

Class: ZCL_ZAS_CONTRACT_VERSI_DPC_EXT

Status: Active

Inh. from: ZCL_ZAS_CONTRACT_VERSI_DPC

Attributes

Description: Data Provider Secondary Class

Instantiation: Public

Not final

Not released

Fixed pt.arithmetic

Category: General Object Type

Package: ZSD

Original lang.: EN

Created by: SG0303539

Created on: 04/01/2019

Last changed on: 04/03/2019

Methods

Redefined Methods

/IWBEF/IF_MGW_APPL_SRV_RUNTIME~EXECUTE_ACTION

```
METHOD /iwbef/if_mgw_appl_srv_runtime~execute_action.
**TRY.
**CALL METHOD SUPER->/IWBEF/IF_MGW_APPL_SRV_RUNTIME~EXECUTE_ACTION
**  EXPORTING
**    iv_action_name           =
**    it_parameter            =
**    io_tech_request_context =
**  IMPORTING
**    er_data                  =
**  .
** CATCH /iwbef/cx_mgw_busi_exception .
** CATCH /iwbef/cx_mgw_tech_exception .
**ENDTRY.
DATA lt_key_tab                TYPE          /iwbef/t_mgw_name_value_pair.
DATA lo_function_import        TYPE REF TO object.
DATA ls_status                 TYPE          zcl_zas_contract_versi_mpc=>ts_status.
DATA: lv_ship_to               TYPE kunnr,
      lv_rel_pkg_atinn         TYPE atinn,
      lv_rel_pkg               TYPE atnam,
      lv_rel_pkg_name          TYPE klschl,
      lv_version                TYPE atwrt.
**<< Begin of insert by vijay on 07/10/2020 - Redesign
DATA: lv_vc_ver_atinn          TYPE atinn,
      lv_vc_rp_id_atinn        TYPE atinn,
      lv_rp_id                 TYPE matnr.
**>> End of insert by vijay on 07/10/2020
**Structure declaration
TYPES: BEGIN OF st_final_tab,
        contract                TYPE vbeln_va,
```

```

        item                                TYPE posnr_va,
        header_enddate                      TYPE vndat_veda,
        item_enddate                        TYPE vndat_veda,
        header_insddate                     TYPE vidat_veda,
        po_description                       TYPE ktext_v,
        product                             TYPE matnr,
        product_hierarchy                   TYPE prodh_d,
        header_objectno                      TYPE objko,
        item_objectno                       TYPE objpo,
        material_desc                        TYPE arktx,
        ship_to                             TYPE kunnr,
        ship_to_customer                     TYPE name1_gp,
        ship_to_2ltr                         TYPE sortl,
        sold_to_customer                     TYPE name1_gp,
        sold_to_2ltr                         TYPE sortl,
        class_desc                           TYPE klschl,
*      prod_characteristic_name              TYPE atnam,
*      prod_characteristic_name_desc         TYPE atbez,
*      prod_characteristic_value             TYPE atwrt,
*      prod_characteristic_desc              TYPE atwtb,
        vc_prod_characteristic_name          TYPE atnam,
        vc_charateristic_desc                TYPE atbez,
        vc_prod_characteristic_value         TYPE atwrt,
        vc_description                       TYPE atwtb,
        vc_config                            TYPE cuobj,
        prodh1                              TYPE prodh_d,
        vtext1                              TYPE bezei40,
        prodh2                              TYPE prodh_d,
        vtext2                              TYPE bezei40,
        prodh3                              TYPE prodh_d,
        vtext3                              TYPE bezei40,
        prodh4                              TYPE prodh_d,
        vtext4                              TYPE bezei40,
        delivery_method                      TYPE atwtb,
        flag(1)                             TYPE c,
        END OF st_final_tab.
**Internal table and Workarea declaration
        DATA: lt_final_tab TYPE TABLE OF st_final_tab,
              ls_final_tab TYPE          st_final_tab.
        DATA: ls_inst_rec TYPE ibco2_instance_rec2,
              ls_values  TYPE ibco2_value_rec.
        DATA: lt_value TYPE TABLE OF string,
              ls_value TYPE string,
              lv_value TYPE string.
        DATA lt_activity_groups TYPE STANDARD TABLE OF bapiagr.
        DATA lt_return          TYPE STANDARD TABLE OF bapiret2.
        DATA lv_valid_action     TYPE          abap_bool VALUE abap_false.
**Structure for Updating the log
        DATA: ls_history TYPE zas_verupd_log.
**file
*      DATA: lv_file_name TYPE string,
*      lv_file_data TYPE string.
        DATA: lt_ship_to TYPE /iwbsp/t_cod_select_options,
              ls_ship_to TYPE /iwbsp/s_cod_select_option.

