





\*&---------------------------------------------------------------------\*  
\*& Report  ZTEST\_PO\_MONITOR  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
report ZTEST\_PO\_MONITOR.  
  
  
data : LV\_DATE type SY-DATUM.  
  
parameters: P\_DATE type SY-DATUM.  
  
start-of-selection.  
  
  LV\_DATE = P\_DATE - 2000.  
  
  select A~EBELN, A~AEDAT, B~EBELP, B~MATNR, B~LGORT, B~MENGE, B~MEINS, B~PLIFZ  
      from EKKO as A inner join EKPO as B on A~EBELN = B~EBELN  
        into table @data(LT\_PO\_DATA)  
        where A~AEDAT ge @LV\_DATE and  
              B~PLIFZ gt 10.  
  if SY-SUBRC eq 0.  
  
    loop at LT\_PO\_DATA assigning field-symbol(<LFS\_DATA>)..  
      write: <LFS\_DATA>-EBELN,  
             <LFS\_DATA>-AEDAT,  
             <LFS\_DATA>-EBELP,  
             <LFS\_DATA>-MATNR,  
             <LFS\_DATA>-LGORT,  
             <LFS\_DATA>-MENGE,  
             <LFS\_DATA>-MEINS,  
             <LFS\_DATA>-PLIFZ .  
    endloop.  
  
  endif.

\*&---------------------------------------------------------------------\*  
\*& Report  ZTEST\_BCKND\_FRMWRK  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
report ZTEST\_BCKND\_FRMWRK.  
  
  
  
data :  LV\_JOBCOUNT   type BTCJOBCNT.  
  
  
select \* from ZBGD\_MONTR\_CUST into table @data(LT\_BCKGD)  
          where ENDDT gt @SY-DATUM.  
if SY-SUBRC eq 0.  
  
  loop at LT\_BCKGD assigning field-symbol(<LFS\_BCK>).  
    call function 'JOB\_OPEN'  
      exporting  
        JOBNAME          = <LFS\_BCK>-JOBNAME  
        SDLSTRTTM        = <LFS\_BCK>-STRTM  
      importing  
        JOBCOUNT         = LV\_JOBCOUNT  
      exceptions  
        CANT\_CREATE\_JOB  = 1  
        INVALID\_JOB\_DATA = 2  
        JOBNAME\_MISSING  = 3  
        others           = 4.  
  
  
    submit (<LFS\_BCK>-PGMNAME) using selection-set <LFS\_BCK>-VARIANT  
                      via job <LFS\_BCK>-JOBNAME  
                      number LV\_JOBCOUNT  
                      and return.  
  
    submit ZTEST\_READ\_SPOOL\_DATA with P\_PGM = <LFS\_BCK>-PGMNAME  
                                 with P\_DATE = SY-DATUM  
                                 via job <LFS\_BCK>-JOBNAME  
                                 number LV\_JOBCOUNT and return.  
  
    call function 'JOB\_CLOSE'  
      exporting  
        JOBCOUNT             = LV\_JOBCOUNT  
        JOBNAME              = <LFS\_BCK>-JOBNAME  
        STRTIMMED            = 'X'  
      exceptions  
        CANT\_START\_IMMEDIATE = 1  
        INVALID\_STARTDATE    = 2  
        JOBNAME\_MISSING      = 3  
        JOB\_CLOSE\_FAILED     = 4  
        JOB\_NOSTEPS          = 5  
        JOB\_NOTEX            = 6  
        LOCK\_FAILED          = 7  
        others               = 8.  
  
  endloop.  
  
endif.

\*&---------------------------------------------------------------------\*  
\*& Report  ZTEST\_READ\_SPOOL\_DATA  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
report ZTEST\_READ\_SPOOL\_DATA.  
  
data: LV\_SPL\_TIME type RSPOCRTIME,  
      LT\_DATA     type table of SOLI.  
  
parameters: P\_PGM  type PROGRAMM,  
            P\_DATE type SY-DATUM.  
  
  
start-of-selection.  
  
  data(LV\_PBM\_NM) = P\_PGM+0(12).  
  
  LV\_SPL\_TIME = P\_DATE && | '00000000' |.  
  
  select RQIDENT from TSP01 up to 1 rows  
      into @data(LV\_SPOOL)  
      where RQ2NAME = @LV\_PBM\_NM and  
            RQCRETIME gt @LV\_SPL\_TIME .  
  endselect.  
  if SY-SUBRC eq 0.  
  
    call function 'RSPO\_RETURN\_SPOOLJOB'  
      exporting  
        RQIDENT              = LV\_SPOOL  
        FIRST\_LINE           = 1  
\*       LAST\_LINE            =  
        DESIRED\_TYPE         = 'RAW'  
\* IMPORTING  
\*       REAL\_TYPE            =  
\*       SP\_LANG              =  
      tables  
        BUFFER               = LT\_DATA  
\*       BUFFER\_PDF           =  
      exceptions  
        NO\_SUCH\_JOB          = 1  
        JOB\_CONTAINS\_NO\_DATA = 2  
        SELECTION\_EMPTY      = 3  
        NO\_PERMISSION        = 4  
        CAN\_NOT\_ACCESS       = 5  
        READ\_ERROR           = 6  
        TYPE\_NO\_MATCH        = 7  
        others               = 8.  
    if SY-SUBRC <> 0.  
\* Implement suitable error handling here  
    endif.  
  
    loop at LT\_DATA assigning field-symbol(<LFS\_DATA>).  
      write : <LFS\_DATA>-LINE.  
  
    endloop.  
  
  endif.