC *

                (10) i_vbak-C_VBELN,sy-vline,"#EC *

                (10) i_vbak-crfkdat,sy-vline,(15) i_vbak-netwr,sy-vline,"#EC *

                (10) i_vbak-D_VBELN,sy-vline,(10) i_vbak-audat,sy-vline,"#EC *

                (12) i_vbak-exnum,sy-vline,


                 (11) i_vbak-zpr0 , sy-vline,(11) i_vbak-jtrd , sy-vline,(11) i_vbak-ZJDI ,
sy-vline,(11) i_vbak-zjd2 , sy-vline,

                 (11) i_vbak-jexp , sy-vline,(11) i_vbak-jcst , sy-vline,(11) i_vbak-jlst ,
sy-vline,(11) i_vbak-jin6 , sy-vline,

                 (11) i_vbak-zfro , sy-vline,(11) i_vbak-zdcp , sy-vline,

                 (11) i_vbak-zdbp , sy-vline,(11) i_vbak-zjd3 , sy-vline,(11) i_vbak-zdfr ,
sy-vline,(11) i_vbak-zder , sy-vline,

                 (11) i_vbak-ztcs , sy-vline,(11) i_vbak-zpac , sy-vline,(11) i_vbak-zcrt ,
sy-vline,(11) i_vbak-jcex , sy-vline,

                 (11) i_vbak-zins , sy-vline

                 ,(11) i_vbak-zfre , sy-vline,(11) i_vbak-jfcg , sy-vline,(11) i_vbak-zfr1 ,
sy-vline,(11) i_vbak-zsfr , sy-vline

                 ,(11) i_vbak-zair , sy-vline,(11) i_vbak-zcha , sy-vline.




    ENDIF.


    FORMAT COLOR OFF.

*    WRITE :/001(786) sy-uline.


    AT END OF KUNNR.

*     COLLECT i_vbak.

    COLLECT wa_fin into i_vbak.

    FORMAT COLOR 1 ON.

    WRITE :/001(699) sy-uline.
```

```
    WRITE :/(223) ' ',sy-vline,(10)  wa_fin-lfimg, sy-vline, (49) ' ', sy-vline, (15)
wa_fin-netwr, sy-vline, (38) ' ',sy-vline, (11) wa_fin-zpr0, sy-vline.

    WRITE :/001(699) sy-uline.

    FORMAT COLOR OFF.

    ENDAT.


*    AT LAST.

**    Collect i_vbak.

*    COLLECT wa_fin into i_vbak.

*    FORMAT COLOR 3 ON.

*    WRITE :/001(699) sy-uline.

*    WRITE :/(223) ' ',sy-vline,(10)  wa_fin-lfimg, sy-vline, (49) ' ', sy-vline, (15)
wa_fin-netwr, sy-vline, (38) ' ',sy-vline, (11) wa_fin-zpr0, sy-vline.

*    WRITE :/001(699) sy-uline.

*    FORMAT COLOR OFF.

*    ENDAT.

    ENDLOOP.


    FORMAT COLOR 3 ON.

    WRITE :/001(699) sy-uline.

    WRITE :/(223) ' ',sy-vline,(10) total, sy-vline, (49) ' ', sy-vline, (15) total2, sy-
vline, (38) ' ',sy-vline, (11) total3, sy-vline.

    WRITE :/001(699) sy-uline.

    FORMAT COLOR OFF.


  ENDIF.

ENDFORM.                    " display


*&---------------------------------------------------------------------*

*&      Form  pricing_cond

*&---------------------------------------------------------------------*

*       text

*----------------------------------------------------------------------*
```

```
*  -->  p1        text
*  <--  p2        text
*---------------------------------------------------------------------*
form pricing_cond .



loop at i_vbak.
if s_gsber NE ' '.
delete i_vbak where gsber NE s_gsber+3(4).
endif.
if
S_kunnr1 <> ' '.
delete i_vbak where name_sales = ' '.
endif.
endloop.

SELECT knumv KSCHL KBETR FROM KONV INTO table CONDITION for all entries in i_vbak
where knumv = i_vbak-knumv.



IF NOT i_vbak[] IS INITIAL.
    CLEAR wa_vbak.
    LOOP AT i_vbak INTO wa_vbak.
    CLEAR wa_cond.
    LOOP AT CONDITION INTO wa_cond.

    IF wa_cond-knumv = wa_vbak-knumv.

      IF wa_cond-KSCHL = 'ZPAC'.
      wa_vbak-ZPAC = wa_cond-KBETR.
      ENDIF.
```

```
IF wa_cond-KSCHL = 'ZFR1'.
wa_vbak-ZFR1 = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'ZPR0'.
wa_vbak-ZPR0 = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'ZCRT'.
wa_vbak-ZCRT = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'JEXP'.
wa_vbak-JEXP = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'ZJD3'.
wa_vbak-ZJD3 = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'JCEX'.
wa_vbak-JCEX = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'ZSFR'.
wa_vbak-ZSFR = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'ZAIR'.
wa_vbak-ZAIR = wa_cond-KBETR.
ENDIF.
```

```
IF wa_cond-KSCHL = 'ZINS'.
wa_vbak-ZINS = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'ZCHA'.
wa_vbak-ZCHA = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'JTRD'.
wa_vbak-JTRD = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'JCST'.
wa_vbak-JCST = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'JLST'.
wa_vbak-JLST = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'JIN6'.
wa_vbak-JIN6 = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'JFCG'.
wa_vbak-JFCG = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'ZJD2'.
wa_vbak-ZJD2 = wa_cond-KBETR.
ENDIF.
```

```
IF wa_cond-KSCHL = 'ZJDI'.
wa_vbak-ZJDI = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'ZFRE'.
wa_vbak-ZFRE = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'B001'.
wa_vbak-B001 = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'JEXQ'.
wa_vbak-JEXQ = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'JEXT'.
wa_vbak-JEXT = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'JCES'.
wa_vbak-JCES = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'JIN7'.
wa_vbak-JIN7 = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'JIN8'.
wa_vbak-JIN8 = wa_cond-KBETR.
ENDIF.
```

```
IF wa_cond-KSCHL = 'ZDCP'.
wa_vbak-ZDCP = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'ZDBP'.
wa_vbak-ZDBP = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'ZDFR'.
wa_vbak-ZDFR = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'ZDER'.
wa_vbak-ZDER = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'ZTCS'.
wa_vbak-ZTCS = wa_cond-KBETR.
ENDIF.

IF wa_cond-KSCHL = 'ZFRO'.
wa_vbak-ZFRO = wa_cond-KBETR.
ENDIF.

CLEAR wa_COND.

MODIFY i_vbak FROM wa_vbak.

ENDIF.
endloop.
CLEAR : wa_vbak.
```

```
      CLEAR wa_COND.
    ENDLOOP.

