

Processing Inbound Files

(See the program ZDMR0090 in SF2, as a template program)

- A. On the selection screen, have parameters
 1. for the input file path (for example, named “a_path”), which may be NO-DISPLAY, defined LIKE zunixfiles-path
 2. for the search pattern for the file name (for example, named “a_search”), which may be NO-DISPLAY, defined LIKE zunixfiles-search
 3. for the file name itself (for example, named “a_filnam”) defined LIKE zunixfiles-name
- B. In the INITIALIZATION event,
 1. determine the current environment (by calling Z_DETERMINE_FUNC_AREA)
 2. initialize the a_path parameter with an ABAP statement like “CONCATENATE '/usr/bridges/' environment '/' directory INTO a_path.”
 3. initialize the a_search parameter with an ABAP statement like “CONCATENATE 'd' c_provider c_feed '.'* INTO a_search.” where c_provider is the four character provider code and c_feed is the three-character feed code
- C. Call Z_UT_GET_INPUT_FILE_NAME
 1. Export parameters
 - a) Required export parameters
 - (i) I_FILE_NAME_FIELD (TYPE Z_FILENAME): The name of a selection-screen parameter into which the input file name (just the file name, not the full path) may be typed
 - b) Optional export parameters
 - (i) I_ACCEPTED_PROVIDER (TYPE ZPROVIDER): If present, an exception will be raised unless this value (usually lowercase) is the “provider code” portion of the file name
 - (ii) I_ACCEPTED_FEED (TYPE ZFEED): If present, an exception will be raised unless this value (usually lowercase) is the “feed code” portion of the file name
 - (iii) I_ACCEPTED_EVENT (TYPE TBTCM-EVENTID): If present, an exception will be raised if the program is started by any SAP event other than this one. (The program need not be started by an SAP event, but if it is started by an SAP event, it has to be this one.)

2. Import parameters

- a) E_INPUT_FILE_NAME (TYPE QCHAR40): The name of the input file the program is being asked to process
- b) E_NAME_FROM_EVENT (TYPE CHAR_01): "X" means the program was started from an SAP event, space means it wasn't

D. Call Z_UT_START_OF_INBOUND

1. Export parameters

a) Required export parameters

- (i) I_INPUT_FILE_NAME (TYPE C): The name of the input file passed to the program. (If you used the Z_UT_GET_INPUT_FILE_NAME function module, then this should be the value of the E_INPUT_FILE_NAME parameter returned by that function module)

b) Optional export parameters

- (i) I_DELETE_FILES_OLDER_THAN (TYPE I): If present, a number of days such that all files with this provider and feed code older than this many days (according to the datestamp part of the filename) will be deleted from the archive directory on the application server. If omitted, 7 days is used.
- (ii) I_HANDLING_RULE (TYPE ZDISPOSERULE): The default entry used from the ZUTARCHIVECTRL table is the one with this provider code, feed code, and a blank rule. If some other entry (with different behavior) is required, you can specify a non-blank rule.
- (iii) I_DIRECTORY (TYPE C): By default, input files are expected to be found in the "sabin" directory. If you need to process an input file in another directory (for example, because the user edited the last component of the a_path parameter on the selection screen to process a file in the archive directory), you can pass that directory here

2. Import parameters

- a) E_JOB RUN2_KEY (TYPE JOBRUN2_KEY): The key of the entry in the ZJOBRUN2 table for the input file that you are to process
- b) E_ENCRYPTED_FILENAME (TYPE C): If there was an encrypted file, this is its filename. If there was no encrypted file, this is blank.
- c) E_CONTROL_RECORD (TYPE ZCONTROLFILE): The contents of the control record, if there was a control file, blank if there wasn't a control file.
- d) E_FULL_PATH (TYPE STRING128): The full path of the cleartext data file that you are going to read. This is the value that you would use in ABAP statements

like “OPEN DATASET e_full_path FOR INPUT IN TEXT MODE.” and “READ DATASET e_full_path INTO some-variable.”

3. Table parameters

- a) There are no required table parameters
 - b) Optional table parameters
 - (i) E_LEFTOVER_FILES (TYPE TABLE OF ZJOB RUN2_KEY): If the ZJOB RUN2 table has entries for input files with this provider code and feed code that were not successfully completed, this table will have the keys for those entries in the ZJOB RUN2 table. Depending on the program specifications, it may ignore this leftover files, process any leftover file before processing the new file, or send a warning that leftover incompletely processed files exist.
- E. Open the input file with an ABAP statement such as “OPEN DATASET e_full_path FOR INPUT IN TEXT MODE.”
- F. Read records from the input file (with an ABAP statement such as “READ DATASET e_full_path INTO some-variable LENGTH len.”) and process them. As you do your processing, update the ZJOB RUN2 entry appropriately and accumulate the appropriate totals for comparison with the control totals in the control record.
- G. Close the input file with an ABAP statement such as “CLOSE DATASET e_full_path.”
- H. Update the program’s control total fields in the ZJOB RUN2 entry for this file. If these totals do not equal the ones from the control record, take appropriate action.
- I. Call Z_UT_END_OF_INBOUND
1. Export parameters
 - a) Required export parameters
 - (i) I_JOB RUN2_KEY (TYPE JOB RUN2_KEY): The key of the entry in the ZJOB RUN2 table for the input file that you just processed
 - b) Optional export parameters
 - (i) I_HANDLING_RULE (TYPE ZDISPOSERULE): The default entry used from the ZUTARCHIVECTRL table is the one with this provider code, feed code, and a blank rule. If some other entry (with different behavior) is required, you can specify a non-blank rule.
 - (ii) I_DIRECTORY (TYPE C): By default, input files are expected to be found in the “sapin” directory. If you need to process an input file in another directory (for example, because the user edited the last component of the a_path parameter on the selection screen to process a file in the archive directory), you can pass that directory here

(iii) I_FILE_STATUS (TYPE ZFILSTAT): If present, this file status will be set in the entry in the ZJOB RUN2 table for the file you just processed. If you don't supply this parameter, it is your responsibility to update the status of the ZJOB RUN2 entry yourself.

2. Table parameters

- a) There are no required table parameters
 - b) Optional table parameters
 - (i) E_ERROR_MESSAGES (TYPE TABLE OF ZMAILMSG): The disposition of the file is controlled by an entry in the ZUTARCHIVECTRL table. If the process of handling the file (possibly archiving it and moving or deleting it) encounters any difficulties, text explaining the difficulty will be returned in this table ready to be E-Mailed with Z_SENDMAIL.
- J. (Possibly) call Z_SENDMAIL. If the specifications for the program state that is supposed to send E-Mail advising of problems encountered and problems were encountered (for example, the E_ERROR_MESSAGES table from Z_UT_END_OF_INBOUND is not empty), call Z_SENDMAIL to send a message. If appropriate, you can insert additional text into the E_ERROR_MESSAGES that you got back from Z_UT_END_OF_INBOUND.