

```

*&-----*
*& Report   Z_NSI_CONTR_KEY_KSSS
*&
*&-----*
*&
*& VMerkurev массовая отправка ключей контрагентов
*& last update: 18.09.20
*&
*&-----*
REPORT Z_NSI_CONTR_KEY_KSSS.

TABLES: lfal, knal.

TYPES:
begin of lfal_str,
    lifnr TYPE lfal-lifnr,
    konzs TYPE lfal-konzs,
    sperr TYPE lfal-sperr,
    loevm TYPE lfal-loevm,
end of lfal_str.
TYPES:
begin of knal_str,
    kunnr TYPE knal-kunnr,
    konzs TYPE knal-konzs,
    sperr TYPE knal-sperr,
    loevm TYPE knal-loevm,
end of knal_str.

DATA:
    cl_fdbk TYPE REF TO ZCO_SI_FEEDBACK_TO_KSSS_AO
    , ls_fdbk TYPE ZMT_FEEDBACK_TO_KSSS
    , ls_fdbk_control TYPE ZDT_FEEDBACK_TO_KSSS
    , ls_fdbk_system TYPE ZDT_FEEDBACK_TO_KSSS_SYSTEM
    , ls_fdbk_data TYPE ZDT_FEEDBACK_TO_KSSS_DATA
    , ls_fdbk_ext_key TYPE ZDT_FEEDBACK_TO_KSSS_EXTERNAL
    , l_time TYPE string
    , l_date TYPE string
    , l_lifnr TYPE lfal-lifnr
    , l_kunnr TYPE lfal-kunnr
    , l_konzs TYPE lfal-konzs
    , l_count_keys TYPE integer
    , lt_keys_lfa TYPE STANDARD TABLE OF lfal_str
    , lt_keys_kna TYPE STANDARD TABLE OF knal_str
    .

FIELD-SYMBOLS <lt_keys_lfa> TYPE lfal_str.
FIELD-SYMBOLS <lt_keys_kna> TYPE knal_str.

SELECTION-SCREEN BEGIN OF BLOCK b-01 WITH FRAME title text-001.
select-options: lifnr for lfal-lifnr.
select-options: kunnr for lfal-kunnr.
parameter:      p_max type integer DEFAULT 1 obligatory.
SELECTION-SCREEN END OF BLOCK b-01.

START-OF-SELECTION.

* Собираем данные

select lifnr konzs sperr loevm
into table lt_keys_lfa

```

```

from lfal
where lifnr in lifnr
.

select kunnr konzs sperr loevm
into table lt_keys_kna
from knal
where kunnr in kunnr
.

* Согласно постановке отдельно обрабатывать lifnr и kunnr

if lifnr is not INITIAL.

* очищаем все данные
l_count_keys = 0.
clear ls_fdbk.
clear ls_fdbk_control.
clear ls_fdbk_system.

* заполняем то, что нужно один раз - можно не чистить
ls_fdbk_system-system_id = 'PERERABOTKA'.
l_time = sy-uzeit+0(2) && ':' && sy-uzeit+2(2) && ':' && sy-uzeit+4(2).
l_date = sy-datum+0(4) && ':' && sy-datum+4(2) && ':' && sy-datum+6(2).
CONCATENATE l_date l_time INTO ls_fdbk_system-message_date
SEPARATED BY space.

loop at lt_keys_lfa ASSIGNING <lt_keys_lfa>.

    clear ls_fdbk_data.
    clear ls_fdbk_ext_key.

    * Добавляем только если поля признака блокировки (sperr), удаления
    (loevm) пустые, а KCCC не пустой
    if <lt_keys_lfa>-sperr is initial
        and <lt_keys_lfa>-loevm is initial
        and <lt_keys_lfa>-konzs is NOT initial.
    * Заполняем DATA и external_key
        ls_fdbk_data-ksss_id = <lt_keys_lfa>-konzs.
        ls_fdbk_data-ksss_catalog = 'Catalog_CONTRAGENT'. "константа
        if <lt_keys_lfa>-lifnr is not initial.
            ls_fdbk_ext_key-key = <lt_keys_lfa>-lifnr.
            ls_fdbk_ext_key-object = 'CREDITOR'.
            ls_fdbk_ext_key-resp_user = sy-uname.
            append ls_fdbk_ext_key to ls_fdbk_data-external_key.
        endif.
        * if <lt_keys_lfa>-kunnr is not initial.
        * ls_fdbk_ext_key-key = <lt_keys_lfa>-kunnr.
        * ls_fdbk_ext_key-object = 'DEBITOR'.
        * append ls_fdbk_ext_key to ls_fdbk_data-external_key.
        * endif.
        append ls_fdbk_data to ls_fdbk_control-data.

        l_count_keys = l_count_keys + 1.
    endif.

    * Отправляем пакет с p_max DATA записей
    if l_count_keys >= p_max.
        ls_fdbk_control-system = ls_fdbk_system.
        ls_fdbk-mt_feedback_to_ksss = ls_fdbk_control.
    endif.