  CLEAR : wa_vbak.
  ENDIF.

loop at i_vbak.
  total = total + i_vbak-lfimg.
  total2 = total2 + i_vbak-netwr.
  total3 = total3 + i_vbak-zpr0.
endloop.

endform.                    " pricing_cond



*&---------------------------------------------------------------------
-------------*
*& Date        Modif.ID   Modification Details
*&---------------------------------------------------------------------
-------------*
*& 02/11/2006  G001       Provided Additional Button on ALV List for
Stock Overview
*&                        display, By clicking on this button user can
access MB52
*&                        transaction. It will display stock for all
the materials
*&                        displayed on the screen
*&---------------------------------------------------------------------
-------------*

REPORT ZSDR0002.
*---------------------------------------------------------------------
-
* Tables
*---------------------------------------------------------------------
-
TABLES : VBAK, VBAP, LIKP, LIPS, VBUP, VBEP.

*---------------------------------------------------------------------
-
* Internal Tables
*---------------------------------------------------------------------
-
DATA : BEGIN OF FINAL OCCURS 0,
       VBELN LIKE VBAK-VBELN,    "SALES ORDER
       POSNR LIKE VBAP-POSNR,    "ITEM
```

```
        AUART LIKE VBAK-AUART,    "ORDER TYPE
        AUDAT LIKE VBAK-AUDAT,    "ORDER DATE
        BSTNK LIKE VBAK-BSTNK,    "CUSTOMER PO NO.
        BSTDK LIKE VBAK-BSTDK,    "CUSTOMER PO DATE
        WAERK LIKE VBAK-WAERK,    "Currency
        LIFSK LIKE VBAK-LIFSK,    "Delivery Block
        ZMENG LIKE VBAP-ZMENG,    "TARGET QTY.
        KWMENG LIKE VBAP-KWMENG,  "SO QTY
        OPENQTY LIKE VBAP-KWMENG, " OPEN QTY
        LFIMG LIKE LIPS-LFIMG,    "DELIVERY QTY
        NETPR LIKE VBAP-NETPR,    "UNIT PRICE
        NETWR LIKE VBAP-NETWR,    "NET VALUE
        BALWR LIKE VBAP-NETWR,    "NET VALUE FOR BALANCE QTY.
        BALLOC LIKE VBAP-NETWR,   "BALANCE VALUE IN LOCAL CURRENCY
        DLVQTY LIKE LIPS-LFIMG,   "DELIVERED QTY.
        KURSK LIKE VBKD-KURSK,    "EXCHANGE RATE
        LOCVAL LIKE VBAP-NETWR,   "AMT. IN LOCAL CURRENCY
        KUNNR LIKE VBAK-KUNNR,    "CUSTOMER
        VKORG LIKE VBAK-VKORG,    "SALES ORG.
        VKBUR LIKE VBAK-VKBUR,    "SALES OFF.
        VDATU LIKE VBAK-VDATU,    "Req.Del.Date
        MATNR LIKE MARA-MATNR,    "MATERIAL NO
        ARKTX LIKE VBAP-ARKTX,    "MATERIAL DESC
        WERKS LIKE VBAP-WERKS,    "PLANT
        LFSTA LIKE VBUP-LFSTA,    "Item Delivery Status
        GBSTA LIKE VBUP-GBSTA,    "Overall Status
*        PONO  LIKE VBFA-VBELN,   "PO NO.
*        POITEM LIKE VBFA-POSNN,  "PO ITEM
        NAME1 LIKE KNA1-NAME1,    "CUSTOMER NAME
        VBTYP LIKE VBAK-VBTYP,    "Doc. Category
        ETENR LIKE VBEP-ETENR,    "Schd.Line Item No.
        EDATU LIKE VBEP-EDATU,    "Schedule line date
        WMENG LIKE VBEP-WMENG,    "Schedule qty.
        BMENG LIKE VBEP-BMENG,    "Confirmed Qty.
        CRSTAT LIKE DD07V-DDTEXT, "Credit check status
        VTEXT  LIKE TVLST-VTEXT,  "Delivery block
        CITY1 LIKE ADRC-CITY1,    "CITY
        CARRIER(30) TYPE C,       "CARRIER
        END OF FINAL.

DATA: FINAL1 LIKE FINAL OCCURS 0 WITH HEADER LINE.
DATA: MCTRAN LIKE TCURF-FFACT.

DATA: BEGIN OF I_TVKBZ OCCURS 0,
      VKORG LIKE TVKBZ-VKORG,
      VKBUR LIKE TVKBZ-VKBUR,
      END OF I_TVKBZ.

DATA: BEGIN OF I_VBUK OCCURS 0,
      VBELN LIKE VBUK-VBELN,
      CMGST LIKE VBUK-CMGST,
      CRSTAT LIKE DD07V-DDTEXT, "Credit check status
      END OF I_VBUK.

DATA: MUGRP LIKE USGRP_USER-USERGROUP.
DATA: MUNAME LIKE SY-UNAME,
      MMSGTX1(50) TYPE C.
```

```
*---------------------------------------------------------------------
-
* Data declarations
*---------------------------------------------------------------------
-
TYPE-POOLS: SLIS.

CONSTANTS:
GC_FORMNAME_TOP_OF_PAGE TYPE SLIS_FORMNAME VALUE 'TOP_OF_PAGE'.
DATA: GT_FIELDCAT TYPE SLIS_T_FIELDCAT_ALV,
      GS_LAYOUT   TYPE SLIS_LAYOUT_ALV,
      GS_PRINT    TYPE SLIS_PRINT_ALV,
      GT_SORT     TYPE SLIS_T_SORTINFO_ALV,
      GT_SP_GROUP TYPE SLIS_T_SP_GROUP_ALV,
      GT_EVENTS   TYPE SLIS_T_EVENT WITH HEADER LINE.


DATA: G_REPID LIKE SY-REPID.
DATA: GT_LIST_TOP_OF_PAGE TYPE SLIS_T_LISTHEADER.

DATA:       G_BOXNAM TYPE SLIS_FIELDNAME VALUE  'BOX',
            P_F2CODE LIKE SY-UCOMM       VALUE  '&ETA',
            P_LIGNAM TYPE SLIS_FIELDNAME VALUE  'LIGHTS',
            G_SAVE(1) TYPE C,
            G_DEFAULT(1) TYPE C,
            G_EXIT(1) TYPE C,
            GX_VARIANT LIKE DISVARIANT,
            G_VARIANT LIKE DISVARIANT.

*---------------------------------------------------------------------
-
* Parameter /  Selection - screens
*---------------------------------------------------------------------
-
SELECTION-SCREEN BEGIN OF BLOCK B1 WITH FRAME TITLE TEXT-001.

