Utilities to upload/download transport requests from/to front-end PC

SDN Community Contribution

(This is not an official SAP document.)

Disclaimer & Liability Notice

Utilities to upload/download transport requests from/to front-end PC

Applies To:

The utilities were originally developed for SAP R/3 4.6c, some additions were made under SAP ERP 4.7. The utilities could be used (though, were not thoroughly tested) in any SAP basis system above 4.6 which employ Transport Management System (TMS).

Summary

These are two simple and useful utilities to upload/download binary files of transport requests from/to front-end PC. They make possible to transfer transport requests between different systems without asking Basis people (nevertheless you have to have a sufficient authority). They also could be a demo of some TMS function modules usage. The utilities were originally developed by Igor Yaskevitch (senior basis/development consultant at IBS Company) and enhanced by Sergei Korolev (senior development consultant at IBS Company). This publication made with explicit permission of Igor Yaskevitch.

Utilities to upload/download transport requests from/to front-end PC

By: Sergei Korolev

Company: IBS Company

Date: September 10, 2005

Upload Transport request utility

This easy-to-use utility uploads transport request binaries from user front end PC, adds request to the import queue, and starts single request import. The report checks the following authority objects:

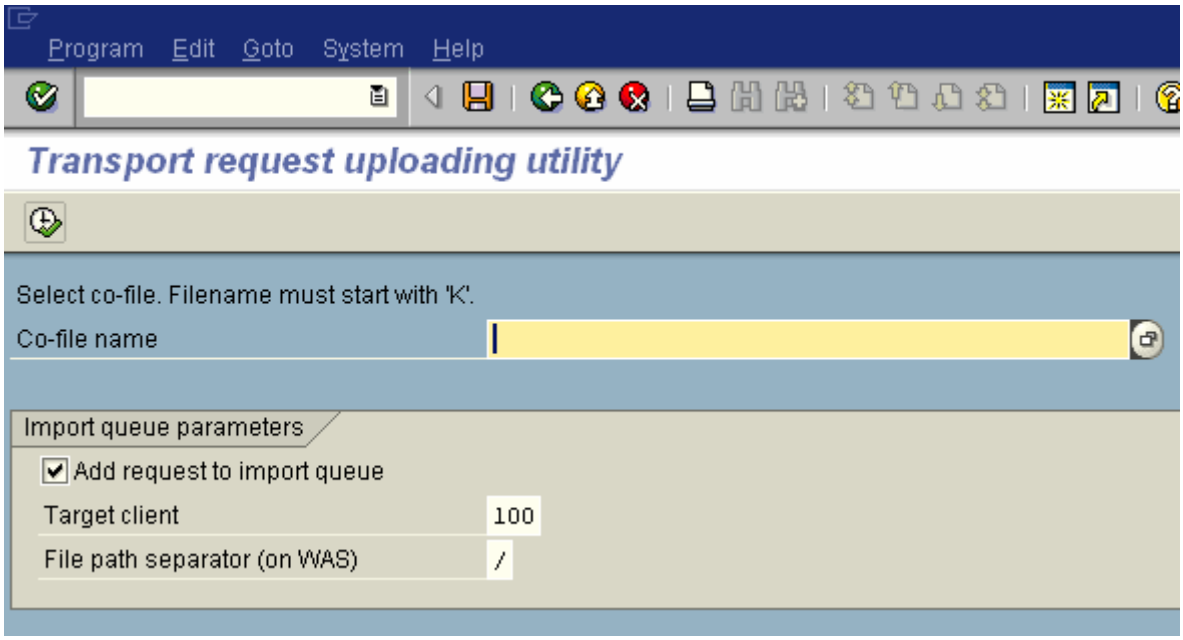
- S_CTS_ADMI, with values 'TADD' (authority to add request to the import queue), 'IMPS' (authority to start request import) for field CTS_ADMFCT
- S_DATASET with field values ACTVT = '34' (Write access for application server files), FILENAME = corresponding request file name

The utility has the following parameters:

- **Co-file name** - full file path of the request command file, file name must be of the form '**Knnnnnnn.SSS**', where '**nnnnnnn**' is the request number, and 'SSS' – source system identifier. Request command file can be selected via F4 value request. Remaining request data files must be located in the same frontend directory.
- **Add request to import queue** – if the flag is set then request is added to the import queue and also
- **Target client** – client number which should be assigned to the request*
- **File path separator (on WAS) –** separating character for building application server file path. The parameter is operating system dependent and is calculated in INITIALIZATION event. As a rule there is no need to change the default value.