```

```
**Constant declataion
  CONSTANTS: con_obtyp_vbk   TYPE jsto-obtyp VALUE 'VBK',
             con_obtyp_vbp   TYPE jsto-obtyp VALUE 'VBP',
             con_del_method  TYPE atnam VALUE 'ZAS_PRODUCT_DELIVERY_METHOD',
             con_agr_name    TYPE agr_name VALUE 'ZS_AS_VERSION_UPDATE',
**<< Begin of insert by vijay on 04/08/2020- For contract header status CSDU/CCU
             con_csd_u       TYPE j_txt04 VALUE 'CSDU',
             con_ccu         TYPE j_txt04 VALUE 'CCU'.
**>> End of insert by vijay on 04/08/2020
**<< Begin of insert by vijay on 07/10/2020 - Redesign
  CONSTANTS: con_vc_ver_char  TYPE cabn-atnam VALUE 'ZAS_RP_CNTR_VERSION',
             con_vc_rp_id_char TYPE cabn-atnam VALUE 'ZAS_RP_ID'.
**>> End of insert by vijay on 07/10/2020
**file
*           con_path          TYPE string VALUE
'boomi/outbound/I0447_AS_Asset_Mass_Updates/',
*           con_delimit       TYPE c VALUE ','.
**Field Symbol Declaration
  FIELD-SYMBOLS <ls_parameter>      TYPE /iwbep/s_mgw_name_value_pair.
  "Retrieve key/value of incoming request parameters:
  lt_key_tab = io_tech_request_context->get_parameters( ).
  CASE iv_action_name.
    WHEN 'UpdateContractVersion'.
      "Retrieve user role data
      CALL FUNCTION 'BAPI_USER_GET_DETAIL'
        EXPORTING
          username      = sy-uname
        TABLES
          activitygroups = lt_activity_groups
          return         = lt_return.
**Check for the update role
  READ TABLE lt_activity_groups WITH KEY agr_name = con_agr_name TRANSPORTING
NO FIELDS.
  IF sy-subrc EQ 0.
    lv_valid_action = abap_true.
  ENDIF.
  IF lv_valid_action EQ abap_true.
**Get Release Package
    READ TABLE lt_key_tab INTO DATA(ls_key_tab) WITH KEY name = 'REL_PKG_NAME'.
    IF sy-subrc = 0 AND ls_key_tab-value IS NOT INITIAL.
      CLEAR: lv_rel_pkg_name.
      lv_rel_pkg_name = ls_key_tab-value.
**<< Begin of insert by vijay on 07/10/2020 - Redesign
**Version Characteristic (Internal Number)
      CLEAR: lv_vc_ver_atinn.
      SELECT SINGLE atinn
                INTO   lv_vc_ver_atinn
                FROM   cabn
                WHERE  atnam EQ  con_vc_ver_char.
      IF sy-subrc = 0.
        ENDIF.
**Release Package ID Characteristic(Internal Number)
      CLEAR: lv_vc_rp_id_atinn.
      SELECT SINGLE atinn
```

```

        INTO lv_vc_rp_id_atinn
        FROM cabn
        WHERE atnam EQ con_vc_rp_id_char.
    IF sy-subrc = 0.
    ENDIF.
**Release Package ID
    CLEAR: lv_rp_id.
    SELECT SINGLE matnr_rpg
        INTO lv_rp_id
        FROM zasv_rpg_data
        WHERE rpg_desc EQ lv_rel_pkg_name.
    IF sy-subrc = 0.
    ENDIF.
**>> End of insert by vijay on 07/10/2020
**<< Begin of comment by vijay on 07/10/2020 - Redesign
*
    SELECT SINGLE internal_num
        FROM zas_clas_rel_pkg
        INTO lv_rel_pkg_atinn
        WHERE class_desc EQ lv_rel_pkg_name.
**>> End of comment by vijay on 07/10/2020
*
    CLEAR: lv_rel_pkg_name.
    lv_rel_pkg_name = ls_key_tab-value.
**Get Release pkg internal number
*
    CLEAR: lv_rel_pkg_atinn.
    SELECT SINGLE atinn
        INTO lv_rel_pkg_atinn
        FROM cabnt
        WHERE atbez EQ lv_rel_pkg_name
        AND spras EQ 'E'.
    IF sy-subrc = 0.
    ENDIF.
ENDIF.
**Get the Ship to
    CLEAR: ls_key_tab.
    READ TABLE lt_key_tab INTO ls_key_tab WITH KEY name = 'SHIP_TO'.
    IF sy-subrc = 0 AND ls_key_tab-value IS NOT INITIAL.
    CLEAR: lv_value.
    lv_value = ls_key_tab-value.
**Split the String value by Comma and construct as a Range
    SPLIT lv_value AT ',' INTO TABLE lt_value.
    LOOP AT lt_value INTO ls_value.
        CONDENSE ls_value.
    CLEAR: lv_ship_to.
**Convert the Customer into Internal SAP Format
    CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
        EXPORTING
            input = ls_value
        IMPORTING
            output = lv_ship_to.
**Move the Values as a range
    ls_ship_to-sign = 'I'.
    ls_ship_to-option = 'EQ'.
    ls_ship_to-low = lv_ship_to.
    APPEND ls_ship_to TO lt_ship_to.
```

```
        CLEAR: ls_ship_to.
      ENDLOOP.
    ENDIF.
**Get Version
    CLEAR:ls_key_tab.
    READ TABLE lt_key_tab INTO ls_key_tab WITH KEY name = 'CURR_PROD_VER'.
    IF sy-subrc = 0 AND ls_key_tab-value IS NOT INITIAL.
      CLEAR: lv_version.
      lv_version = ls_key_tab-value.
      *      CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
      *      EXPORTING
      *          input  = ls_key_tab-value
      *      IMPORTING
      *          output = lv_version.
    ENDIF.
**Get the data from View zas_con_prod_rel (As it contains all the data)
    SELECT
      contract
      item
      header_enddate
      item_enddate
      header_insddate
      po_description
      product
      product_hierarchy
      header_objectno
      item_objectno
      material_desc
      ship_to
      ship_to_customer
      ship_to_2ltr
      sold_to_customer
      sold_to_2ltr
      class_desc
      *      prod_characteristic_name
      *      prod_characteristic_name_desc
      *      prod_characteristic_value
      *      prod_characteristic_desc
      *      vc_prod_characteristic_name
      *      vc_charateristic_desc
      *      vc_prod_characteristic_value
      *      vc_description
      *      vc_config
    FROM zas_con_prod_rel
    INTO TABLE lt_final_tab
    WHERE ship_to IN lt_ship_to
**      AND ship_to IN lr_ship_to
**      AND ship_to_2ltr IN lr_ship_to_2ltr
      *      AND prod_characteristic_name_desc EQ lv_rel_pkg_name
      *      AND class_desc EQ lv_rel_pkg_name.
**      AND contract IN lr_contract
      *      AND ( header_enddate >= sy-datum OR header_enddate EQ '00000000').
"Commented by 303638
      *      AND ( item_enddate >= sy-datum OR item_enddate EQ '00000000').
```

```
IF sy-subrc = 0.
**Get header status details only for CRX(To ignore the contract)
  SELECT a~objnr,
         b~stat,
         c~txt04
         INTO TABLE @DATA(lt_header_status)
         FROM jsto AS a
         INNER JOIN jest AS b
         ON a~objnr = b~objnr
         INNER JOIN tj30t AS c
         ON a~stsma = c~stsma
         AND b~stat = c~estat
         FOR ALL ENTRIES IN @lt_final_tab
         WHERE a~objnr EQ @lt_final_tab-header_objectno
         AND a~obtyp EQ @con_obtyp_vbk
         AND b~stat LIKE 'E%'
         AND b~inact EQ ''
         AND c~spras EQ 'E'
         AND c~txt04 EQ 'CRX'.
IF sy-subrc = 0.
ENDIF.
**Get Item status details only for (DUP, PDX, PRX, PCA)(To ignore the contract)
  SELECT a~objnr,
         b~stat,
         c~txt04
         INTO TABLE @DATA(lt_item_status)
         FROM jsto AS a
         INNER JOIN jest AS b
         ON a~objnr = b~objnr
         INNER JOIN tj30t AS c
         ON a~stsma = c~stsma
         AND b~stat = c~estat
         FOR ALL ENTRIES IN @lt_final_tab
         WHERE a~objnr EQ @lt_final_tab-item_objectno
         AND a~obtyp EQ @con_obtyp_vbp
         AND b~stat LIKE 'E%'
         AND b~inact EQ ''
         AND c~spras EQ 'E'
         AND c~txt04 IN ( 'DUP' , 'PDX' , 'PRX' , 'PCA', 'PCE' , 'PEX' ).
"Insert PCE & PEX by 303638
IF sy-subrc = 0.
ENDIF.
**Get Coontrat item delivery method details
  SELECT sales_doc,
         item,
         material,
         description
         FROM zas_con_prod_ver
         INTO TABLE @DATA(lt_delivery_method)
         FOR ALL ENTRIES IN @lt_final_tab
         WHERE sales_doc EQ @lt_final_tab-contract
         AND item EQ @lt_final_tab-item
         AND material EQ @lt_final_tab-product
         AND characteristic_name EQ @con_del_method.
```

```
        IF sy-subrc = 0.
        ENDIF.
**<< Begin of insert by vijay on 04/08/2020 - To consider only the contracts which is
having header status as CSDU/CCU.
**Get the header status details for CDSU and CCU
        SELECT a~objnr,
               b~stat,
               c~txt04
        INTO TABLE @DATA(lt_hdr_stat_csd)
        FROM jsto AS a
        INNER JOIN jest AS b
        ON a~objnr = b~objnr
        INNER JOIN tj30t AS c
        ON a~stsma = c~stsma
        AND b~stat = c~estat
        FOR ALL ENTRIES IN @lt_final_tab
        WHERE a~objnr EQ @lt_final_tab-header_objectno
        AND a~obtyp EQ @con_obtyp_vbk
        AND b~stat LIKE 'E%'
        AND b~inact EQ @abap_false
        AND c~spras EQ 'E'
        AND c~txt04 IN ( @con_csd , @con_ccu ).
        IF sy-subrc = 0.
        ENDIF.
**>> End of insert by vijay on 04/08/2020
        ENDIF.
        LOOP AT lt_final_tab ASSIGNING FIELD-SYMBOL(<fs_final>).
**Check the header status. If CRX , then set the delete flag to ignore the contract.
        READ TABLE lt_header_status INTO DATA(ls_header_status) WITH KEY objnr =
<fs_final>-header_objectno.
        IF sy-subrc = 0.
            <fs_final>-flag = 'X'.
        ENDIF.
**<< Begin of insert by vijay on 04/08/2020 - For contract header status CSDU/CCU.
****Check the header status. If CSDU or CCU then consider those contract. If not then
set the delete flag to ignore the contract.
        READ TABLE lt_hdr_stat_csd INTO DATA(ls_hdr_stat_csd) WITH KEY objnr =
<fs_final>-header_objectno.
        IF sy-subrc NE 0.
            <fs_final>-flag = 'X'.
        ENDIF.
**>> End of insert by vijay on 04/08/2020
**Check the item status. If its falls under 'DUP' , 'PDX' , 'PRX' , 'PCA' , 'PCE' ,
'PEX' , then set the delete flag to ignore the c
        READ TABLE lt_item_status INTO DATA(ls_item_status) WITH KEY objnr =
<fs_final>-item_objectno.
        IF sy-subrc = 0.
            <fs_final>-flag = 'X'.
        ENDIF.
        IF <fs_final>-flag IS INITIAL AND <fs_final>-product_hierarchy IS NOT
INITIAL.
**<< Begin of comment by vijay on 09/03/2021 - To incorporate new prod hier design
*
        <fs_final>-prodh1 = <fs_final>-product_hierarchy+0(5).
*
        <fs_final>-prodh2 = <fs_final>-product_hierarchy+0(9).
```

```
*          <fs_final>-prodh3 = <fs_final>-product_hierarchy+0(14).
*          <fs_final>-prodh4 = <fs_final>-product_hierarchy+0(18).
**>> End of comment by vijay on 09/03/2021
**<< Begin of insert by vijay on 09/03/2021 - To incorporate new prod hier design
          <fs_final>-prodh1 = <fs_final>-product_hierarchy+0(2).
          <fs_final>-prodh2 = <fs_final>-product_hierarchy+0(5).
          <fs_final>-prodh3 = <fs_final>-product_hierarchy+0(9).
          <fs_final>-prodh4 = <fs_final>-product_hierarchy+0(14).
**>> End of insert by vijay on 09/03/2021
      ENDIF.
**Move Delivery method to final table
      IF <fs_final>-flag IS INITIAL.
        READ TABLE lt_delivery_method INTO DATA(ls_delivery_method) WITH KEY
sales_doc = <fs_final>-contract
item = <fs_final>-item
material = <fs_final>-product.
        IF sy-subrc = 0.
          <fs_final>-delivery_method = ls_delivery_method-description.
        ENDIF.
      ENDIF.
    ENDLOOP.
**Delete the records where Header and item status not met our condition
    DELETE lt_final_tab WHERE flag = 'X'.
    SORT lt_final_tab BY product contract item.
    IF lt_final_tab[] IS NOT INITIAL.
***Get the product hierarhy description
*      SELECT prodh,
*          vtext
*          INTO TABLE @DATA(lt_suite)
*          FROM t179t
*          FOR ALL ENTRIES IN @lt_final_tab
*          WHERE prodh EQ @lt_final_tab-prodh1.
*      IF sy-subrc = 0.
*      ENDIF.
*
*      SELECT prodh,
*          vtext
*          INTO TABLE @DATA(lt_sol_family)
*          FROM t179t
*          FOR ALL ENTRIES IN @lt_final_tab
*          WHERE prodh EQ @lt_final_tab-prodh2.
*      IF sy-subrc = 0.
*      ENDIF.
*
*      SELECT prodh,
*          vtext
*          INTO TABLE @DATA(lt_solution)
*          FROM t179t
*          FOR ALL ENTRIES IN @lt_final_tab
*          WHERE prodh EQ @lt_final_tab-prodh3.
*      IF sy-subrc = 0.
*      ENDIF.
**For Capability
      SELECT prodh,
```



```
        vtext
        INTO TABLE @DATA(lt_capability)
        FROM t179t
        FOR ALL ENTRIES IN @lt_final_tab
        WHERE prodh EQ @lt_final_tab-prodh4.
    IF sy-subrc = 0.
    ENDIF.

***file
***Create the file name
*
*   CLEAR: lv_file_name.
*   CONCATENATE con_path
*               'version_update_'
*               sy-datum
*               '_'
*               sy-uzeit
*               '.csv' INTO lv_file_name.
***& Open the Output file
*
*   OPEN DATASET lv_file_name FOR OUTPUT IN TEXT MODE ENCODING DEFAULT.
**Now we got all the records so we can go ahead and update it.
*   LOOP AT lt_final_tab ASSIGNING <fs_final>.
**Get the existing values
*   CLEAR:ls_inst_rec , ls_values.
*   CALL FUNCTION 'CUCB_GET_SINGLE_INSTANCE'
*       EXPORTING
*           instance                        = <fs_final>-vc_config
*       IMPORTING
*           instance_rec                   = ls_inst_rec
*       EXCEPTIONS
*           invalid_instance               = 1
*           instance_is_a_classification = 2
*           OTHERS                         = 3.
**<< Begin of comment by vijay on 07/10/2020 - Redesign
*
*   READ TABLE ls_inst_rec-values ASSIGNING FIELD-SYMBOL(<lfs_values>)
*       WITH KEY atinn = lv_rel_pkg_atinn.
**>> End of comment by vijay on 07/10/2020
**<< Begin of insert by vijay on 07/10/2020 - Redesign
*   READ TABLE ls_inst_rec-values ASSIGNING FIELD-SYMBOL(<lfs_values>)
*       WITH KEY atinn = lv_vc_ver_atinn.
**>> End of insert by vijay on 07/10/2020
*   IF sy-subrc = 0.
*       IF <lfs_values>-atwrt = lv_version.
*           CONTINUE.
*       ELSE.
*           <lfs_values>-atwrt = lv_version.
*       ENDIF.
*   ELSE.
**<< Begin of comment by vijay on 07/10/2020 - Redesign
*
*   ls_values-atinn = lv_rel_pkg_atinn.
**>> End of comment by vijay on 07/10/2020
**<< Begin of insert by vijay on 07/10/2020 - Redesign
*   ls_values-atinn = lv_vc_ver_atinn.
**<< Begin of insert by vijay on 07/10/2020 - Redesign
*   ls_values-atwrt = lv_version.
*   ls_values-atcod = 1.
```

```
        APPEND ls_values TO ls_inst_rec-values.
        CLEAR: ls_values.
**<< Begin of insert by vijay on 07/10/2020 - Redesign
**For Updating Release Package ID
        ls_values-atinn = lv_vc_rp_id_atinn.
        ls_values-atwrt = lv_rp_id.
        ls_values-atcod = 1.
        APPEND ls_values TO ls_inst_rec-values.
        CLEAR: ls_values.
**>> End of insert by vijay on 07/10/2020
    ENDIF.
**Set the Vc value
    CALL FUNCTION 'CUCB_SET_VALUES_TO_INSTANCE'
      EXPORTING
        instance_rec           = ls_inst_rec
      EXCEPTIONS
        invalid_instance       = 1
        instance_is_a_classification = 2
        OTHERS                  = 3.
    IF sy-subrc = 0.
**Save to DB
        CALL FUNCTION 'CUCB_CONFIGURATION_TO_DB'
          EXPORTING
            root_instance       = <fs_final>-vc_config
            root_object         = ls_inst_rec-owner
          EXCEPTIONS
            invalid_instance     = 1
            invalid_root_instance = 2
            no_changes           = 3
            already_registered_for_update = 4
            instance_is_a_classification = 5
            OTHERS                = 6.
        IF sy-subrc = 0.
            COMMIT WORK AND WAIT.
        ENDIF.
    ENDIF.
***  Get Suite Description
*      READ TABLE lt_suite INTO DATA(ls_suite) WITH KEY prodh =
<fs_final>-prodh1.
*      IF sy-subrc = 0.
*          <fs_final>-vtext1 = ls_suite-vtext.
*      ENDIF.
***  Get Solution Family Description
*      READ TABLE lt_sol_family INTO DATA(ls_sol_family) WITH KEY prodh =
<fs_final>-prodh2.
*      IF sy-subrc = 0.
*          <fs_final>-vtext2 = ls_sol_family-vtext.
*      ENDIF.
***  Get Solution Description
*      READ TABLE lt_solution INTO DATA(ls_solution) WITH KEY prodh =
<fs_final>-prodh3.
*      IF sy-subrc = 0.
*          <fs_final>-vtext3 = ls_solution-vtext.
*      ENDIF.
```

```
** Get Capability Description
    READ TABLE lt_capability INTO DATA(ls_capability) WITH KEY prodh =
<fs_final>-prodh4.
    IF sy-subrc = 0.
        <fs_final>-vtext4 = ls_capability-vtext.
    ENDIF.
**Update the log table
    CLEAR: ls_history.
    ls_history-contract      = <fs_final>-contract.
    ls_history-item          = <fs_final>-item.
    ls_history-processed_date = sy-datum.
    ls_history-processed_time = sy-uzeit.
    ls_history-processed_by   = sy-uname.
    ls_history-rel_pkg_name   = <fs_final>-class_desc.
    ls_history-ship_to       = <fs_final>-ship_to.
    ls_history-product       = <fs_final>-product.
    ls_history-capability     = <fs_final>-vtext4.
    ls_history-curr_prod_ver  = lv_version.
    ls_history-po_desc       = <fs_final>-po_description.
    ls_history-delivery_method = <fs_final>-delivery_method.
    ls_history-cont_end_date  = <fs_final>-header_enddate.
    MODIFY zas_verupd_log FROM ls_history.

***file
***Write the data into application server file
*
    CLEAR: lv_file_data.
*
    CONCATENATE <fs_final>-ship_to
*
               <fs_final>-ship_to_customer
*
               <fs_final>-ship_to_2ltr
*
               <fs_final>-sold_to_customer
*
               <fs_final>-sold_to_2ltr
*
               <fs_final>-product
*
               <fs_final>-vtext1
*
               <fs_final>-vtext2
*
               <fs_final>-vtext3
*
               <fs_final>-vtext4
*
               <fs_final>-material_desc
*
               <fs_final>-class_desc
**
               <fs_final>-vc_prod_characteristic_value
*
               lv_version
*
               <fs_final>-contract
*
               <fs_final>-item
*
               <fs_final>-po_description
*
               <fs_final>-header_enddate
*
               <fs_final>-header_insdate
*
               <fs_final>-delivery_method
*
               INTO lv_file_data
*
               SEPARATED BY con_delimit.
*
    TRANSFER lv_file_data TO lv_file_name.
*
    CLEAR: lv_file_data.
    ENDLOOP.

**file
**& Close the file
*
    CLOSE DATASET lv_file_name.
    CLEAR: ls_status.
```

```
        ls_status-identifier = 1.
        ls_status-success   = abap_true.
    ELSE.
        CLEAR: ls_status.
        ls_status-identifier = 1.           "Data Issue - Update Failed
        ls_status-success   = abap_false.
    ENDIF.
*   "Set return status based on action:
*   me->copy_data_to_ref(
*       EXPORTING
*           is_data = ls_status
*       CHANGING
*           cr_data = er_data ).
    ELSE.
        "Failed due to invalid action by user:
        ls_status-identifier = 2.           "Access Issue - No Auth
        ls_status-success   = abap_false.
    ENDIF.
    "Set return status based on action:
    me->copy_data_to_ref(
        EXPORTING
            is_data = ls_status
        CHANGING
            cr_data = er_data ).
    WHEN OTHERS.
    ENDCASE.
ENDMETHOD.
```