SELECT-OPTIONS : S_VKORG FOR VBAK-VKORG OBLIGATORY,
                 S_VKBUR FOR VBAK-VKBUR,
                 S_WERKS FOR VBAP-WERKS OBLIGATORY,
                 S_VBELN FOR VBAP-VBELN,
                 S_KUNNR FOR VBAK-KUNNR,
                 S_MATNR FOR VBAP-MATNR,
                 S_AUART FOR VBAK-AUART,
                 S_EDATU FOR VBEP-EDATU,
                 S_MATKL FOR VBAP-MATKL,
                 S_STAT  FOR VBUP-LFSTA.
*PARAMETERS : C_SUMREP AS CHECKBOX.

SELECTION-SCREEN END OF BLOCK B1.
*---------------------------------------------------------------------
-
* Initialization
*---------------------------------------------------------------------
-
INITIALIZATION.
```

```
*---------------------------------------------------------------------
-
* Validation Section
*---------------------------------------------------------------------
-
AT SELECTION-SCREEN.

*---------------------------------------------------------------------
-
* START-OF-SELECTION
*---------------------------------------------------------------------
-
START-OF-SELECTION.

  PERFORM DATA_COLLECTION.
*  IF C_SUMREP = 'X'.
*    PERFORM SUMMARISE_DATA.
*    ENDIF.
  PERFORM DATA_FORMAT.

END-OF-SELECTION.

  REFRESH S_VKBUR.
  CLEAR S_VKBUR.

*---------------------------------------------------------------------
-
* End of selection
*---------------------------------------------------------------------
-
*&---------------------------------------------------------------------
*
*&      Form  DATA_COLLECTION
*&---------------------------------------------------------------------
*
FORM DATA_COLLECTION.

DATA: BEGIN OF I_SODATA OCCURS 0,
      VBELN  LIKE VBAP-VBELN,
      POSNR  LIKE VBAP-POSNR,
      KWMENG LIKE VBAP-KWMENG,
      NETWR  LIKE VBAP-NETWR,
      END OF I_SODATA.

DATA: BEGIN OF I_LIPS OCCURS 0,
      VBELN  LIKE LIPS-VBELN,
      POSNR  LIKE LIPS-POSNR,
      WERKS  LIKE LIPS-WERKS,
      LFIMG  LIKE LIPS-LFIMG,
      VGBEL  LIKE LIPS-VGBEL,
      VGPOS  LIKE LIPS-VGPOS,
      END OF I_LIPS.

RANGES: R_VBTYP FOR VBAK-VBTYP.
RANGES: R_VBTYP_N FOR VBFA-VBTYP_N.

R_VBTYP-SIGN   = 'E'.
R_VBTYP-OPTION = 'EQ'.
```

```
R_VBTYP-LOW    = 'B'.
APPEND R_VBTYP.

R_VBTYP_N-SIGN   = 'I'.
R_VBTYP_N-OPTION = 'EQ'.
R_VBTYP_N-LOW    = 'V'.
APPEND R_VBTYP_N.

SELECT VBAK~AUART VBAK~AUDAT VBAK~KUNNR VBAK~VKORG VBAK~VKBUR
VBAK~BSTDK VBAK~BSTNK VBAK~VDATU VBAK~LIFSK
       VBAK~WAERK VBAP~ZMENG VBAP~NETWR VBAP~NETPR VBAP~VBELN
VBAP~POSNR VBAP~KWMENG VBAP~MATKL
       VBAP~MATNR VBAP~WERKS VBAP~ARKTX VBUP~LFSTA VBKD~KURSK
VBEP~ETENR VBEP~EDATU VBEP~WMENG VBEP~BMENG
       INTO CORRESPONDING FIELDS OF TABLE FINAL
       FROM VBAK
       JOIN VBAP ON VBAP~VBELN  = VBAK~VBELN
       JOIN VBUP ON  VBUP~VBELN = VBAP~VBELN
       AND           VBUP~POSNR = VBAP~POSNR
       JOIN VBKD ON  VBKD~VBELN = VBAK~VBELN
       AND           VBKD~POSNR = '000000'
       JOIN VBEP ON VBEP~VBELN  = VBAP~VBELN
       AND          VBEP~POSNR  = VBAP~POSNR
       WHERE VBAK~VBELN IN S_VBELN
       AND   ( VBAK~VBTYP EQ 'C' OR VBAK~VBTYP = 'L' OR VBAK~VBTYP =
'G' )
       AND   VBAK~AUART IN S_AUART
       AND   VBAK~VBTYP IN R_VBTYP
       AND   VBAK~VKORG IN S_VKORG
       AND   VBAK~VKBUR IN S_VKBUR
       AND   VBAK~KUNNR IN S_KUNNR
       AND   VBAP~WERKS IN S_WERKS
       AND   VBAP~MATNR IN S_MATNR
       AND   VBAP~MATKL IN S_MATKL
       AND   VBUP~LFSTA IN S_STAT
       AND   VBEP~EDATU IN S_EDATU.


*SORT FINAL BY VBTYP AUART.
*DELETE FINAL WHERE VBTYP = 'G' AND AUART <> 'ZSM1'.
*
*SORT FINAL BY VBELN POSNR.
*SELECT VBELN POSNR WERKS LFIMG VGBEL VGPOS
*       INTO CORRESPONDING FIELDS OF TABLE I_LIPS
*       FROM LIPS
*       FOR ALL ENTRIES IN FINAL
*       WHERE VGBEL = FINAL-VBELN
*       AND   VGPOS = FINAL-POSNR.
*
*
*LOOP AT FINAL.
*  LOOP AT I_LIPS WHERE VGBEL = FINAL-VBELN
*                 AND   VGPOS = FINAL-POSNR.
*    FINAL-LFIMG = FINAL-LFIMG + I_LIPS-LFIMG.
*  ENDLOOP.
*  MODIFY FINAL.
*ENDLOOP.
```

```
LOOP AT FINAL.
     CALL FUNCTION 'ISU_SD_DELIVERY_TO_SCHDL_GET'
       EXPORTING
         in_vbeln           = FINAL-VBELN
         in_posnr           = FINAL-POSNR
         in_etenr           = FINAL-ETENR
      IMPORTING
         OUT_QUANTITY       = FINAL-DLVQTY.
      MODIFY FINAL.
      ENDLOOP.

  IF FINAL[] IS INITIAL.
    EXIT.
  ENDIF.

  LOOP AT FINAL.
    FINAL-LOCVAL = FINAL-NETWR * FINAL-KURSK.
    IF FINAL-KWMENG = 0.
      FINAL-KWMENG = FINAL-ZMENG.
    ENDIF.
    FINAL-OPENQTY = FINAL-KWMENG.
    MODIFY FINAL.
  ENDLOOP.