---

* Currently **Target client** parameter does not work properly due to some incorrect TMS function usage (problem still unsolved).



Selection parameter texts

Set selection criteria texts according to table below.

Parameter	*Text*
P_ADDQUE	Add request to import queue
P_COFILE	Co-file name
P_SEPR	File path separator (on WAS)
P_TARCLI	Target client

Source code

```
*=====================================================================*
*   Initial idea and first release by Igor Yaskevitch (IBS), 2003     *
*   Enhancements by Sergei Korolev (IBS), 2005 (added import queue    *
*   manipulations, authority checking, minor interface improvements)  *
*---------------------------------------------------------------------*
*   Function     : This is a utility tool for uploading binary        *
*                  files of a transport request from a Client PC,     *
*                  adding to an import queue and importing into the   *
*                  system.                                            *
*=====================================================================*
REPORT zrs_upload_tr.

TYPE-POOLS:
  abap,
  sabc,
  stms.

CONSTANTS:
      gc_tp_fillclient       LIKE stpa-command   VALUE 'FILLCLIENT'.

DATA: lt_request             TYPE stms_tr_requests,
      lt_tp_maintain         TYPE stms_tp_maintains.

DATA: sl TYPE i,
      l_datafile(255) TYPE c,
      datafiles TYPE i,
      ret TYPE i,
      ans TYPE c.

DATA:
  et_request_infos TYPE stms_wbo_requests,
  request_info TYPE stms_wbo_request,
  system TYPE tmscsys-sysnam,
  request LIKE e070-trkorr.

DATA: folder TYPE string,
      retval LIKE TABLE OF ddshretval WITH HEADER LINE,
      fldvalue LIKE help_info-fldvalue,
      transdir TYPE text255,
      filename LIKE authb-filename,
      trfile(20) TYPE c.

DATA:
  BEGIN OF datatab OCCURS 0,
```

```
      buf(8192) TYPE c,
  END OF datatab.

DATA: len TYPE i,
      flen TYPE i.

SELECTION-SCREEN COMMENT /1(79) comm_sel.

PARAMETERS:
  p_cofile(255) TYPE c LOWER CASE OBLIGATORY.

SELECTION-SCREEN SKIP.
SELECTION-SCREEN BEGIN OF BLOCK b01 WITH FRAME TITLE bl_title.
PARAMETERS:
  p_addque AS CHECKBOX DEFAULT 'X',
  p_tarcli LIKE tmsbuffer-tarcli
           DEFAULT sy-mandt
           MATCHCODE OBJECT h_t000,
  p_sepr   OBLIGATORY.

SELECTION-SCREEN END OF BLOCK b01.

INITIALIZATION.
  bl_title = 'Import queue parameters'(b01).
  comm_sel = 'Select co-file. Filename must start with ''K''.'(001).

  CALL FUNCTION 'WSAF_BUILD_SEPARATOR'
    IMPORTING
      separator                  = p_sepr
    EXCEPTIONS
      separator_not_maintained   = 1
      wrong_call                 = 2
      wsaf_config_not_maintained = 3
      OTHERS                     = 4.

  IF sy-subrc NE 0.
    MESSAGE s001(00) WITH 'Unable to find out the separator symbol for the system.'(008).
  ENDIF.

AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_cofile.
  DATA:
    file TYPE file_table,
    rc TYPE i,
    title TYPE string,
    file_table TYPE filetable,
    file_filter TYPE string VALUE 'CO-files (K*.*)|K*.*||'.
```

```
  title = 'Select CO-file'(006).
  CALL METHOD cl_gui_frontend_services=>file_open_dialog
    EXPORTING
      window_title            = title
      file_filter             = file_filter
    CHANGING
      file_table              = file_table
      rc                      = rc
    EXCEPTIONS
      file_open_dialog_failed = 1
      cntl_error              = 2
      error_no_gui            = 3
      not_supported_by_gui    = 4
      OTHERS                  = 5.

  IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
               WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  ENDIF.

  READ TABLE file_table INTO file INDEX 1.

  p_cofile = file.

AT SELECTION-SCREEN.
  DATA:
    file TYPE string.

  sl = STRLEN( p_cofile ).
  IF sl < 11.
    MESSAGE e001(00)
      WITH 'Invalid co-file name format. File name format must be KNNNNNNN.SSS'(009).
  ENDIF.