CONTRACTVERSIOND_GET_ENTITYSET

```
    METHOD contractversiond_get_entityset.
**TRY.
**CALL METHOD SUPER->CONTRACTVERSIOND_GET_ENTITYSET
*   EXPORTING
*       IV_ENTITY_NAME           =
*       IV_ENTITY_SET_NAME      =
*       IV_SOURCE_NAME          =
*       IT_FILTER_SELECT_OPTIONS =
*       IS_PAGING               =
*       IT_KEY_TAB              =
*       IT_NAVIGATION_PATH      =
*       IT_ORDER                =
*       IV_FILTER_STRING        =
*       IV_SEARCH_STRING        =
**   io_tech_request_context =
** IMPORTING
**   et_entityset              =
**   es_response_context      =
*       .
** CATCH /iwbep/cx_mgw_busi_exception .
** CATCH /iwbep/cx_mgw_tech_exception .
**ENDTRY.
**Structure declaration
    TYPES: BEGIN OF st_final_tab,
```

```

        contract                TYPE vbeln_va,
        item                    TYPE posnr_va,
        header_enddate          TYPE vndat_veda,
        item_enddate            TYPE vndat_veda,
        header_insddate         TYPE vidat_veda,
        po_description           TYPE ktext_v,
        product                  TYPE matnr,
        product_hierarchy        TYPE prodh_d,
        header_objectno          TYPE objko,
        item_objectno            TYPE objpo,
        material_desc            TYPE arktx,
        ship_to                  TYPE kunnr,
        ship_to_customer         TYPE name1_gp,
        ship_to_2ltr             TYPE sortl,
        sold_to_customer         TYPE name1_gp,
        sold_to_2ltr             TYPE sortl,
        class_desc               TYPE klschl,
*        prod_characteristic_name TYPE atnam,
*        prod_characteristic_name_desc TYPE atbez,
*        prod_characteristic_value TYPE atwrt,
*        prod_characteristic_desc TYPE atwtb,
        vc_prod_characteristic_name TYPE atnam,
        vc_charateristic_desc     TYPE atbez,
        vc_prod_characteristic_value TYPE atwrt,
        vc_description           TYPE atwtb,
        prodh1                   TYPE prodh_d,
        vtext1                   TYPE bezei40,
        prodh2                   TYPE prodh_d,
        vtext2                   TYPE bezei40,
        prodh3                   TYPE prodh_d,
        vtext3                   TYPE bezei40,
        prodh4                   TYPE prodh_d,
        vtext4                   TYPE bezei40,
        delivery_method          TYPE atwtb,
        flag(1)                  TYPE c,
        END OF st_final_tab.
**Internal table and Workarea declaration
        DATA: lt_final_tab TYPE TABLE OF st_final_tab,
              ls_final_tab TYPE          st_final_tab.
        DATA ls_entity LIKE LINE OF et_entityset.
        DATA lo_filter TYPE REF TO /iwbep/if_mgw_req_filter.
        DATA lt_filter_select_options TYPE /iwbep/t_mgw_select_option.
        DATA ls_filter TYPE /iwbep/s_mgw_select_option.
        DATA lv_filter_str TYPE string.
        DATA ls_converted_keys LIKE LINE OF et_entityset.
        DATA: ls_order TYPE /iwbep/s_mgw_sorting_order.
**Structure and Internal table for input parameters
        DATA: lr_ship_to LIKE RANGE OF ls_converted_keys-ship_to,
              ls_ship_to LIKE LINE OF lr_ship_to,
              lr_ship_to_cust LIKE RANGE OF ls_converted_keys-ship_to_cust,
              ls_ship_to_cust LIKE LINE OF lr_ship_to_cust,
              lr_ship_to_2ltr LIKE RANGE OF ls_converted_keys-ship_to_2ltr,
              ls_ship_to_2ltr LIKE LINE OF lr_ship_to_2ltr,
              lr_sold_to_cust LIKE RANGE OF ls_converted_keys-sold_to_cust,