* TO FIND THE OPEN ORDER QUANTITY
DATA: MSONO LIKE VBAK-VBELN, MSOITEM LIKE VBAP-POSNR, MDLVQTY LIKE
LIPS-LFIMG, MBALQTY LIKE VBAP-KWMENG.
LOOP AT FINAL.
*    IF MSONO <> FINAL-VBELN.
*       MDLVQTY = FINAL-DLVQTY.
*       MBALQTY = FINAL-KWMENG.
*       ENDIF.
*    IF MSONO = FINAL-VBELN AND MSOITEM <> FINAL-POSNR.
*       MDLVQTY = FINAL-DLVQTY.
*       MBALQTY = FINAL-KWMENG.
*       ENDIF.

*    FINAL-OPENQTY = MBALQTY - FINAL-DLVQTY.  "FINAL-LFIMG. "PARTIALLY
DELIVERED.
    FINAL-OPENQTY = FINAL-BMENG - FINAL-DLVQTY.
    FINAL-BALWR = FINAL-NETPR * FINAL-OPENQTY.
    FINAL-BALLOC  = FINAL-BALWR * FINAL-KURSK.
*    IF FINAL-MATNR(1) = '0'.
*       PACK FINAL-MATNR TO FINAL-MATNR.
*       ENDIF.
*    MDLVQTY = MDLVQTY + FINAL-DLVQTY.
*    MSONO = FINAL-VBELN.
*    MSOITEM = FINAL-POSNR.
*    MBALQTY = MBALQTY - FINAL-DLVQTY.
    MODIFY FINAL.
    ENDLOOP.

SORT FINAL BY KUNNR.
FINAL1[] = FINAL[].
DELETE ADJACENT DUPLICATES FROM FINAL1 COMPARING KUNNR.
IF NOT FINAL1[] IS INITIAL.
   LOOP AT FINAL1.
```

```
     SELECT SINGLE KNA1~NAME1 ADRC~CITY1 INTO (FINAL1-NAME1, FINAL1-
CITY1)
        FROM KNA1
        JOIN ADRC ON ADRC~ADDRNUMBER = KNA1~ADRNR
        WHERE KNA1~KUNNR = FINAL1-KUNNR
        AND   KNA1~SPRAS = 'EN'.
     MODIFY FINAL1.
     ENDLOOP.

   LOOP AT FINAL1.
     LOOP AT FINAL WHERE KUNNR = FINAL1-KUNNR.
       FINAL-NAME1 = FINAL1-NAME1.
       FINAL-CITY1 = FINAL1-CITY1.
*       IF FINAL-KUNNR(1) = '0'.
*          PACK FINAL-KUNNR TO FINAL-KUNNR.
*          ENDIF.
       MODIFY FINAL.
       ENDLOOP.
     ENDLOOP.
  ENDIF.

SORT FINAL BY WAERK.
FINAL1[] = FINAL[].
DELETE ADJACENT DUPLICATES FROM FINAL1 COMPARING WAERK.
DELETE FINAL1 WHERE WAERK = 'INR'.

LOOP AT FINAL1.
  MCTRAN = 0.
  SELECT SINGLE FFACT INTO MCTRAN
         FROM TCURF
         WHERE KURST = 'M'
         AND   FCURR = FINAL1-WAERK
         AND   TCURR = 'INR'.

  LOOP AT FINAL WHERE VBELN = FINAL1-VBELN.
      FINAL-LOCVAL = FINAL-NETWR * FINAL-KURSK.
      FINAL-BALLOC = FINAL-BALWR * FINAL-KURSK.
      FINAL-NETWR = FINAL-NETWR * MCTRAN.
      FINAL-NETPR = FINAL-NETPR * MCTRAN.
      FINAL-BALWR = FINAL-BALWR * MCTRAN.
      MODIFY FINAL.
      ENDLOOP.
  ENDLOOP.

LOOP AT FINAL WHERE LFSTA = 'C'.
    FINAL-OPENQTY = 0.
    FINAL-BALWR  = 0.
    FINAL-BALLOC = 0.
    MODIFY FINAL.
    ENDLOOP.

PERFORM GET_CREDIT_CHECK.
PERFORM GET_DLV_BLOCK.
PERFORM GET_CARRIER.


ENDFORM.                    " DATA_COLLECTION
```

```
*&---------------------------------------------------------------------*
*
*&      Form  DATA_FORMAT
*&---------------------------------------------------------------------*
*
FORM DATA_FORMAT.
  G_REPID = SY-REPID.

  PERFORM E01_FIELDCAT_INIT  USING GT_FIELDCAT[].
  PERFORM E04_COMMENT_BUILD USING GT_LIST_TOP_OF_PAGE[].

  G_SAVE = 'A'.

  CLEAR GT_EVENTS.
  REFRESH GT_EVENTS.
  GT_EVENTS-NAME = 'TOP_OF_PAGE'.
  GT_EVENTS-FORM = 'TOP_OF_PAGE'.
  APPEND GT_EVENTS.

  SORT FINAL BY WERKS AUART VBELN POSNR.
  PERFORM DISPLAY_REPORT TABLES FINAL.



ENDFORM.                    " DATA_FORMAT
*&---------------------------------------------------------------------*
*
*&      Form  e01_fieldcat_init
*&---------------------------------------------------------------------*
*
*       text
*----------------------------------------------------------------------*
*
*      -->P_GT_FIELDCAT[]  text
*----------------------------------------------------------------------*
*
FORM E01_FIELDCAT_INIT USING E01_LT_FIELDCAT TYPE SLIS_T_FIELDCAT_ALV.

  DATA: LS_FIELDCAT TYPE SLIS_FIELDCAT_ALV.
  REFRESH E01_LT_FIELDCAT.

