class /CGDC/CL\_CLRQ\_PROCESS definition  
  public  
  create public .  
  
public section.  
  
  types:  
    gtt\_cr\_doctype TYPE STANDARD TABLE OF /cgdc/t\_clrtypet .  
  types:  
    BEGIN OF GTY\_CF\_POP\_enDAT,  
        vbeln  TYPE vbeln\_va,  
        posnr  TYPE posnr\_va,  
        fendat TYPE /cgdc/fendat,  
        fstdat TYPE /cgdc/fstdat,  
      END OF gty\_cf\_pop\_endat .  
  types:  
    gtt\_cf\_pop\_endat TYPE STANDARD TABLE OF gty\_cf\_pop\_endat .  
  types:  
    gtt\_clrq\_err TYPE STANDARD TABLE OF /cgdc/t\_clrq\_err .  
  types:  
    BEGIN OF gty\_bcpop,  
        vbeln TYPE vbeln\_va,  
        bcpop TYPE /cgdc/bcopop,  
      END OF gty\_bcpop .  
  types:  
    gtt\_bcpop TYPE STANDARD TABLE OF gty\_bcpop .  
  types:  
    BEGIN OF gty\_prps,  
        pspnr TYPE ps\_posnr,  
        psphi TYPE ps\_psphi,  
      END OF gty\_prps .  
  types:  
    gtt\_prps TYPE HASHED TABLE OF gty\_prps WITH UNIQUE KEY pspnr .  
  
  constants:  
*\*\*\*New status related changes*  
    BEGIN OF gc\_itm\_bill\_status,  
        open              TYPE /cgdc/clrqi\_billstat VALUE ' ',  
        billing\_req       TYPE /cgdc/clrqi\_billstat VALUE '1',  
        pbd\_created       TYPE /cgdc/clrqi\_billstat VALUE '2',  
        pbd\_finalized     TYPE /cgdc/clrqi\_billstat VALUE '3',  
        pbd\_in\_rework     TYPE /cgdc/clrqi\_billstat VALUE '4',  
        pbd\_approved      TYPE /cgdc/clrqi\_billstat VALUE '5',  
        invoice\_created   TYPE /cgdc/clrqi\_billstat VALUE '6',  
        erroneous\_invoice TYPE /cgdc/clrqi\_billstat VALUE '7',  
        invoice\_posted    TYPE /cgdc/clrqi\_billstat VALUE '8',  
        invoice\_cleared   TYPE /cgdc/clrqi\_billstat VALUE '9',  
      END OF gc\_itm\_bill\_status .  
*\*\*\**  
  class-data GT\_ACDOCA type FINST\_ACDOCA .  
  class-data GT\_COEP type TT\_COEP .  
  class-data GT\_COST\_CONDITIONS type KOMV\_ITAB .  
  class-data GT\_COBK type TTCOBK .  
  class-data GT\_CLRQ\_CS\_KOMV type KOMV\_ITAB .  
  class-data GT\_PRPS type TT\_PRPS .  
  class-data GS\_RTBP type /CGDC/T\_RTBP .  
  class-data GT\_CALC\_GROUP type /CGDC/TT\_RTBPCG .  
  class-data GT\_CALC\_GROUP\_FUN type /CGDC/TT\_RTBPCGF .  
  class-data GT\_CALC\_GROUP\_FUN\_SRC type /CGDC/TT\_RTBPCGFC .  
  class-data GT\_CGROUP\_FUN\_SRC\_QUALS type /CGDC/TT\_TRPCGFSQ .  
  class-data GS\_PERFORMANCE\_DATE type /CGDC/S\_CLRQ\_POPDATE .  
  class-data GT\_CE\_CLRIT\_KEY type /CGDC/TT\_CLRQIT\_KEY .  
  class-data GV\_VBELN type VBAK-VBELN .  
  class-data GV\_NO\_NEW\_BILL\_ITEM type FLAG .  
  class-data:  
    gt\_contract\_itm TYPE STANDARD TABLE OF vbap .  
  class-data GT\_MESG type TY\_T\_MSEG .  
  class-data GT\_PRPS\_PSPHI type GTT\_PRPS .  
  class-data GT\_BCPOP type GTT\_BCPOP .  
  class-data GT\_CLRQ\_ERR type GTT\_CLRQ\_ERR .  
  class-data GT\_CF\_POP\_ENDAT type GTT\_CF\_POP\_ENDAT .  
  class-data GT\_COST\_CONDITIONS\_ADD type /CGDC/TT\_CLRQ\_CS\_CON\_DATA .  
  class-data GT\_CR\_DOCTYPE type GTT\_CR\_DOCTYPE .  
  class-data GV\_SO\_RTB\_FLAG type ABAP\_BOOL .  
  