```

```
ls_sold_to_cust LIKE LINE OF lr_sold_to_cust,
lr_sold_to_2ltr LIKE RANGE OF ls_converted_keys-sold_to_2ltr,
ls_sold_to_2ltr LIKE LINE OF lr_sold_to_2ltr,
lr_rel_pkg_name LIKE RANGE OF ls_converted_keys-rel_pkg_name,
ls_rel_pkg_name LIKE LINE OF lr_rel_pkg_name,
lr_contract LIKE RANGE OF ls_converted_keys-contract,
ls_contract LIKE LINE OF lr_contract.
**Constant declataion
CONSTANTS: con_obtyp_vbk TYPE jsto-obtyp VALUE 'VBK',
            con_obtyp_vbp TYPE jsto-obtyp VALUE 'VBP',
            con_del_method TYPE atnam VALUE 'ZAS_PRODUCT_DELIVERY_METHOD',
**<< Begin of insert by vijay on 04/08/2020- For contract header status CSDU/CCU
            con_csd_u TYPE j_txt04 VALUE 'CSDU',
            con_ccu TYPE j_txt04 VALUE 'CCU'.
**>> End of insert by vijay on 04/08/2020
* Get filter or select option information
lo_filter = io_tech_request_context->get_filter( ).
lt_filter_select_options = lo_filter->get_filter_select_options( ).
lv_filter_str = lo_filter->get_filter_string( ).
IF lt_filter_select_options[] IS NOT INITIAL.
    LOOP AT lt_filter_select_options INTO ls_filter.
        CASE ls_filter-property.
            WHEN 'SHIP_TO'.
                lo_filter->convert_select_option(
                    EXPORTING
                        is_select_option = ls_filter
                    IMPORTING
                        et_select_option = lr_ship_to ).
            WHEN 'SHIP_TO_CUST'.
                lo_filter->convert_select_option(
                    EXPORTING
                        is_select_option = ls_filter
                    IMPORTING
                        et_select_option = lr_ship_to_cust ).
            WHEN 'SHIP_TO_2LTR'.
                lo_filter->convert_select_option(
                    EXPORTING
                        is_select_option = ls_filter
                    IMPORTING
                        et_select_option = lr_ship_to_2ltr ).
            WHEN 'CONTRACT'.
                lo_filter->convert_select_option(
                    EXPORTING
                        is_select_option = ls_filter
                    IMPORTING
                        et_select_option = lr_contract ).
            WHEN 'SOLD_TO_CUST'.
                lo_filter->convert_select_option(
                    EXPORTING
                        is_select_option = ls_filter
                    IMPORTING
                        et_select_option = lr_sold_to_cust ).
            WHEN 'SOLD_TO_2LTR'.
                lo_filter->convert_select_option(
```

```
*      EXPORTING
*      is_select_option = ls_filter
*      IMPORTING
*      et_select_option = lr_sold_to_2ltr ).
  WHEN 'REL_PKG_NAME'.
    lo_filter->convert_select_option(
      EXPORTING
        is_select_option = ls_filter
      IMPORTING
        et_select_option = lr_rel_pkg_name ).
  WHEN OTHERS.
  ENDCASE.
ENDLOOP.
ENDIF.
**Get the data from View zas_con_prod_rel (As it contains all the data)
SELECT
  contract
  item
  header_enddate
  item_enddate
  header_insddate
  po_description
  product
  product_hierarchy
  header_objectno
  item_objectno
  material_desc
  ship_to
  ship_to_customer
  ship_to_2ltr
  sold_to_customer
  sold_to_2ltr
  class_desc
*   prod_characteristic_name
*   prod_characteristic_name_desc
*   prod_characteristic_value
*   prod_characteristic_desc
  vc_prod_characteristic_name
  vc_charateristic_desc
  vc_prod_characteristic_value
  vc_description
FROM zas_con_prod_rel
INTO TABLE lt_final_tab
WHERE ship_to_customer IN lr_ship_to_cust
AND   ship_to IN lr_ship_to
AND   ship_to_2ltr      IN lr_ship_to_2ltr
*   AND   prod_characteristic_name_desc IN lr_rel_pkg_name
AND   class_desc IN lr_rel_pkg_name
AND   contract IN lr_contract.
*   AND   ( header_enddate >= sy-datum OR header_enddate EQ '00000000' ).
"commented by 303638
*   AND   ( item_enddate >= sy-datum OR item_enddate EQ '00000000' ).
IF sy-subrc = 0.
**Get header status details only for CRX(To ignore the contract)
```

```
SELECT a~objnr,
       b~stat,
       c~txt04
       INTO TABLE @DATA(lt_header_status)
       FROM jsto AS a
       INNER JOIN jest AS b
       ON a~objnr = b~objnr
       INNER JOIN tj30t AS c
       ON a~stsma = c~stsma
       AND b~stat = c~estat
       FOR ALL ENTRIES IN @lt_final_tab
       WHERE a~objnr EQ @lt_final_tab-header_objectno
       AND a~obtyp EQ @con_obtyp_vbk
       AND b~stat LIKE 'E%'
       AND b~inact EQ ''
       AND c~spras EQ 'E'
       AND c~txt04 EQ 'CRX'.
IF sy-subrc = 0.
ENDIF.
**Get Item status details only for (DUP, PDX, PRX, PCA) (To ignore the contract)
SELECT a~objnr,
       b~stat,
       c~txt04
       INTO TABLE @DATA(lt_item_status)
       FROM jsto AS a
       INNER JOIN jest AS b
       ON a~objnr = b~objnr
       INNER JOIN tj30t AS c
       ON a~stsma = c~stsma
       AND b~stat = c~estat
       FOR ALL ENTRIES IN @lt_final_tab
       WHERE a~objnr EQ @lt_final_tab-item_objectno
       AND a~obtyp EQ @con_obtyp_vbp
       AND b~stat LIKE 'E%'
       AND b~inact EQ ''
       AND c~spras EQ 'E'
       AND c~txt04 IN ( 'DUP' , 'PDX' , 'PRX' , 'PCA' , 'PCE' , 'PEX' ).
"Insert PCE & PEX by 303638
IF sy-subrc = 0.
ENDIF.
**Get Contract item delivery method details
SELECT sales_doc,
       item,
       material,
       description
       FROM zas_con_prod_ver
       INTO TABLE @DATA(lt_delivery_method)
       FOR ALL ENTRIES IN @lt_final_tab
       WHERE sales_doc EQ @lt_final_tab-contract
       AND item EQ @lt_final_tab-item
       AND material EQ @lt_final_tab-product
       AND characteristic_name EQ @con_del_method.
IF sy-subrc = 0.
ENDIF.
```