* SO TYPE
  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'AUART'.
  LS_FIELDCAT-COL_POS      = 1.
  LS_FIELDCAT-SELTEXT_L    = 'Type'.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

* Sales Order Date
  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'AUDAT'.
  LS_FIELDCAT-COL_POS      = 2.
  LS_FIELDCAT-SELTEXT_L    = 'Order Date'.
  LS_FIELDCAT-OUTPUTLEN    = 10.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

* Customer PO No.
  CLEAR LS_FIELDCAT.
```

```
  LS_FIELDCAT-FIELDNAME    = 'BSTNK'.
  LS_FIELDCAT-COL_POS      = 3.
  LS_FIELDCAT-SELTEXT_L    = 'Cust.PO No.'.
  LS_FIELDCAT-OUTPUTLEN    = 20.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

* SO NUMBER
  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'VBELN'.
  LS_FIELDCAT-COL_POS      = 4.
  LS_FIELDCAT-HOTSPOT = 'X'.
  LS_FIELDCAT-SELTEXT_L    = 'S.O. No'.
  LS_FIELDCAT-OUTPUTLEN    = 12.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

*  IF C_SUMREP = ''.
* SO ITEM
    CLEAR LS_FIELDCAT.
    LS_FIELDCAT-FIELDNAME    = 'POSNR'.
    LS_FIELDCAT-COL_POS      = 5.
    LS_FIELDCAT-SELTEXT_L    = 'Item No.'.
    APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

* MATERIAL NO.
    CLEAR LS_FIELDCAT.
    LS_FIELDCAT-FIELDNAME    = 'MATNR'.
    LS_FIELDCAT-COL_POS      = 6.
    LS_FIELDCAT-SELTEXT_L    = 'Material No.'.
    LS_FIELDCAT-OUTPUTLEN    = 15.
    APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

* MATERIAL DESC.
    CLEAR LS_FIELDCAT.
    LS_FIELDCAT-FIELDNAME    = 'ARKTX'.
    LS_FIELDCAT-COL_POS      = 7.
    LS_FIELDCAT-SELTEXT_L    = 'Material Description'.
    LS_FIELDCAT-OUTPUTLEN    = 25.
    APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

* Req.Del.Date
    CLEAR LS_FIELDCAT.
    LS_FIELDCAT-FIELDNAME    = 'VDATU'.
    LS_FIELDCAT-COL_POS      = 8.
    LS_FIELDCAT-SELTEXT_L    = 'Req.Del.Date'.
    LS_FIELDCAT-OUTPUTLEN    = 12.
    APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

* SO QTY
    CLEAR LS_FIELDCAT.
    LS_FIELDCAT-FIELDNAME    = 'KWMENG'.
    LS_FIELDCAT-COL_POS      = 9.
    LS_FIELDCAT-SELTEXT_L    = 'Order Qty.'.
    APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

* SO Value
  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'NETWR'.
  LS_FIELDCAT-COL_POS      = 10.
```

```
  LS_FIELDCAT-SELTEXT_L    = 'Order Value'.
  LS_FIELDCAT-OUTPUTLEN    = 15.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

*Exchange Rate
  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'KURSK'.
  LS_FIELDCAT-COL_POS      = 11.
  LS_FIELDCAT-SELTEXT_L    = 'Exch.Rate'.
  LS_FIELDCAT-OUTPUTLEN    = 10.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

*Amount In Local Currency
  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'LOCVAL'.
  LS_FIELDCAT-COL_POS      = 12.
  LS_FIELDCAT-SELTEXT_L    = 'Order Val.Loc.Curr.'.
  LS_FIELDCAT-DO_SUM       = 'X'.
  LS_FIELDCAT-OUTPUTLEN    = 15.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

*  ENDIF.

  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'DLVQTY'.
  LS_FIELDCAT-COL_POS      = 13.
  LS_FIELDCAT-SELTEXT_L    = 'Delivered Qty.'.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

*  IF C_SUMREP = ''.
* Open Order Value
    CLEAR LS_FIELDCAT.
    LS_FIELDCAT-FIELDNAME    = 'OPENQTY'.
    LS_FIELDCAT-COL_POS      = 14.
    LS_FIELDCAT-SELTEXT_L    = 'Balance Qty.'.
    APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.
*    ENDIF.

* Open Order Value
  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'BALWR'.
  LS_FIELDCAT-COL_POS      = 15.
  LS_FIELDCAT-SELTEXT_L    = 'Open Order Val.'.
  LS_FIELDCAT-OUTPUTLEN    = 15.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

* Open Order Value In Local Currency
  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'BALLOC'.
  LS_FIELDCAT-COL_POS      = 16.
  LS_FIELDCAT-SELTEXT_L    = 'Open Val.Loc.Curr.'.
  LS_FIELDCAT-DO_SUM       = 'X'.
  LS_FIELDCAT-OUTPUTLEN    = 15.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

* CUSTOMER NO
  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'WAERK'.
```

```
  LS_FIELDCAT-COL_POS      = 17.
  LS_FIELDCAT-SELTEXT_L    = 'Currency'.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'EDATU'.
  LS_FIELDCAT-COL_POS      = 18.
  LS_FIELDCAT-SELTEXT_L    = 'Schedule Line Date'.
  LS_FIELDCAT-OUTPUTLEN    = 15.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'WMENG'.
  LS_FIELDCAT-COL_POS      = 19.
  LS_FIELDCAT-SELTEXT_L    = 'Schedule Qty.'.
  LS_FIELDCAT-OUTPUTLEN    = 10.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'BMENG'.
  LS_FIELDCAT-COL_POS      = 20.
  LS_FIELDCAT-SELTEXT_L    = 'Confirm Qty.'.
  LS_FIELDCAT-OUTPUTLEN    = 10.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.


* CUSTOMER NO
  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'KUNNR'.
  LS_FIELDCAT-COL_POS      = 21.
  LS_FIELDCAT-SELTEXT_L    = 'Customer No.'.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

* CUSTOMER NAME
  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'NAME1'.
  LS_FIELDCAT-COL_POS      = 22.
  LS_FIELDCAT-SELTEXT_L    = 'Customer Name'.
  LS_FIELDCAT-OUTPUTLEN    = 35.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

* CITY
  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'CITY1'.
  LS_FIELDCAT-COL_POS      = 23.
  LS_FIELDCAT-SELTEXT_L    = 'Destination'.
  LS_FIELDCAT-OUTPUTLEN    = 20.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.


* PLANT
  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'WERKS'.
  LS_FIELDCAT-COL_POS      = 24.
  LS_FIELDCAT-SELTEXT_L    = 'Plant'.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

* Sales Office
```

```
  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'VKBUR'.
  LS_FIELDCAT-COL_POS      = 25.
  LS_FIELDCAT-SELTEXT_L    = 'Sales Office'.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.


*  IF C_SUMREP = ''.
*Item Delivery / Quotation Status
    CLEAR LS_FIELDCAT.
    LS_FIELDCAT-FIELDNAME    = 'LFSTA'.
    LS_FIELDCAT-SELTEXT_L    = 'Item Del.Status'.
    LS_FIELDCAT-COL_POS      = 26.
    LS_FIELDCAT-OUTPUTLEN    = 14.
    APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.
*  ENDIF.

  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'CRSTAT'.
  LS_FIELDCAT-SELTEXT_L    = 'Credit Check Status'.
  LS_FIELDCAT-COL_POS      = 27.
  LS_FIELDCAT-OUTPUTLEN    = 40.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'VTEXT'.
  LS_FIELDCAT-SELTEXT_L    = 'Delivery Block'.
  LS_FIELDCAT-COL_POS      = 28.
  LS_FIELDCAT-OUTPUTLEN    = 40.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

  CLEAR LS_FIELDCAT.
  LS_FIELDCAT-FIELDNAME    = 'CARRIER'.
  LS_FIELDCAT-SELTEXT_L    = 'Carrier'.
  LS_FIELDCAT-COL_POS      = 29.
  LS_FIELDCAT-OUTPUTLEN    = 30.
  APPEND LS_FIELDCAT TO E01_LT_FIELDCAT.