  sl = sl - 11.
  IF p_cofile+sl(1) NE 'K'.
    MESSAGE e001(00)
      WITH 'Invalid co-file name format. File name format must be KNNNNNNN.SSS'(009).
  ENDIF.
  sl = sl + 1.
  IF NOT p_cofile+sl(6) CO '0123456789'.
    MESSAGE e001(00)
      WITH 'Invalid co-file name format. File name format must be KNNNNNNN.SSS'(009).
  ENDIF.
  sl = sl + 6.
  IF p_cofile+sl(1) NE '.'.
    MESSAGE e001(00)
      WITH 'Invalid co-file name format. File name format must be KNNNNNNN.SSS'(009).
```

```
  ENDIF.
  sl = sl - 7.
  CLEAR datafiles.
  l_datafile = p_cofile.
  l_datafile+sl(1) = 'R'.

  file = l_datafile.
  IF cl_gui_frontend_services=>file_exist( file = file ) = 'X'.
    ADD 1 TO datafiles.
  ENDIF.

  l_datafile+sl(1) = 'D'.
  file = l_datafile.
  IF cl_gui_frontend_services=>file_exist( file = file ) = 'X'.
    ADD 1 TO datafiles.
  ENDIF.

  sl = sl + 8.
  request = p_cofile+sl(3).
  sl = sl - 8.
  CONCATENATE request p_cofile+sl(7) INTO request.
  TRANSLATE request TO UPPER CASE.

  IF datafiles = 0.
    MESSAGE e398(00)
      WITH 'Corresponding data-files of transport request'(010)
           request
           'not found.'(011).
  ELSE.
    MESSAGE s398(00)
      WITH datafiles
           'data-files have been found for transport request'(012)
           request.
  ENDIF.

START-OF-SELECTION.
  DATA:
    parameter TYPE spar,
    parameters TYPE TABLE OF spar.

  CALL FUNCTION 'RSPO_R_SAPGPARAM'
    EXPORTING
      name   = 'DIR_TRANS'
    IMPORTING
      value  = transdir
    EXCEPTIONS
      error  = 1
```

```
    OTHERS = 2.

IF sy-subrc <> 0.
  MESSAGE ID sy-msgid TYPE 'E' NUMBER sy-msgno
          WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

filename = p_cofile+sl(11).
TRANSLATE filename TO UPPER CASE.

CONCATENATE transdir 'cofiles' filename
  INTO filename
  SEPARATED BY p_sepr.

OPEN DATASET filename FOR INPUT IN BINARY MODE.
ret = sy-subrc.
CLOSE DATASET filename.
IF NOT ret = 0.
  CALL FUNCTION 'POPUP_TO_CONFIRM'
    EXPORTING
      text_question  = 'Copy all files?'(a03)
    IMPORTING
      answer         = ans
    EXCEPTIONS
      text_not_found = 1
      OTHERS         = 2.
ELSE.
  parameter-param = 'FILE'.
  parameter-value = filename.
  APPEND parameter TO parameters.

  CALL FUNCTION 'POPUP_TO_CONFIRM'
    EXPORTING
      text_question  = 'File ''&FILE&'' already exists. Rewrite?'(a04)
    IMPORTING
      answer         = ans
    TABLES
      parameter      = parameters
    EXCEPTIONS
      text_not_found = 1
      OTHERS         = 2.
ENDIF.

CHECK ans = '1'.

trfile =  p_cofile+sl(11).
TRANSLATE trfile TO UPPER CASE.
```

```
  PERFORM copy_file USING 'cofiles' trfile p_cofile.
  trfile(1) = 'R'.
  l_datafile+sl(1) = 'R'.
  PERFORM copy_file USING 'data' trfile l_datafile.
  IF datafiles > 1.
    trfile(1) = 'D'.
    l_datafile+sl(1) = 'D'.
    PERFORM copy_file USING 'data' trfile l_datafile.
  ENDIF.

  IF p_addque = 'X'.
    system = sy-sysid.

    DO 1 TIMES.

*     Check authority to add request to the import queue
      CALL FUNCTION 'TR_AUTHORITY_CHECK_ADMIN'
        EXPORTING
          iv_adminfunction = 'TADD'
        EXCEPTIONS
          e_no_authority   = 1
          e_invalid_user   = 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.
        EXIT.
      ENDIF.

      CALL FUNCTION 'TMS_UI_APPEND_TR_REQUEST'
        EXPORTING
          iv_system             = system
          iv_request            = request
          iv_expert_mode        = 'X'
          iv_ctc_active         = 'X'
        EXCEPTIONS
          cancelled_by_user     = 1
          append_request_failed = 2
          OTHERS                = 3.

      CHECK sy-subrc = 0.

      CALL FUNCTION 'TMS_MGR_READ_TRANSPORT_REQUEST'
        EXPORTING
          iv_request                = request
```

```
          iv_target_system           = system
        IMPORTING
          et_request_infos           = et_request_infos
        EXCEPTIONS
          read_config_failed         = 1