**<< Begin of insert by vijay on 04/08/2020 - To consider only the contracts which is having header status as CSDU/CCU.

**Get the header status details for CDSU and CCU

```
SELECT a~objnr,
       b~stat,
       c~txt04
INTO TABLE @DATA(lt_hdr_stat_csdu)
FROM jsto AS a
INNER JOIN jest AS b
ON a~objnr = b~objnr
INNER JOIN tj30t AS c
ON a~stsma = c~stsma
AND b~stat = c~estat
FOR ALL ENTRIES IN @lt_final_tab
WHERE a~objnr EQ @lt_final_tab-header_objectno
AND a~obtyp EQ @con_obtyp_vbk
AND b~stat LIKE 'E%'
AND b~inact EQ @abap_false
AND c~spras EQ 'E'
AND c~txt04 IN ( @con_csdu , @con_ccu ).

IF sy-subrc = 0.
ENDIF.
```

**>> End of insert by vijay on 04/08/2020

ENDIF.

LOOP AT lt_final_tab ASSIGNING FIELD-SYMBOL(<fs_final>).

**Check the header status. If CRX , then set the delete flag to ignore the contract.

```
READ TABLE lt_header_status INTO DATA(ls_header_status) WITH KEY objnr =
<fs_final>-header_objectno.
IF sy-subrc = 0.
    <fs_final>-flag = 'X'.
ENDIF.
```

**<< Begin of insert by vijay on 04/08/2020 - For contract header status CSDU/CCU.

****Check the header status. If CSDU or CCU then consider those contract. If not then set the delete flag to ignore the contract.

```
READ TABLE lt_hdr_stat_csdu INTO DATA(ls_hdr_stat_csdu) WITH KEY objnr =
<fs_final>-header_objectno.
IF sy-subrc NE 0.
    <fs_final>-flag = 'X'.
ENDIF.
```

**>> End of insert by vijay on 04/08/2020

**Check the item status. If its falls under 'DUP' , 'PDX' , 'PRX' , 'PCA' , 'PCE' , 'PEX' , then set the delete flag to ignore the c

```
READ TABLE lt_item_status INTO DATA(ls_item_status) WITH KEY objnr =
<fs_final>-item_objectno.
IF sy-subrc = 0.
    <fs_final>-flag = 'X'.
ENDIF.
```

IF <fs_final>-flag IS INITIAL AND <fs_final>-product_hierarchy IS NOT INITIAL.

**<< Begin of comment by vijay on 09/03/2021 - To incorporate new prod hier design

```
*    <fs_final>-prodh1 = <fs_final>-product_hierarchy+0(5).
*    <fs_final>-prodh2 = <fs_final>-product_hierarchy+0(9).
*    <fs_final>-prodh3 = <fs_final>-product_hierarchy+0(14).
*    <fs_final>-prodh4 = <fs_final>-product_hierarchy+0(18).
```

