report  zlr\_program\_14.

include: <icon>.
type-pools: seok.
tables: tadir.

types: begin of t\_info,
         object type char4,
         status type c,
         ddtext type ddtext,
       end of t\_info.

Types:

begin of t\_rep,
         name type sobj\_name,
       end of t\_rep.

types: t\_reps type table of t\_rep.
types t\_infos type table of t\_info.

  data: rep         type t\_rep,
        dummy\_trdir type trdir.
data:
    report  type progname.
data: reports type t\_reps.

select-options: so\_devc for tadir-devclass obligatory.
parameters: p\_path(255) type c obligatory.
parameters: p\_a as checkbox default 'X'.
parameters: p\_b   as checkbox default 'X'.
select-options so\_rep  for report.

at selection-screen on value-request for p\_path.
  data: folder    type string.

  call method cl\_gui\_frontend\_services=>directory\_browse
    changing
      selected\_folder = folder
    exceptions
      cntl\_error      = 1
      error\_no\_gui    = 2
      others          = 3.

  if sy-subrc = 0.
    p\_path = folder.
  endif.

start-of-selection.

  perform get\_reports.

  if reports is initial.
    write: /, icon\_red\_light, 'No objects found for export'.
    exit.
  endif.

  types: ts\_line  type string.

  data: lv\_sysid     type sysysid,
        lv\_timestamp type timestamp,
        lv\_date(8)   type c,
        lv\_filename  type string.
  data: ls\_rep type t\_rep,
        ls\_log type t\_info.
  data: lt\_source\_code  type table of ts\_line.

  lv\_sysid = sy-sysid.
  ls\_log-object = 'PROG'.

data ev\_count\_reps\_err type i.
data ct\_infos type t\_infos.
data ev\_subrc type sysubrc.
data ev\_count\_reps\_ok type i.

  loop at reports into ls\_rep.
    clear lt\_source\_code.

    read report ls\_rep-name into lt\_source\_code.
    if sy-subrc <> 0.
      ev\_count\_reps\_err = ev\_count\_reps\_err + 1.
      ls\_log-status = 'R'.
      concatenate ls\_rep-name ' - problem reading report'
        into ls\_log-ddtext.
      append ls\_log to ct\_infos.
      continue.
    endif.

    if ls\_rep-name(1) = '/'.
      shift ls\_rep-name left deleting leading '/'.
    endif.
    concatenate p\_path '\' lv\_sysid '\_' ls\_rep-name '.abap'
      into lv\_filename.
    replace all occurrences of '/' in lv\_filename with '\_'.

    call function 'GUI\_DOWNLOAD'
      exporting
        filename                = lv\_filename
      tables
        data\_tab                = lt\_source\_code
      exceptions
        file\_write\_error        = 1
        no\_batch                = 2
        gui\_refuse\_filetransfer = 3
        invalid\_type            = 4
        no\_authority            = 5
        unknown\_error           = 6
        header\_not\_allowed      = 7
        separator\_not\_allowed   = 8
        filesize\_not\_allowed    = 9
        header\_too\_long         = 10
        dp\_error\_create         = 11
        dp\_error\_send           = 12
        dp\_error\_write          = 13
        unknown\_dp\_error        = 14
        access\_denied           = 15
        dp\_out\_of\_memory        = 16
        disk\_full               = 17
        dp\_timeout              = 18
        file\_not\_found          = 19
        dataprovider\_exception  = 20
        control\_flush\_error     = 21
        others                  = 22.
    if sy-subrc <> 0.
      ev\_subrc = sy-subrc.
      ev\_count\_reps\_err = ev\_count\_reps\_err + 1.

      ls\_log-status = 'R'.
      concatenate ls\_rep-name ' was not downloaded succesfully'
        into ls\_log-ddtext.
      append ls\_log to ct\_infos.
    elseif sy-subrc = 0.
      ev\_count\_reps\_ok = ev\_count\_reps\_ok + 1.

      ls\_log-status = 'G'.
      concatenate ls\_rep-name ' was downloaded succesfully'
        into ls\_log-ddtext.
      append ls\_log to ct\_infos.
    endif.

  endloop.

  data: lv\_count\_reps\_err\_string(5) type c,
        lv\_count\_reps\_ok\_string(5)  type c.
\*  DATA: ls\_log                      TYPE t\_info.

\*  lv\_count\_reps\_ok\_string = ev\_count\_reps\_ok.
\*  lv\_count\_reps\_err\_string = ev\_count\_reps\_err.
\*  concatenate 'Reports/Includes -> OK: ' lv\_count\_reps\_ok\_string
\*              ' / ERROR: ' lv\_count\_reps\_err\_string
\*              into ls\_log-ddtext.
\*
\*  append ls\_log to ct\_infos .

\*  PERFORM download\_objs\_to\_txt\_and\_log
\*    USING p\_path
\*          reports
\*          gt\_fms
\*          gt\_clas
\*    CHANGING gt\_infos.

\*  PERFORM display\_logs
\*   USING gt\_infos
\*         p\_a
\*         p\_b.

\*  uline.
\*  read table ct\_infos into ls\_log index 1.
\*  write: /, ls\_log.
\*  read table ct\_infos into ls\_log index 2.
\*  write: /, ls\_log.
\*  uline.
\*  if p\_a <> 'X' and p\_b <> 'X'. exit. endif.

  if p\_b = 'X'.
    loop at ct\_infos into ls\_log where status = 'R'.
      write: /, icon\_red\_light, ls\_log-object, ls\_log-ddtext.
    endloop.
  endif.

  if p\_a = 'X'.
    loop at ct\_infos into ls\_log where status = 'G'.
      write: /, icon\_green\_light, ls\_log-object, ls\_log-ddtext.
    endloop.
  endif.

form get\_reports.

  select obj\_name as name from tadir
    appending table reports
    where pgmid = 'R3TR'
      and object = 'PROG'
      and obj\_name in so\_rep
      and devclass in so\_devc
      and author = sy-uname
      and delflag <> 'X'.

endform.