          table_of_requests_is_empty = 2
          system_not_available       = 3
          OTHERS                     = 4.

      CLEAR request_info.
      READ TABLE et_request_infos INTO request_info INDEX 1.

      IF  request_info-e070-korrdev = 'CUST'
      AND NOT p_tarcli IS INITIAL.
        CALL FUNCTION 'TMS_MGR_MAINTAIN_TR_QUEUE'
          EXPORTING
            iv_command                = gc_tp_fillclient
            iv_system                 = system
            iv_request                = request
            iv_tarcli                 = p_tarcli
            iv_monitor                = 'X'
            iv_verbose                = 'X'
          IMPORTING
            et_tp_maintains           = lt_tp_maintain
          EXCEPTIONS
            read_config_failed         = 1
            table_of_requests_is_empty = 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.
          EXIT.
        ENDIF.
      ENDIF.

*     Check authority to start request import
      CALL FUNCTION 'TR_AUTHORITY_CHECK_ADMIN'
        EXPORTING
          iv_adminfunction = 'IMPS'
        EXCEPTIONS
          e_no_authority   = 1
          e_invalid_user   = 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.
        EXIT.
      ENDIF.

      CALL FUNCTION 'TMS_UI_IMPORT_TR_REQUEST'
        EXPORTING
          iv_system             = system
          iv_request            = request
          iv_tarcli             = p_tarcli
          iv_some_active        = space
        EXCEPTIONS
          cancelled_by_user     = 1
          import_request_denied = 2
          import_request_failed = 3
          OTHERS                = 4.

    ENDDO.
  ENDIF.

*&---------------------------------------------------------------------*
*&      Form  copy_file
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->SUBDIR     text
*      -->FNAME      text
*      -->SOURCE_FILEtext
*----------------------------------------------------------------------*
FORM copy_file USING subdir fname source_file.
  DATA: l_filename TYPE string.
  l_filename = source_file.
  CONCATENATE transdir subdir fname
    INTO filename
    SEPARATED BY p_sepr.
  REFRESH datatab.
  CLEAR flen.

  CALL METHOD cl_gui_frontend_services=>gui_upload
    EXPORTING
      filename                = l_filename
      filetype                = 'BIN'
    IMPORTING
      filelength              = flen
    CHANGING
      data_tab                = datatab[]
    EXCEPTIONS
      file_open_error         = 1
```


```
    file_read_error         = 2
    no_batch                = 3
    gui_refuse_filetransfer = 4
    invalid_type            = 5
    no_authority            = 6
    unknown_error           = 7
    bad_data_format         = 8
    header_not_allowed      = 9
    separator_not_allowed   = 10
    header_too_long         = 11
    unknown_dp_error        = 12
    access_denied           = 13
    dp_out_of_memory        = 14
    disk_full               = 15
    dp_timeout              = 16
    not_supported_by_gui    = 17
    error_no_gui            = 18
    OTHERS                  = 19.

IF sy-subrc NE 0.
  WRITE: / 'Error uploading file'(003), l_filename.
  EXIT.
ENDIF.

CALL FUNCTION 'AUTHORITY_CHECK_DATASET'
  EXPORTING
    activity         = sabc_act_write
    filename         = filename
  EXCEPTIONS
    no_authority     = 1
    activity_unknown = 2
    OTHERS           = 3.

IF sy-subrc <> 0.
  FORMAT COLOR COL_NEGATIVE.
  WRITE: / 'Write access denied. File'(013), filename.
  FORMAT COLOR OFF.
  EXIT.
ENDIF.

OPEN DATASET filename FOR OUTPUT IN BINARY MODE.
IF sy-subrc NE 0.
  WRITE: / 'File open error'(004), trfile.
  EXIT.
ENDIF.

LOOP AT datatab.
```

```
    IF flen < 8192.
      len = flen.
    ELSE.
      len = 8192.
    ENDIF.
    TRANSFER datatab-buf TO filename LENGTH len.
    flen = flen - len.
  ENDLOOP.
  CLOSE DATASET filename.
  WRITE: / 'File'(005), trfile, 'uploaded'(007).
ENDFORM.                    "copy_file
```