**>> End of comment by vijay on 09/03/2021

```
**<< Begin of insert by vijay on 09/03/2021 - To incorporate new prod hier design
    <fs_final>-prodh1 = <fs_final>-product_hierarchy+0(2).
    <fs_final>-prodh2 = <fs_final>-product_hierarchy+0(5).
    <fs_final>-prodh3 = <fs_final>-product_hierarchy+0(9).
    <fs_final>-prodh4 = <fs_final>-product_hierarchy+0(14).
**>> End of insert by vijay on 09/03/2021
    ENDIF.
**Move Delivery method to final table
    IF <fs_final>-flag IS INITIAL.
        READ TABLE lt_delivery_method INTO DATA(ls_delivery_method) WITH KEY
sales_doc = <fs_final>-contract
item = <fs_final>-item
material = <fs_final>-product.
        IF sy-subrc = 0.
            <fs_final>-delivery_method = ls_delivery_method-description.
        ENDIF.
    ENDIF.
ENDLOOP.
DELETE lt_final_tab WHERE flag = 'X'.
IF lt_final_tab[] IS NOT INITIAL.
**Get the product hierarhy description
    SELECT prodh,
           vtext
           INTO TABLE @DATA(lt_suite)
           FROM t179t
           FOR ALL ENTRIES IN @lt_final_tab
           WHERE prodh EQ @lt_final_tab-prodh1.
    IF sy-subrc = 0.
    ENDIF.
    SELECT prodh,
           vtext
           INTO TABLE @DATA(lt_sol_family)
           FROM t179t
           FOR ALL ENTRIES IN @lt_final_tab
           WHERE prodh EQ @lt_final_tab-prodh2.
    IF sy-subrc = 0.
    ENDIF.
    SELECT prodh,
           vtext
           INTO TABLE @DATA(lt_solution)
           FROM t179t
           FOR ALL ENTRIES IN @lt_final_tab
           WHERE prodh EQ @lt_final_tab-prodh3.
    IF sy-subrc = 0.
    ENDIF.
    SELECT prodh,
           vtext
           INTO TABLE @DATA(lt_capability)
           FROM t179t
           FOR ALL ENTRIES IN @lt_final_tab
           WHERE prodh EQ @lt_final_tab-prodh4.
    IF sy-subrc = 0.
    ENDIF.
**<< Begin of insert by vijay on 02/20/2020 - To display logs
```

```
**Get the Log details from ZAS_VERUPD_LOG table.
SELECT contract,
       item,
       processed_date,
       processed_time,
       processed_by,
       rel_pkg_name,
       curr_prod_ver
FROM zas_verupd_log
INTO TABLE @DATA(lt_verupd_log)
FOR ALL ENTRIES IN @lt_final_tab
WHERE contract EQ @lt_final_tab-contract
AND item EQ @lt_final_tab-item
AND rel_pkg_name EQ @lt_final_tab-class_desc
AND curr_prod_ver EQ @lt_final_tab-vc_prod_characteristic_value.
IF sy-subrc = 0.
    SORT lt_verupd_log BY contract item ASCENDING
                        processed_date DESCENDING
                        processed_time DESCENDING.
    DELETE ADJACENT DUPLICATES FROM lt_verupd_log COMPARING contract item.
**Get the processed by name from USER_ADDR table
SELECT bname,
       name_textc
FROM user_addr
INTO TABLE @DATA(lt_user_addr)
FOR ALL ENTRIES IN @lt_verupd_log
WHERE bname EQ @lt_verupd_log-processed_by.
IF sy-subrc = 0.
    ENDIF.
ENDIF.

**>> End of insert by vijay on 02/20/2020
**Get Release Package ID from CDS View ZASV_RPG_DATA
SELECT matnr_rpg,
       rpg_desc
FROM zasv_rpg_data
INTO TABLE @DATA(lt_rpg_data)
FOR ALL ENTRIES IN @lt_final_tab
WHERE rpg_desc EQ @lt_final_tab-class_desc.
IF sy-subrc = 0.
    SORT lt_rpg_data BY rpg_desc.
ENDIF.
SORT lt_final_tab BY product contract.
LOOP AT lt_final_tab INTO ls_final_tab.
** Get Suite Description
    READ TABLE lt_suite INTO DATA(ls_suite) WITH KEY prodh = ls_final_tab-prodh1.
    IF sy-subrc = 0.
        ls_final_tab-vtext1 = ls_suite-vtext.
    ENDIF.
** Get Solution Family Description
    READ TABLE lt_sol_family INTO DATA(ls_sol_family) WITH KEY prodh =
ls_final_tab-prodh2.
    IF sy-subrc = 0.
        ls_final_tab-vtext2 = ls_sol_family-vtext.
    ENDIF.
```

```
** Get Solution Description
  READ TABLE lt_solution INTO DATA(ls_solution) WITH KEY prodh =
ls_final_tab-prodh3.
  IF sy-subrc = 0.
    ls_final_tab-vtext3 = ls_solution-vtext.
  ENDIF.
** Get Capability Description
  READ TABLE lt_capability INTO DATA(ls_capability) WITH KEY prodh =
ls_final_tab-prodh4.
  IF sy-subrc = 0.
    ls_final_tab-vtext4 = ls_capability-vtext.
  ENDIF.
** Move the values to entity set
  ls_entity-ship_to      = ls_final_tab-ship_to.
  ls_entity-ship_to_cust = ls_final_tab-ship_to_customer.
  ls_entity-ship_to_2ltr = ls_final_tab-ship_to_2ltr.
  ls_entity-sold_to_cust = ls_final_tab-sold_to_customer.
  ls_entity-sold_to_2ltr = ls_final_tab-sold_to_2ltr.
  ls_entity-product      = ls_final_tab-product.
  ls_entity-suite        = ls_final_tab-vtext1.
  ls_entity-solution_family = ls_final_tab-vtext2.
  ls_entity-solution      = ls_final_tab-vtext3.
  ls_entity-capability     = ls_final_tab-vtext4.
  ls_entity-material_descr = ls_final_tab-material_desc.
*   ls_entity-rel_pkg_desc = ls_final_tab-prod_characteristic_name_desc.
*   ls_entity-rel_pkg_name = ls_final_tab-prod_characteristic_name_desc.
  ls_entity-rel_pkg_name = ls_final_tab-class_desc.
*   ls_entity-rel_pkg_name = ls_final_tab-prod_characteristic_desc.
  ls_entity-curr_prod_ver = ls_final_tab-vc_prod_characteristic_value.
*   ls_entity-curr_prod_ver = ls_final_tab-vc_prod_characteristic_name.
  ls_entity-contract      = ls_final_tab-contract.
  ls_entity-item          = ls_final_tab-item.
  ls_entity-po_desc       = ls_final_tab-po_description.
  IF ls_final_tab-header_enddate EQ ''.
    ls_entity-cont_end_date = '00000000'.
  ELSE.
    ls_entity-cont_end_date = ls_final_tab-header_enddate.
  ENDIF.
  IF ls_final_tab-header_insddate EQ ''.
    ls_entity-install_date = '00000000'.
  ELSE.
    ls_entity-install_date = ls_final_tab-header_insddate.
  ENDIF.
  ls_entity-delivery_method = ls_final_tab-delivery_method.
**<< Begin of insert by vijay on 02/20/2020 - for Log(Last updated by and date)
**Get the log
  READ TABLE lt_verupd_log INTO DATA(ls_verupd_log) WITH KEY contract =
ls_final_tab-contract
                                                                    item      =
ls_final_tab-item
                                                                    rel_pkg_name =
ls_final_tab-class_desc
                                                                    curr_prod_ver =
ls_final_tab-vc_prod_characteristic_value.
```

```
        IF sy-subrc = 0.
**Move the Processed date as last updated date
        ls_entity-updateddate = ls_verupd_log-processed_date.
**Get the processed by user name
        READ TABLE lt_user_addr INTO DATA(ls_user_addr) WITH KEY bname =
ls_verupd_log-processed_by.
        IF sy-subrc = 0.
**Move the processed by user name to updated by
        ls_entity-updatedby = ls_user_addr-name_textc.
        ENDIF.
        ENDIF.
**>> End of insert by vijay on 02/20/2020
**Move Release Package ID
        READ TABLE lt_rpg_data INTO DATA(ls_rpg_data) WITH KEY rpg_desc =
ls_final_tab-class_desc BINARY SEARCH.
        IF sy-subrc = 0.
        ls_entity-rel_pkg_id = ls_rpg_data-matnr_rpg.
        ENDIF.
        APPEND ls_entity TO et_entityset.
        CLEAR: ls_entity,
                ls_rpg_data.
        ENDLOOP.
    ENDIF.
** $inlinecount query option for all count entries.
    IF io_tech_request_context->has_inlinecount( ) = abap_true.
        DESCRIBE TABLE et_entityset LINES es_response_context-inlinecount.
    ELSE.
        CLEAR: es_response_context-inlinecount.
    ENDIF.
**** The function module for $top and $skip Query Options
*   CALL METHOD /iwbep/cl_mgw_data_util=>paging
*       EXPORTING
*           is_paging = is_paging
*       CHANGING
*           ct_data   = et_entityset.
IF et_entityset[] IS NOT INITIAL.
    READ TABLE it_order INTO ls_order INDEX 1.
    IF sy-subrc = 0.
        IF ls_order-order EQ 'asc'.
            CASE ls_order-property.
                WHEN 'ShipTo'.
                    SORT et_entityset BY ship_to ASCENDING.
                WHEN 'ShipToCust'.
                    SORT et_entityset BY ship_to_cust ASCENDING.
                WHEN 'ShipTo2ltr'.
                    SORT et_entityset BY ship_to_2ltr ASCENDING.
                WHEN 'SoldToCust'.
                    SORT et_entityset BY sold_to_cust ASCENDING.
                WHEN 'SoldTo2ltr'.
                    SORT et_entityset BY sold_to_2ltr ASCENDING.
                WHEN 'Product'.
                    SORT et_entityset BY product ASCENDING.
                WHEN 'Suite'.
                    SORT et_entityset BY suite ASCENDING.
```

```
    WHEN 'SolutionFamily'.
      SORT et_entityset BY solution_family ASCENDING.
    WHEN 'Solution'.
      SORT et_entityset BY solution ASCENDING.
    WHEN 'Capability'.
      SORT et_entityset BY capability ASCENDING.
    WHEN 'MaterialDescr'.
      SORT et_entityset BY material_descr ASCENDING.
    WHEN 'RelPkgName'.
      SORT et_entityset BY rel_pkg_name ASCENDING.
    WHEN 'CurrProdVer'.
      SORT et_entityset BY curr_prod_ver ASCENDING.
    WHEN 'Contract'.
      SORT et_entityset BY contract ASCENDING.
    WHEN 'Item'.
      SORT et_entityset BY item ASCENDING.
    WHEN 'PoDesc'.
      SORT et_entityset BY po_desc ASCENDING.
    WHEN 'ContEndDate'.
      SORT et_entityset BY cont_end_date ASCENDING.
    WHEN 'InstallDate'.
      SORT et_entityset BY install_date ASCENDING.
    WHEN 'DeliveryMethod'.
      SORT et_entityset BY delivery_method ASCENDING.
**<< Begin of insert by vijay on 02/20/2020 - log
    WHEN 'Updatedby'.
      SORT et_entityset BY updatedby ASCENDING.
    WHEN 'Updateddate'.
      SORT et_entityset BY updateddate ASCENDING.
**>> End of insert by vijay on 02/20/2020
    WHEN 'RelPkgId'.
      SORT et_entityset BY rel_pkg_id ASCENDING.
  ENDCASE.
ELSEIF ls_order-order EQ 'desc'.
  CASE ls_order-property.
    WHEN 'ShipTo'.
      SORT et_entityset BY ship_to DESCENDING.
    WHEN 'ShipToCust'.
      SORT et_entityset BY ship_to_cust DESCENDING.
    WHEN 'ShipTo2ltr'.
      SORT et_entityset BY ship_to_2ltr DESCENDING.
    WHEN 'SoldToCust'.
      SORT et_entityset BY sold_to_cust DESCENDING.
    WHEN 'SoldTo2ltr'.
      SORT et_entityset BY sold_to_2ltr DESCENDING.
    WHEN 'Product'.
      SORT et_entityset BY product DESCENDING.
    WHEN 'Suite'.
      SORT et_entityset BY suite DESCENDING.
    WHEN 'SolutionFamily'.
      SORT et_entityset BY solution_family DESCENDING.
    WHEN 'Solution'.
      SORT et_entityset BY solution DESCENDING.
    WHEN 'Capability'.
```

```
        SORT et_entityset BY capability DESCENDING.
      WHEN 'MaterialDescr'.
        SORT et_entityset BY material_descr DESCENDING.
      WHEN 'RelPkgName'.
        SORT et_entityset BY rel_pkg_name DESCENDING.
      WHEN 'CurrProdVer'.
        SORT et_entityset BY curr_prod_ver DESCENDING.
      WHEN 'Contract'.
        SORT et_entityset BY contract DESCENDING.
      WHEN 'Item'.
        SORT et_entityset BY item DESCENDING.
      WHEN 'PoDesc'.
        SORT et_entityset BY po_desc DESCENDING.
      WHEN 'ContEndDate'.
        SORT et_entityset BY cont_end_date DESCENDING.
      WHEN 'InstallDate'.
        SORT et_entityset BY install_date DESCENDING.
      WHEN 'DeliveryMethod'.
        SORT et_entityset BY delivery_method DESCENDING.
**<< Begin of insert by vijay on 02/20/2020 - log
      WHEN 'Updatedby'.
        SORT et_entityset BY updatedby DESCENDING.
      WHEN 'Updateddate'.
        SORT et_entityset BY updateddate DESCENDING.
**>> End of insert by vijay on 02/20/2020
      WHEN 'RelPkgId'.
        SORT et_entityset BY rel_pkg_id DESCENDING.
    ENDCASE.
  ENDIF.
ENDIF.
ENDIF.
*** The function module for $stop and $skip Query Options
CALL METHOD /iwbsp/cl_mgw_data_util=>paging
  EXPORTING
    is_paging = is_paging
  CHANGING
    ct_data   = et_entityset.
***Get the product release package data with contract
*   SELECT contract
*     item
*     po_description
*     product
*     material_desc
*     sold_to_customer
*     sold_to_2ltr
***     characteristic_name,
***     characteristic_value,
*     characteristic_desc
*   FROM zas_con_prod_rel
*   INTO TABLE lt_final_tab
*   WHERE characteristic_desc IN lr_rel_pkg_name
*     AND contract IN lr_contract.
*   IF sy-subrc = 0.
***Get contract product version from view ZCDS_CON_VERSION
```

```
*      SELECT sales_doc,
*             item,
*             characteristic_name,
*             charateristic_desc,
*             characteristic_value,
*             description,
*             material
*      FROM zas_con_prod_ver
*      INTO TABLE @DATA(lt_con_version)
*      FOR ALL ENTRIES IN @lt_final_tab
*      WHERE sales_doc = @lt_final_tab-contract
*      AND   item = @lt_final_tab-item
*      AND   material = @lt_final_tab-product.
*
***Get VEDA details
*      SELECT vbeln,
*             vposn,
*             venddat,
*             vkuesch
*      FROM veda
*      INTO TABLE @DATA(lt_veda)
*      FOR ALL ENTRIES IN @lt_final_tab
*      WHERE vbeln EQ @lt_final_tab-contract
**      AND venddat GT @sy-datum
*      AND vkuesch = ''.
*
*      IF sy-subrc = 0.
*      ENDIF.
***Get Shift to details from VBPA
*      SELECT vbeln,
*             posnr,
*             parvw,
*             kunnr
*      FROM vbpa
*      INTO TABLE @DATA(lt_vbpa)
*      FOR ALL ENTRIES IN @lt_final_tab
*      WHERE vbeln EQ @lt_final_tab-contract
*      AND parvw EQ 'WE'.
*
*      IF sy-subrc = 0.
***Get Customer Name and 2ltr code
*      SELECT kunnr,
*             namel,
*             sortl
*      FROM knal
*      INTO TABLE @DATA(lt_knal)
*      FOR ALL ENTRIES IN @lt_vbpa
*      WHERE kunnr EQ @lt_vbpa-kunnr.
*
*      IF sy-subrc = 0.
*      ENDIF.
*
*      ENDIF.
*
*      ENDIF.
*
*      LOOP AT lt_final_tab ASSIGNING FIELD-SYMBOL(<fs_final_tab>).
***      Check contract is active or not?? if not just ignore the entry
*      READ TABLE lt_veda INTO DATA(ls_veda) WITH KEY vbeln = <fs_final_tab>-contract
```



```
*      IF lr_ship_to_cust[] IS NOT INITIAL.
*          DELETE lt_final_tab WHERE ship_to_cust NOT IN lr_ship_to_cust.
*      ENDIF.
*      IF lr_ship_to_2ltr[] IS NOT INITIAL.
*          DELETE lt_final_tab WHERE ship_to_2ltr NOT IN lr_ship_to_2ltr.
*      ENDIF.
*
*      LOOP AT    lt_final_tab INTO ls_final_tab.
*          ls_entity-ship_to_cust = ls_final_tab-ship_to_cust.
*          ls_entity-ship_to_2ltr = ls_final_tab-ship_to_2ltr.
*          ls_entity-sold_to_cust = ls_final_tab-sold_to_cust.
*          ls_entity-sold_to_2ltr = ls_final_tab-sold_to_2ltr.
*          ls_entity-product      = ls_final_tab-product.
*          ls_entity-material_descr = ls_final_tab-material_descr.
*          ls_entity-rel_pkg_name = ls_final_tab-rel_pkg_name.
*          ls_entity-curr_prod_ver = ls_final_tab-curr_prod_ver.
*          ls_entity-contract     = ls_final_tab-contract.
*          ls_entity-item         = ls_final_tab-item.
*          ls_entity-po_desc      = ls_final_tab-po_desc.
*          APPEND ls_entity TO et_entityset.
*          CLEAR: ls_entity.
*      ENDLOOP.
*  ENDMETHOD.
```