ENDFORM.                    " e01_fieldcat_init
*&---------------------------------------------------------------------*
*
*&      Form  display_report
*&---------------------------------------------------------------------*
*
*       text
*----------------------------------------------------------------------*
*
*      -->P_FINAL  text
*----------------------------------------------------------------------*
*
FORM DISPLAY_REPORT  TABLES ITAB.
  GS_layout-colwidth_optimize = 'X'.

  CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
    EXPORTING
      I_BACKGROUND_ID         = 'ALV_BACKGROUND'
      I_CALLBACK_PROGRAM      = G_REPID
      I_CALLBACK_PF_STATUS_SET = 'S100'
```

```
      I_CALLBACK_USER_COMMAND = 'USER_COMMAND '
      IS_LAYOUT               = GS_LAYOUT
      IT_FIELDCAT             = GT_FIELDCAT[]
      IT_SPECIAL_GROUPS       = GT_SP_GROUP[]
      IT_SORT                 = GT_SORT[]
      I_SAVE                  = G_SAVE
      IS_VARIANT              = G_VARIANT
      IT_EVENTS               = GT_EVENTS[]
      IS_PRINT                = GS_PRINT
    TABLES
      T_OUTTAB                = FINAL.

ENDFORM.                    " display_report
*&---------------------------------------------------------------------*
*
*&      Form  e04_comment_build
*&---------------------------------------------------------------------*
*
*       text
*----------------------------------------------------------------------*
*
*      -->P_GT_LIST_TOP_OF_PAGE[]  text
*----------------------------------------------------------------------*
*
FORM E04_COMMENT_BUILD  USING E04_LT_TOP_OF_PAGE TYPE
SLIS_T_LISTHEADER.
  DATA: LS_LINE TYPE SLIS_LISTHEADER,V_TEXT(50) TYPE C.
  DATA: MSTRLEN TYPE I.
  DATA: MDATE LIKE SY-DATUM.

  CLEAR E04_LT_TOP_OF_PAGE.
  REFRESH E04_LT_TOP_OF_PAGE.
  LS_LINE-TYP  = 'H'.
  LS_LINE-INFO = 'English Indian Clays Limited'.
  APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.
*  IF C_SUMREP = 'X'.
*     LS_LINE-INFO = 'Sales Order Summary'.
*  ELSE.
     LS_LINE-INFO = 'Dispatch Plan & Sales Order Status'.
*     ENDIF.
  APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.

  CLEAR LS_LINE.
  LS_LINE-TYP  = 'S'.
  LS_LINE-KEY = 'Print Date & Time'.
  CONCATENATE SY-DATUM+6(2) '/' SY-DATUM+4(2) '/' SY-DATUM(4) INTO
LS_LINE-INFO.
  CONCATENATE LS_LINE-INFO '***' SY-UZEIT(2) ':' SY-UZEIT+2(2) ':' SY-
UZEIT+4(2) INTO LS_LINE-INFO.
  APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.

  IF NOT S_VKORG-LOW IS INITIAL.
    CLEAR LS_LINE.
    LS_LINE-TYP  = 'S'.
    LS_LINE-KEY = 'Sales Organisation'.
    LS_LINE-INFO = S_VKORG-LOW.
    IF NOT S_VKORG-HIGH IS INITIAL.
```

```
      CONCATENATE LS_LINE-INFO '-    To    -' S_VKORG-HIGH INTO
LS_LINE-INFO.
    ENDIF.
    APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.
  ENDIF.

  IF NOT S_VKBUR-LOW IS INITIAL.
    CLEAR LS_LINE.
    LS_LINE-TYP  = 'S'.
    LS_LINE-KEY = 'Sales Office'.
    READ TABLE S_VKBUR INDEX 1.
    IF S_VKBUR-HIGH IS INITIAL.
      LOOP AT S_VKBUR.
        IF SY-TABIX = 1.
          CONCATENATE LS_LINE-INFO S_VKBUR-LOW INTO  LS_LINE-INFO.
        ELSE.
          CONCATENATE LS_LINE-INFO S_VKBUR-LOW INTO  LS_LINE-INFO
SEPARATED BY ', '.
        ENDIF.
        CONDENSE LS_LINE-INFO.
        MSTRLEN = STRLEN( LS_LINE-INFO ).
        IF MSTRLEN = 58.
          APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.
          CLEAR: LS_LINE-INFO, LS_LINE-KEY.
        ENDIF.
      ENDLOOP.
      IF LS_LINE-INFO <> ', ' AND LS_LINE-INFO <> ''.
        APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.
      ENDIF.
    ELSE.
      CONCATENATE S_VKBUR-LOW '-    To    -' S_VKBUR-HIGH INTO LS_LINE-
INFO.
      APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.
    ENDIF.
  ENDIF.

  IF NOT S_WERKS-LOW IS INITIAL.
    CLEAR LS_LINE.
    LS_LINE-TYP  = 'S'.
    LS_LINE-KEY = 'Plant'.
    READ TABLE S_WERKS INDEX 1.
    IF S_WERKS-HIGH IS INITIAL.
      LOOP AT S_WERKS.
        IF SY-TABIX = 1.
          CONCATENATE LS_LINE-INFO S_WERKS-LOW INTO  LS_LINE-INFO.
        ELSE.
          CONCATENATE LS_LINE-INFO S_WERKS-LOW INTO  LS_LINE-INFO
SEPARATED BY ', '.
        ENDIF.
        CONDENSE LS_LINE-INFO.
        MSTRLEN = STRLEN( LS_LINE-INFO ).
        IF MSTRLEN = 58.
          APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.
          CLEAR: LS_LINE-INFO, LS_LINE-KEY.
        ENDIF.
      ENDLOOP.
      IF LS_LINE-INFO <> ', ' AND LS_LINE-INFO <> ''.
        APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.
```

```
      ENDIF.
    ELSE.
      CONCATENATE S_WERKS-LOW '-    To    -' S_WERKS-HIGH INTO LS_LINE-
INFO.
      APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.
    ENDIF.
  ENDIF.

  IF NOT S_VBELN-LOW IS INITIAL.
    CLEAR LS_LINE.
    LS_LINE-TYP  = 'S'.
    LS_LINE-KEY  = 'Quotation / Order No.'.
    LS_LINE-INFO = S_VBELN-LOW.
    IF NOT S_VBELN-HIGH IS INITIAL.
      CONCATENATE LS_LINE-INFO '-    To    -' S_VBELN-HIGH INTO
LS_LINE-INFO.
    ENDIF.
    APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.
  ENDIF.