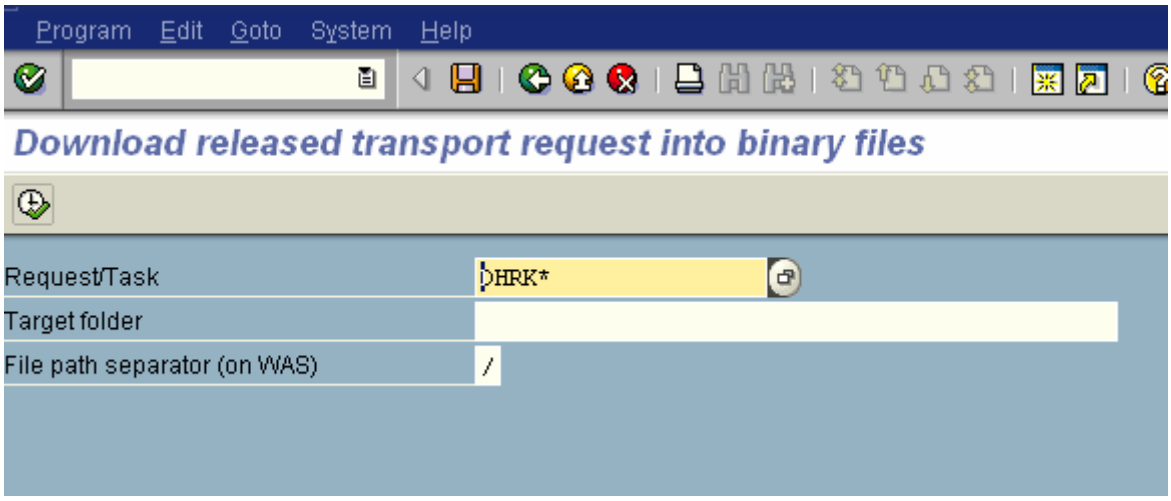
Download Transport request utility

The utility downloads transport request binaries from application server to the user client PC. The request must be released. The utility checks authority to access application server files, object S_DATASET with field values:

- ACTVT = '33' (Read access)
- FILENAME = corresponding request file name

The utility has the following parameters:

- **Request/Task** – Number of the request to download. The request must be released. Search help provides a tree list of released requests.
- **Target folder** – Full path name of client PC folder where request binary files are to be downloaded to.
- **File path separator (on WAS) –** separating character for building application server file path. The parameter is operating system dependent and is calculated in INITIALIZATION event. As a rule there is no need to change the default value.