IF_SADL_GW_QUERY_CONTROL~SET_QUERY_OPTIONS

```
METHOD if_sadl_gw_query_control~set_query_options.
**TRY.
**CALL METHOD SUPER->IF_SADL_GW_QUERY_CONTROL~SET_QUERY_OPTIONS
*   EXPORTING
*       IV_ENTITY_SET      =
*       IO_QUERY_OPTIONS =
*
** CATCH /iwbsp/cx_mgw_busi_exception .
** CATCH /iwbsp/cx_mgw_tech_exception .
**ENDTRY.

"Method-local data declarations:
DATA lt_parameters      TYPE if_sadl_public_types=>tt_entity_parameters.
DATA lt_sort_elements  TYPE if_sadl_public_types=>tt_sort_elements.
DATA lt_search_scope    TYPE if_sadl_public_types=>tt_search_scope.
DATA lt_req_elements    TYPE if_sadl_public_types=>tt_requested_elements.
DATA lo_provider        TYPE REF TO if_sadl_cond_prov_auth_objects.
DATA lv_search_term     TYPE string.
CASE iv_entity_set.
    WHEN 'ShipToCustSet'.
        "Set search scope to include all visible/requested (above) fields:
        lt_search_scope = VALUE #( ( `CUSTOMER` ) ( `NAME` ) ( `CODE` ) ).
        io_query_options->set_text_search_scope( lt_search_scope ).
        lv_search_term = io_query_options->get_text_search_term( ).
        lv_search_term = `*` && lv_search_term && `*`.
        io_query_options->set_text_search_term( lv_search_term ).
    WHEN OTHERS.
ENDCASE.
ENDMETHOD.
```