```

```

*      передача прокси
      CREATE OBJECT cl_fdbk.
      cl_fdbk->si_feedback_to_ksss_ao( EXPORTING output = ls_fdbk
                                      EXCEPTIONS OTHERS = 1 ).

*      очищаем все после передачи
      l_count_keys = 0.
      clear cl_fdbk.
      clear ls_fdbk.
      clear ls_fdbk_control.

      endif.

endloop.

*      отправка последней партии
if l_count_keys <> 0.
  ls_fdbk_control-system = ls_fdbk_system.
  ls_fdbk-mt_feedback_to_ksss = ls_fdbk_control.
*      передача прокси
  CREATE OBJECT cl_fdbk.
  cl_fdbk->si_feedback_to_ksss_ao( EXPORTING output = ls_fdbk
                                  EXCEPTIONS OTHERS = 1 ).

  endif.

  COMMIT WORK.
  MESSAGE s094(00).

endif.

if kunnr is not INITIAL.
*      очищаем все данные
  l_count_keys = 0.
  clear ls_fdbk.
  clear ls_fdbk_control.
  clear ls_fdbk_system.

*      заполняем то, что нужно один раз - можно не чистить
  ls_fdbk_system-system_id = 'PERERABOTKA'.
  l_time = sy-uzeit+0(2) && ':' && sy-uzeit+2(2) && ':' && sy-uzeit+4(2).
  l_date = sy-datum+0(4) && ':' && sy-datum+4(2) && ':' && sy-datum+6(2).
  CONCATENATE l_date l_time INTO ls_fdbk_system-message_date
    SEPARATED BY space.

  loop at lt_keys_kna ASSIGNING <lt_keys_kna>.

*      Добавляем только если поля признака блокировки (sperr), удалениз (loevm)
      пустые а KCCC не пустой
      if <lt_keys_kna>-sperr is initial
        and <lt_keys_kna>-loevm is initial
        and <lt_keys_kna>-konzs is NOT initial.
        clear ls_fdbk_data.
        clear ls_fdbk_ext_key.

*      Заполняем DATA и external_key
        ls_fdbk_data-ksss_id = <lt_keys_kna>-konzs.
        ls_fdbk_data-ksss_catalog = 'Catalog_CONTRAGENT'. "константа
        if <lt_keys_kna>-kunnr is not initial.
          ls_fdbk_ext_key-key = <lt_keys_kna>-kunnr.