  IF NOT S_KUNNR-LOW IS INITIAL.
    CLEAR LS_LINE.
    LS_LINE-TYP  = 'S'.
    LS_LINE-KEY = 'Customer'.
    LS_LINE-INFO = S_KUNNR-LOW .
    IF NOT S_KUNNR-HIGH IS INITIAL.
      CONCATENATE LS_LINE-INFO '-    To    -' S_KUNNR-HIGH INTO
LS_LINE-INFO.
    ENDIF.
    APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.
  ENDIF.

  IF NOT S_MATNR-LOW IS INITIAL.
    CLEAR LS_LINE.
    LS_LINE-TYP  = 'S'.
    LS_LINE-KEY = 'Material No.'.
    LS_LINE-INFO = S_MATNR-LOW .
    IF NOT S_MATNR-HIGH IS INITIAL.
      CONCATENATE LS_LINE-INFO '-    To    -' S_MATNR-HIGH INTO
LS_LINE-INFO.
    ENDIF.
    APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.
  ENDIF.

  IF NOT S_AUART-LOW IS INITIAL.
    CLEAR LS_LINE.
    LS_LINE-TYP  = 'S'.
    LS_LINE-KEY = 'Order Type'.
    IF NOT S_AUART-LOW IS INITIAL.
      LS_LINE-INFO = S_AUART-LOW.
    ENDIF.
    IF NOT S_AUART-HIGH IS INITIAL.
      CONCATENATE LS_LINE-INFO '-    To    -' S_AUART-HIGH INTO
LS_LINE-INFO.
    ENDIF.
    APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.
  ENDIF.
```

```
    CLEAR LS_LINE.
    IF S_EDATU[] IS INITIAL.
       LS_LINE-INFO = 'As On Today'.
    ELSE.
       MDATE = S_EDATU-LOW.
       IF S_EDATU-LOW IS INITIAL.
          MDATE = '20060101'.
          ENDIF.
       CONCATENATE MDATE+6(2) '/' MDATE+4(2) '/' MDATE+(4) INTO
LS_LINE-INFO.
       ENDIF.
    LS_LINE-TYP  = 'S'.
    LS_LINE-KEY = 'Schedule Line Date'.
    IF NOT S_EDATU-HIGH IS INITIAL.
      CONCATENATE LS_LINE-INFO '-    To    - ' S_EDATU-HIGH+6(2) '/'
S_EDATU-HIGH+4(2) '/' S_EDATU-HIGH+(4) INTO LS_LINE-INFO.
    ELSE.
      IF LS_LINE-INFO <> 'As On Today'.
         CONCATENATE LS_LINE-INFO '-    To    - ' S_EDATU-LOW+6(2) '/'
S_EDATU-LOW+4(2) '/' S_EDATU-LOW+(4) INTO LS_LINE-INFO.
         ENDIF.
    ENDIF.
    APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.


  IF NOT S_STAT-LOW IS INITIAL.
    CLEAR LS_LINE.
    LS_LINE-TYP  = 'S'.
    LS_LINE-KEY = 'Item Delivery Status'.
    IF NOT S_STAT-LOW IS INITIAL.
      LS_LINE-INFO = S_STAT-LOW.
    ENDIF.
    IF NOT S_STAT-HIGH IS INITIAL.
      CONCATENATE LS_LINE-INFO '-    To    -' S_STAT-HIGH INTO LS_LINE-
INFO.
    ENDIF.
    APPEND LS_LINE TO E04_LT_TOP_OF_PAGE.
  ENDIF.



ENDFORM.                    " e04_comment_build


*&---------------------------------------------------------------------*
*
*&      Form  TOP_OF_PAGE
*&---------------------------------------------------------------------*
*
*       text
*----------------------------------------------------------------------*
*
*      -->P_GT_LIST_TOP_OF_PAGE[]  text
*----------------------------------------------------------------------*
*

FORM TOP_OF_PAGE.
  CALL FUNCTION 'REUSE_ALV_COMMENTARY_WRITE'
```

```
    EXPORTING
      IT_LIST_COMMENTARY = GT_LIST_TOP_OF_PAGE.
ENDFORM.                    "top_of_page


*---------------------------------------------------------------------*
*       ........                                                      *
*---------------------------------------------------------------------*
FORM USER_COMMAND USING R_UCOMM LIKE
  SY-UCOMM RS_SELFIELD TYPE SLIS_SELFIELD.
  CASE SY-UCOMM.
     WHEN '&STK'.
        PERFORM SHOW_STOCK.
     WHEN 'OTHERS'.
        READ TABLE FINAL INDEX RS_SELFIELD-TABINDEX.
        SET PARAMETER ID 'AUN' FIELD FINAL-VBELN.
        CALL TRANSACTION 'VA03' AND SKIP FIRST SCREEN.
     ENDCASE.

ENDFORM.                    "USER_COMMAND

 FORM S100 USING lt_extab type slis_t_extab.
  set pf-status 'S100'.


ENDFORM.                    "USER_COMMAND

*&---------------------------------------------------------------------
*
*&      Form  SUMMARISE_DATA
*&---------------------------------------------------------------------
*
*       text
*----------------------------------------------------------------------
*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------
*
FORM SUMMARISE_DATA .
  DATA: MNETWR LIKE VBAP-NETWR,
        MBALWR LIKE VBAP-NETWR,
        MLOCVAL LIKE VBAP-NETWR,
        MBALLOC LIKE VBAP-NETWR.

  SORT FINAL BY VBELN.
  FINAL1[] = FINAL[].
  REFRESH FINAL.
  CLEAR FINAL.
  LOOP AT FINAL1.
    AT END OF VBELN.
      SUM.
      MNETWR = FINAL1-NETWR.
      MBALWR = FINAL1-BALWR.
      MLOCVAL = FINAL1-LOCVAL.
      MBALLOC = FINAL1-BALLOC.
    ENDAT.
    IF MNETWR > 0.
```

```
      FINAL = FINAL1.
      FINAL-NETWR = MNETWR.
      FINAL-BALWR = MBALWR.
      FINAL-LOCVAL = MLOCVAL.
      FINAL-BALLOC = MBALLOC.
      APPEND FINAL.
      MNETWR = MBALWR = MLOCVAL = 0.
    ENDIF.
  ENDLOOP.


