

Processing outbound files

(See the program ZDMR0093 in SF2, as a template program)

A. Call Z_UT_START_OF_OUTBOUND

1. Export parameters

a) Required export parameters

- (i) I_PROVIDER (TYPE ZPROVIDER): The provider code for the file that we want to create
- (ii) I_FEED (TYPE ZFEED): The feed code for the file that we want to create

b) Optional export parameters

- (i) I_FILE_CONTENTS_DATE (TYPE D): The date associated with the contents of the file. It will be used in the datestamp part of the file name. If not supplied, it defaults to the current date (i.e. SY-DATUM). For files whose contents are related to a particular day, week, or month, it is customary to pass the last date of that period in this parameter.
- (ii) I_FILE_CONTENTS_TIME (TYPE T): The time associated with the contents of the file. It will be used in the timestamp part of the file name. If not supplied, it defaults to the current time (i.e. SY-UZEIT). For files whose contents are related to a particular day, week, or month, but not any particular time of day, it is customary to pass "000000" in this parameter.
- (iii) I_DIRECTORY (TYPE C): By default, output files are created in the "sapout" directory. If you need to create an output file in another directory, you can pass that directory here
- (iv) 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.
- (v) 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.

2. Import parameters

- a) E_JOB RUN2_KEY (TYPE JOBRUN2_KEY): The key of the entry in the ZJOBRUN2 table for the output file that you are to create
- b) E_FULL_PATH (TYPE STRING128): The full path of the cleartext data file that you are going to write. This is the value that you would use in ABAP statements

like “OPEN DATASET e_full_path FOR OUTPUT IN TEXT MODE.” and
“TRANSFER some-variable TO e_full_path.”

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 output 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.

B. Open the output file with an ABAP statement such as “OPEN DATASET e_full_path FOR OUTPUT IN TEXT MODE.”

C. Write records to the output file (with an ABAP statement such as “TRANSFER some-variable TO e_full_path.”). As you do your processing, update the ZJOB RUN2 entry appropriately and accumulate the appropriate totals for writing to the control record.

D. Close the output file with an ABAP statement such as “CLOSE DATASET e_full_path.”

E. Update the program’s control total fields in the ZJOB RUN2 entry for this file.

F. Call Z_UT_END_OF_OUTBOUND

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 output file that you just created

b) Optional export parameters

(i) I_CONTROL_RECORD (TYPE ZCONTROLFILE): You should fill in the record count field (it is the number of lines in the cleartext data file that you wrote) and as many of control totals 1 through 4 as you use. All fields must be entirely numeric, right-aligned, and zero filled on the left. No field should contain a sign, a comma, or an explicit decimal point. Control totals 1 through 4 have an implicit decimal point between the second and third digits from the right. All unused fields should be filled with zeros. The byte count field should be left empty. It will automatically be filled with the byte count of the encrypted file (if there is one) or the cleartext file (if there is no encrypted file). If this parameter is not supplied, no control file will be written.

- (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, output files are expected to be created in the "sapout" directory. If you need to create an output file in another directory, you can pass that directory here
- (iv) 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 created. 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 encrypting it, FTPing it, archiving it, or 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.
- G. (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_OUTBOUND 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_OUTBOUND.