Selection parameter texts

Parameter	*Text*
P_FOLDER	Target folder
P_REQEST	Request
P_SEPR	File path separator (on WAS)

Source code

```
*=====================================================================*
*   Initial idea and first release by Igor Yaskevitch (IBS), 2003     *
*   Enhancements by Sergei Korolev (IBS), 2005 (Added F4 value        *
*   requests,  authority checks, TMS function usage)                  *
*---------------------------------------------------------------------*
*   Function     : This is a utility tool for downloading binary      *
*                  files of transport request to a Client PC          *
*=====================================================================*
REPORT zrs_download_tr .
```

```
PARAMETERS:
  p_reqest      TYPE trkorr
                OBLIGATORY,
  p_folder(255) TYPE c
                LOWER CASE,
  p_sepr        OBLIGATORY.

DATA:
  folder     TYPE string,
  retval     LIKE TABLE OF ddshretval WITH HEADER LINE,
  fldvalue   LIKE help_info-fldvalue,
  transdir   TYPE text255,
  filename   LIKE authb-filename,
  trfile(20) TYPE c,

  datatab TYPE TABLE OF text8192 WITH HEADER LINE,

  len TYPE i,
  flen TYPE i.

TYPE-POOLS:
  sabc,
  stms,
  trwbo.

INITIALIZATION.
  CONCATENATE sy-sysid 'K*' INTO p_reqest.

  CALL FUNCTION 'WSAF_BUILD_SEPARATOR'
    IMPORTING
      separator                  = p_sepr
    EXCEPTIONS
      separator_not_maintained   = 1
      wrong_call                 = 2
      wsaf_config_not_maintained = 3
      OTHERS                     = 4.

  IF sy-subrc NE 0.
    MESSAGE s001(00) WITH 'Unable to find out the separator symbol for the system.'(011).
  ENDIF.

AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_reqest.
  DATA:
    tt_system            TYPE     TABLE OF tmscsys WITH HEADER LINE,
    es_selected_request  TYPE    trwbo_request_header,
    es_selected_task     TYPE    trwbo_request_header,
```

```
    iv_organizer_type    TYPE    trwbo_calling_organizer,
    is_selection         TYPE    trwbo_selection.

  iv_organizer_type = 'W'.
  is_selection-reqstatus = 'R'.

  CALL FUNCTION 'TR_PRESENT_REQUESTS_SEL_POPUP'
    EXPORTING
      iv_organizer_type   = iv_organizer_type
      is_selection        = is_selection
    IMPORTING
      es_selected_request = es_selected_request
      es_selected_task    = es_selected_task.

  p_reqest = es_selected_request-trkorr.

AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_folder.
  DATA:
    title TYPE string.

  title = 'Select target folder'(005).
  CALL METHOD cl_gui_frontend_services=>directory_browse
    EXPORTING
      window_title    = title
    CHANGING
      selected_folder = folder
    EXCEPTIONS
      cntl_error      = 1
      error_no_gui    = 2
      OTHERS          = 3.

  CALL FUNCTION 'CONTROL_FLUSH'
    EXCEPTIONS
      cntl_system_error = 1
      cntl_error        = 2
      OTHERS            = 3.

  p_folder = folder.

AT SELECTION-SCREEN ON p_reqest.
  DATA:
    request_info TYPE stms_wbo_request,
    request_infos TYPE stms_wbo_requests.

  REFRESH request_infos.
  CALL FUNCTION 'TMS_MGR_READ_TRANSPORT_REQUEST'
    EXPORTING
```

```
      iv_request                = p_reqest
      iv_header_only            = 'X'
    IMPORTING
      et_request_infos          = request_infos
    EXCEPTIONS
      read_config_failed        = 1
      table_of_requests_is_empty = 2