ENDFORM.                    " SUMMARISE_DATA
*&---------------------------------------------------------------------*
*
*&      Form  GET_CREDIT_CHECK
*&---------------------------------------------------------------------*
*
*       text
*----------------------------------------------------------------------*
*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
*
FORM GET_CREDIT_CHECK .
DATA: MDOMNAME LIKE  DD07V-DOMNAME,
      MDOMVALUE LIKE  DD07V-DOMVALUE_L,
      MDDTEXT LIKE  DD07V-DDTEXT.

FINAL1[] = FINAL[].
SORT FINAL1 BY VBELN.
DELETE ADJACENT DUPLICATES FROM FINAL1 COMPARING VBELN.

SELECT VBELN CMGST INTO CORRESPONDING FIELDS OF TABLE I_VBUK
       FROM VBUK
       FOR ALL ENTRIES IN FINAL1
       WHERE VBELN = FINAL1-VBELN.

LOOP AT I_VBUK.
    MDOMNAME  = 'CMGST'.
    MDOMVALUE = I_VBUK-CMGST.

    CALL FUNCTION 'DOMAIN_VALUE_GET'
      EXPORTING
        I_DOMNAME        = MDOMNAME
        I_DOMVALUE       = MDOMVALUE
      IMPORTING
        E_DDTEXT         = MDDTEXT
      EXCEPTIONS
        NOT_EXIST        = 1
        OTHERS           = 2 .

      I_VBUK-CRSTAT = MDDTEXT.
      MODIFY I_VBUK.
      CLEAR MDDTEXT.
      ENDLOOP.

LOOP AT I_VBUK.
```

```
     LOOP AT FINAL WHERE VBELN = I_VBUK-VBELN.
         FINAL-CRSTAT = I_VBUK-CRSTAT.
         MODIFY FINAL.
         ENDLOOP.
     ENDLOOP.

LOOP AT FINAL WHERE CRSTAT = ''.
    FINAL-CRSTAT = 'Released'.
    MODIFY FINAL.
    ENDLOOP.

ENDFORM.                    " GET_CREDIT_CHECK
*&---------------------------------------------------------------------*
*
*&      Form  GET_DLV_BLOCK
*&---------------------------------------------------------------------*
*
*       text
*----------------------------------------------------------------------*
*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
*
FORM GET_DLV_BLOCK .
FINAL1[] = FINAL[].
SORT FINAL1 BY LIFSK.
DELETE FINAL1 WHERE LIFSK = ''.
DELETE ADJACENT DUPLICATES FROM FINAL1 COMPARING LIFSK.

LOOP AT FINAL1.
   SELECT SINGLE VTEXT INTO FINAL1-VTEXT
        FROM TVLST
        WHERE SPRAS EQ SY-LANGU
        AND   LIFSP EQ FINAL1-LIFSK.
   MODIFY FINAL1.
   ENDLOOP.

SORT FINAL BY LIFSK.
LOOP AT FINAL1.
    LOOP AT FINAL WHERE LIFSK = FINAL1-LIFSK.
        FINAL-VTEXT = FINAL1-VTEXT.
        MODIFY FINAL.
        ENDLOOP.
    ENDLOOP.

LOOP AT FINAL WHERE VTEXT = ''.
    FINAL-VTEXT = 'Released'.
    MODIFY FINAL.
    ENDLOOP.
SORT FINAL BY VKORG WERKS KUNNR VBELN POSNR EDATU.

ENDFORM.                    " GET_DLV_BLOCK

*&---------------------------------------------------------------------*
*
*&      Form  GET_CARRIER
```

```
*&---------------------------------------------------------------------*
*
*       text
*----------------------------------------------------------------------*
*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
*
FORM GET_CARRIER .
DATA: MID   LIKE THEAD-TDID,
      MNAME LIKE THEAD-TDNAME,
      MOBJ  LIKE THEAD-TDOBJECT VALUE 'VBBK',
      MLANG LIKE THEAD-TDSPRAS VALUE 'E'.

DATA: TDLINE LIKE TLINE OCCURS 0 WITH HEADER LINE.

FINAL1[] = FINAL[].
SORT FINAL1 BY VBELN.
DELETE ADJACENT DUPLICATES FROM FINAL1 COMPARING VBELN.

LOOP AT FINAL1.
    REFRESH TDLINE.
    CLEAR TDLINE.
    MNAME = FINAL1-VBELN.
    MID = 'Z001'.
    CALL FUNCTION 'READ_TEXT'
      EXPORTING
        CLIENT                        = SY-MANDT
        ID                            = MID
        LANGUAGE                      = MLANG
        NAME                          = MNAME
        OBJECT                        = MOBJ
      TABLES
        LINES                         = TDLINE
      EXCEPTIONS
        ID                            = 1
        LANGUAGE                      = 2
        NAME                          = 3
        NOT_FOUND                     = 4
        OBJECT                        = 5
        REFERENCE_CHECK               = 6
        WRONG_ACCESS_TO_ARCHIVE       = 7
        OTHERS                        = 8  .

      LOOP AT TDLINE.
           CONCATENATE FINAL1-CARRIER TDLINE-TDLINE INTO FINAL1-CARRIER
SEPARATED BY SPACE.
           ENDLOOP.
      MODIFY FINAL1.
    ENDLOOP.

LOOP AT FINAL1.
     LOOP AT FINAL WHERE VBELN = FINAL1-VBELN.
         FINAL-CARRIER = FINAL1-CARRIER.
         MODIFY FINAL.
         ENDLOOP.
     ENDLOOP.
```

```
ENDFORM.                    " GET_CARRIER

*&---------------------------------------------------------------------*
*
*&      Form  SHOW_STOCK
*&---------------------------------------------------------------------*
*
*       text
*----------------------------------------------------------------------*
*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
*
form SHOW_STOCK .

RANGES: RMATNR FOR MARA-MATNR,
        RWERKS FOR VBAP-WERKS.

FINAL1[] = FINAL[].
SORT FINAL1 BY MATNR.
DELETE ADJACENT DUPLICATES FROM FINAL1 COMPARING MATNR.

RMATNR-OPTION = 'EQ'.
RMATNR-SIGN = 'I'.
LOOP AT FINAL1.
    RMATNR-LOW = FINAL1-MATNR.
    APPEND RMATNR.
    ENDLOOP.

RWERKS-OPTION = 'EQ'.
RWERKS-SIGN = 'I'.
LOOP AT FINAL1.
    RWERKS-LOW = FINAL1-WERKS.
    APPEND RWERKS.
    ENDLOOP.

FINAL1[] = FINAL[].
SORT FINAL1 BY WERKS.
DELETE ADJACENT DUPLICATES FROM FINAL1 COMPARING WERKS.

SUBMIT RM07MLBS WITH MATNR IN RMATNR
                WITH WERKS IN RWERKS
                WITH P_VARI EQ '/ZTRV'
                AND RETURN.

endform.                    " SHOW_STOCK
??

??

??

??
```