```

```

        ls_fdbk_ext_key-object = 'DEBITOR'.
        ls_fdbk_ext_key-resp_user = sy-uname.
        append ls_fdbk_ext_key to ls_fdbk_data-external_key.
    endif.
    append ls_fdbk_data to ls_fdbk_control-data.

    l_count_keys = l_count_keys + 1.
endif.

*   Отправляем пакет с p_max DATA записей
if l_count_keys >= p_max.
    ls_fdbk_control-system = ls_fdbk_system.
    ls_fdbk-mt_feedback_to_ksss = ls_fdbk_control.

*   передача прокси
    CREATE OBJECT cl_fdbk.
    cl_fdbk->si_feedback_to_ksss_ao( EXPORTING output = ls_fdbk
                                    EXCEPTIONS OTHERS = 1 ).

*   очищаем все после передачи
    l_count_keys = 0.
    clear cl_fdbk.
    clear ls_fdbk.
    clear ls_fdbk_control.

endif.

endloop.

*   отправка последней партии
if l_count_keys <> 0.
    ls_fdbk_control-system = ls_fdbk_system.
    ls_fdbk-mt_feedback_to_ksss = ls_fdbk_control.
*   передача прокси
    CREATE OBJECT cl_fdbk.
    cl_fdbk->si_feedback_to_ksss_ao( EXPORTING output = ls_fdbk
                                    EXCEPTIONS OTHERS = 1 ).

endif.

COMMIT WORK.
MESSAGE s094(00).

endif.

---

*{ VMERKUREV 29.09.2020
*** Проверка правильности заполнения заказа
DATA:
*   l_objek TYPE ausp-objek
    l_atinn TYPE ausp-atinn
    , l_atwrt TYPE ausp-atwrt
    , l_klart TYPE ausp-klart
    , lt_ausep TYPE STANDARD TABLE OF ausp
    , ls_ausep LIKE LINE OF lt_ausep
    .
FIELD-SYMBOLS:
    <lt_drad> LIKE LINE OF lt_drad.

*   считываем MM000116_SF
CLEAR: l_devstatus.

```

```

CALL FUNCTION 'ZL_DEVSTATUS_GET'
EXPORTING
  i_devname = 'MM000116_SF'
  i_bukrs   = ms_data_header-bukrs
IMPORTING
  e_status  = l_devstatus.

* отработывает только для транзакции при наличии статуса
IF ( sy-tcode = 'ME21N' OR sy-tcode = 'ME22N' ) and l_devstatus IS NOT
INITIAL.

* 1. Находим прикрепленные документы - lt_drad
CALL METHOD im_header->get_items
RECEIVING
  re_items = lt_items.

LOOP AT lt_items INTO lo_items. " цикл по заказам
  CLEAR ls_mepoitem.
  CALL METHOD lo_items-item->get_data " получаем заказ
  RECEIVING
    re_data = ls_mepoitem.

  IF ls_mepoitem-loekz <> 'L'. " не обрабатывать заказы помеченные на
удаление

    CLEAR: l_objky.

    IF ls_mepoitem-ebeln IS INITIAL.
      l_objky = ls_mepoitem-ebelp.
      SHIFT l_objky BY 10 PLACES RIGHT.
    ELSE.
      CONCATENATE ls_mepoitem-ebeln ls_mepoitem-ebelp
        INTO l_objky.
    ENDIF.

    CLEAR: lt_drad.

    CALL FUNCTION 'DOKUMENTE_ZU_OBJEKT' " получаем документы заказа
    EXPORTING
      key                = l_objky
      objekt              = 'EKPO'
      check_buffer_and_db = '' " только из буфера
    TABLES
      doktab              = lt_drad
    EXCEPTIONS
      kein_dokument       = 1
      OTHERS               = 2.

    LOOP AT lt_drad INTO ls_drad. " цикл по документам заказа

* 2.1. Проводим проверку 1
    IF ls_drad-dokar(2) <> 'DG' or ls_drad-dokvr <> 00.
      MESSAGE ID 'ZL_MM000247' TYPE 'E' NUMBER '014' WITH ls_mepoitem-
ebelp ls_drad-dokar ls_drad-doknr.
    ENDIF.

* 2.2. Проводим проверку 2
    l_objek = ls_drad-dokar && ls_drad-doknr && '%' && ls_drad-doktl &&
ls_drad-dokvr.
    l_atinn = 0000000850.

```

```

l_atwrt = 'PIT'.
SELECT SINGLE klart
  FROM ausp
  INTO l_klart
  WHERE objek like l_objek
        AND atinn = l_atinn
        AND atwrt = l_atwrt.
IF sy-subrc = 0.
  MESSAGE ID 'ZL_MM000247' TYPE 'E' NUMBER '015' WITH ls_mepoitem-
ebelp ls_drad-dokar ls_drad-doknr.
ENDIF.

ENDLOOP. " lt_drad

ENDIF. " ls_mepoitem-loekz <> 'L'

ENDLOOP. " lt_items

ENDIF.
*} VMERKUREV

```