      system_not_available      = 3
      OTHERS                    = 4.

  IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  ENDIF.

  CLEAR request_info.
  READ TABLE request_infos INTO request_info INDEX 1.

  IF sy-subrc NE 0
  OR request_info-e070-trkorr IS INITIAL.
    MESSAGE e398(00)
      WITH 'Request'(006) p_reqest
           'not found'(007).
  ELSEIF request_info-e070-trstatus NE 'R'.
    MESSAGE e398(00)
      WITH 'You must release request'(008)
           request_info-e070-trkorr
           'before downloading'(009).
  ENDIF.

START-OF-SELECTION.

  folder = p_folder.

  CONCATENATE p_reqest+3(7) '.' p_reqest(3) INTO trfile.

  CALL FUNCTION 'RSPO_R_SAPGPARAM'
    EXPORTING
      name   = 'DIR_TRANS'
    IMPORTING
      value  = transdir
    EXCEPTIONS
      error  = 0
      OTHERS = 0.

  PERFORM copy_file USING 'cofiles' trfile.
  trfile(1) = 'R'.
```

```
  PERFORM copy_file USING 'data' trfile.
  trfile(1) = 'D'.
  PERFORM copy_file USING 'data' trfile.


*---------------------------------------------------------------------*
*       FORM copy_file                                                *
*---------------------------------------------------------------------*
*  -->  SUBDIR                                                        *
*  -->  FNAME                                                         *
*---------------------------------------------------------------------*
FORM copy_file USING subdir fname.
  DATA:
    gui_filename TYPE string.

  CONCATENATE transdir subdir fname
    INTO filename
    SEPARATED BY p_sepr.

  REFRESH datatab.
  CLEAR flen.
  CALL FUNCTION 'AUTHORITY_CHECK_DATASET'
    EXPORTING
      activity         = sabc_act_read
      filename         = filename
    EXCEPTIONS
      no_authority     = 1
      activity_unknown = 2
      OTHERS           = 3.

  IF sy-subrc <> 0.
    FORMAT COLOR COL_NEGATIVE.
    WRITE: / 'Read access denied. File'(001), filename.
    FORMAT COLOR OFF.
    EXIT.
  ENDIF.

  OPEN DATASET filename FOR INPUT IN BINARY MODE.
  IF sy-subrc NE 0.
    FORMAT COLOR COL_TOTAL.
    WRITE: / 'File open error'(010), filename.
    FORMAT COLOR OFF.
    EXIT.
  ENDIF.
  DO.
    CLEAR len.
    READ DATASET filename INTO datatab LENGTH len.
```


```
  flen = flen + len.
  IF len > 0.
    APPEND datatab.
  ENDIF.
  IF sy-subrc NE 0.
    EXIT.
  ENDIF.
ENDDO.
CLOSE DATASET filename.

CONCATENATE p_folder '\' fname INTO gui_filename.

CALL METHOD cl_gui_frontend_services=>gui_download
  EXPORTING
    bin_filesize            = flen
    filename                = gui_filename
    filetype                = 'BIN'
  CHANGING
    data_tab                = datatab[]
  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
    not_supported_by_gui    = 22
    error_no_gui            = 23
    OTHERS                  = 24.

IF sy-subrc = 0.
  WRITE: / 'File'(002), filename, 'downloaded. Length'(003), flen.
```

```
  ELSE.
    FORMAT COLOR COL_NEGATIVE.
    WRITE: / 'File download error. Filename:'(004), filename.
    FORMAT COLOR OFF.
  ENDIF.

ENDFORM.                    "copy_file
```

Author Bio

In his pre-SAP era Sergei Korolev was a loyal Delphi programmer, developing component libraries and proprietary applications. Since 1999 he has been working as an ABAP developer. Now he is a senior ABAP consultant at IBS Company (http://www.ibs-company.com/) working for various customers..