RELEASEPKGDATAASE_GET_ENTITYSET

```
METHOD releasepkgdataase_get_entityset.
**TRY.
*CALL METHOD SUPER->RELEASEPKGDATAASE_GET_ENTITYSET
*   EXPORTING
*       IV_ENTITY_NAME           =
*       IV_ENTITY_SET_NAME       =
*       IV_SOURCE_NAME           =
*       IT_FILTER_SELECT_OPTIONS =
*       IS_PAGING                =
*       IT_KEY_TAB               =
*       IT_NAVIGATION_PATH       =
*       IT_ORDER                 =
*       IV_FILTER_STRING         =
*       IV_SEARCH_STRING         =
**       io_tech_request_context =
**   IMPORTING
**       et_entityset            =
**       es_response_context    =
*       .
** CATCH /iwbep/cx_mgw_busi_exception .
** CATCH /iwbep/cx_mgw_tech_exception .
**ENDTRY.
**<< Begin of insert by vijay on 07/10/2020 - Redesign
    SELECT rpg_desc
          FROM zasv_rpg_data
          INTO TABLE @DATA(lt_rel_pkg).
    IF sy-subrc = 0.
        SORT lt_rel_pkg BY rpg_desc.
        et_entityset[] = lt_rel_pkg[].
    ENDIF.
**>> End of insert by vijay on 07/10/2020
**<< Begin of comment by vijay on 07/10/2020 - Redesign
*   SELECT class_desc
*           FROM zas_clas_rel_pkg
*           INTO TABLE @DATA(lt_rel_pkg).
*   IF sy-subrc = 0.
*       SORT lt_rel_pkg BY class_desc.
*       et_entityset[] = lt_rel_pkg[].
*   ENDIF.
**>> End of comment by vijay on 07/10/2020
ENDMETHOD.
```

RELEASEPKGVERNDA_GET_ENTITYSET

```
METHOD releasepkgvernda_get_entityset.
**TRY.
*CALL METHOD SUPER->RELEASEPKGVERNDA_GET_ENTITYSET
*   EXPORTING
*       IV_ENTITY_NAME           =
*       IV_ENTITY_SET_NAME       =
*       IV_SOURCE_NAME           =
*       IT_FILTER_SELECT_OPTIONS =
```

```
*      IS_PAGING                =
*      IT_KEY_TAB                =
*      IT_NAVIGATION_PATH       =
*      IT_ORDER                  =
*      IV_FILTER_STRING          =
*      IV_SEARCH_STRING          =
**      io_tech_request_context  =
**      IMPORTING
**      et_entityset             =
**      es_response_context      =
*
** CATCH /iwbsp/cx_mgw_busi_exception .
** CATCH /iwbsp/cx_mgw_tech_exception .
**ENDTRY.
*Internal table and Workarea declaration
  DATA: lo_filter TYPE REF TO /iwbsp/if_mgw_req_filter.
  DATA: lt_filter_select_options TYPE /iwbsp/t_mgw_select_option.
  DATA: ls_filter TYPE /iwbsp/s_mgw_select_option.
  DATA: lv_filter_str TYPE string.
  DATA: ls_converted_keys LIKE LINE OF et_entityset.
  DATA: ls_entityset LIKE LINE OF et_entityset.
  DATA: lt_order TYPE /iwbsp/t_mgw_tech_order.
  DATA: ls_order TYPE /iwbsp/s_mgw_tech_order.
  DATA: lt_otab TYPE abap_sortorder_tab,
        ls_oline TYPE abap_sortorder.
**Structure and Internal table for input parameters
  DATA: lr_rel_pkg_name LIKE RANGE OF ls_converted_keys-release_pkg_name,
        ls_rel_pkg_name LIKE LINE OF lr_rel_pkg_name.
**<< Begin of insert by vijay on 08/31/2020 -To display only serviceable version
**Constants declaration
  CONSTANTS: lc_flag_y TYPE c VALUE 'Y'.
**>> End of insert by vijay on 08/31/2020
**Get filter or select option information
  lo_filter = io_tech_request_context->get_filter( ).
  lt_filter_select_options = lo_filter->get_filter_select_options( ).
  lv_filter_str = lo_filter->get_filter_string( ).
  IF lt_filter_select_options[] IS NOT INITIAL.
    LOOP AT lt_filter_select_options INTO ls_filter.
      CASE ls_filter-property.
        WHEN 'RELEASE_PKG_NAME'.
          lo_filter->convert_select_option(
            EXPORTING
              is_select_option = ls_filter
            IMPORTING
              et_select_option = lr_rel_pkg_name ).
        WHEN OTHERS.
      ENDCASE.
    ENDLOOP.
  ENDIF.
**Get Release Package with Version details
  SELECT zas_rp_desc,
        zas_rp_version
  FROM zas_rp_vrsn_mast
  INTO TABLE @DATA(lt_rp_vrsn_mast)
```

```

        WHERE zas_rp_desc IN @l_r_rel_pkg_name
        AND    zas_serv_flag = @l_c_flag_y.
- To display only serviceable flag
    IF sy-subrc = 0.
        et_entityset[] = lt_rp_vrsn_mast[].
        SORT et_entityset BY release_pkg_name version.
** $inlinecount query option for all count entries.
    IF io_tech_request_context->has_inlinecount( ) = abap_true.
        DESCRIBE TABLE et_entityset LINES es_response_context-inlinecount.
    ELSE.
        CLEAR: es_response_context-inlinecount.
    ENDIF.
*** The function module for $stop and $skip Query Options
    CALL METHOD /iwbep/cl_mgw_data_util=>paging
        EXPORTING
            is_paging = is_paging
        CHANGING
            ct_data   = et_entityset.
**Sorting
    lt_order = io_tech_request_context->get_orderby( ).
    LOOP AT lt_order INTO ls_order.
        ls_oline-name = ls_order-property.
        IF ls_order-order = /iwbep/cl_mgw_data_util=>gcs_sorting_order-descending.
            ls_oline-descending = abap_true.
        ENDIF.
        APPEND ls_oline TO lt_otab.
        CLEAR ls_oline.
    ENDLOOP.
    IF lt_otab[] IS NOT INITIAL.
        SORT et_entityset BY (lt_otab).
    ENDIF.
ENDIF.
ENDMETHOD.

```

Local Types

```

*** use this source file for any type of declarations (class
*** definitions, interfaces or type declarations) you need for
*** components in the private section

```

Local class definitions

```

*** use this source file for the definition and implementation of
*** local helper classes, interface definitions and type
*** declarations

```

Macros

```

*** use this source file for any macro definitions you need
*** in the implementation part of the class

```

Overview

Attributes	1
Methods	1
Redefined Methods	1
	1
	12
	26
	27
	27
Local Types	29
Local class definitions	29
Macros	29