  class-methods PROCESS\_REALTIME\_DATA  
    importing  
      !IT\_COEP type TT\_COEP optional  
      !IT\_COBK type TTCOBK optional  
      !IS\_BKPF type BKPF optional  
      !IT\_ACDOCA type FINST\_ACDOCA optional  
      value(IV\_RTB\_EVENT) type /CGDC/RTB\_EVENT optional  
      !IT\_CLRQSRC\_DATA type /CGDC/TT\_CLRQSRC\_DATA optional  
    exporting  
      !ET\_RETURN type /CGDC/T\_MESSAGE .  
  class-methods GET\_PREVIOUSLY\_BILLED  
    importing  
      !IV\_BILL\_MATERIAL type MATNR optional  
      !IV\_BILL\_DOC type VBELN  
      !IV\_SPLIT\_BY\_BIL\_ELEMENT type CHAR1 default SPACE  
    exporting  
      !ET\_PREV\_BILL\_DETAILS type /CGDC/T\_KONV\_DET  
      !ET\_PREV\_BILL\_SUMMARY type /CGDC/T\_VBRP\_DET .  
  class-methods SUBSEQUENT\_DOC\_DATA\_UPDATE  
    importing  
      !IT\_VBRK type VBRKVB\_T  
      !IT\_VBRP type VBRPVB\_T  
      !IV\_PROCESS type CHAR1 optional .  
  class-methods BUFFER\_COSTSHEET\_CONDITIONS  
    importing  
      !IT\_KOMV type KOMV\_ITAB  
      !IT\_KOMV\_ADD\_DATA type /CGDC/TT\_CLRQ\_CS\_CON\_DATA .  
  class-methods GET\_COSTSHEET\_CONDITIONS  
    exporting  
      !ET\_KOMV type KOMV\_ITAB  
      !ET\_KOMV\_ADD\_DATA type /CGDC/TT\_CLRQ\_CS\_CON\_DATA .  
  class-methods CHECK\_COSTING\_SHEET\_INTG  
    importing  
      !I\_OBJNR type OBJPO  
      !I\_RTBPN type /CGDC/RTBPN  
    exporting  
      !E\_STATUS type /CGDC/CLRQ\_CSIST .  
  class-methods SET\_CLRQ\_COSTSHEET\_CONDITIONS  
    importing  
      !IT\_KOMV type KOMV\_TAB  
      !IS\_TKA01 type TKA01 optional .  
  class-methods GET\_CLRQ\_COSTSHEET\_CONDITIONS  
    importing  
      !I\_KSCHL type KSCHL  
    exporting  
      !ES\_KOMV type KOMV  
      !ES\_TKA01 type TKA01 .  
  class-methods SELECT\_AMOUNTS\_CALC\_GRP\_ROW  
    importing  
      !IS\_CLRQHD type /CGDC/S\_CLRQHD  
      !IT\_CRGRRF type /CGDC/TT\_CLRQGRRF  
      !IS\_CRGRR type /CGDC/S\_CLRQGRR  
      !I\_KSCHL type KSCHL  
      !IV\_CONSTR\_DT type DATUM optional  
    changing  
      !C\_VALUE type KWERT .  
  class-methods SELECT\_REALTIME\_BILLPRF\_DATA  
    importing  
      !I\_RTBPN type /CGDC/RTBPN .  
  class-methods CHECK\_CALC\_GRP\_PROC\_COND  
    importing  
      !I\_KSCHL type KSCHL  
    exporting  
      !E\_SUBRC type SYSUBRC .  
  class-methods GET\_CALC\_GRP\_PROC\_COND\_VAL  
    importing  
      !I\_KSCHL type KSCHL  
    exporting  
      !E\_KWERT type KWERT .  
  class-methods BDR\_REJECT\_CHECK  
    importing  
      !IM\_KOMKBV3 type KOMKBV3  
    exporting  
      !EX\_SUBRC type SYST\_SUBRC .  
  class-methods BDR\_REJECT  
    importing  
      !IT\_PREBILLING type IF\_SD\_BIL\_TYPE\_DEF=>TT\_DRAFT\_DOCUMENT  
    exporting  
      !CV\_SUBRC type SY-SUBRC .  
  class-methods GET\_FEE\_VALUE  
    importing  
      !I\_KZWI1 type KZWIS  
      !I\_KZWI4 type KZWIS  
      !I\_KZWI5 type KZWIS  
    changing  
      !C\_XKWERT type KWERT .  
  methods MESSAGE\_HANDLING  
    importing  
      !IT\_ERROR type /CGDC/T\_CLRQ\_ERR .  
  class-methods DETERMINE\_CALCREQ\_AND\_CONTRACT  
    importing  
      !IS\_VBRK type VBRK  
    exporting  
      !ES\_CLRQHD type /CGDC/S\_CLRQHD  
      !ES\_VBAK type VBAK .  
  class-methods GET\_PERFORMANCE\_DATE  
    importing  
      !IS\_CLRQHDR type /CGDC/S\_CLRQHD optional  
      !IS\_CLRQIT type /CGDC/S\_CLRQIT optional  
      !IS\_CLRQBI type /CGDC/S\_CLRQBI optional  
      !IV\_OBJTY type CHAR1  
      !IS\_CLRQGRR type /CGDC/S\_CLRQGRR optional  
      !IV\_DATE\_TY type /CGDC/CLRQ\_CUST\_DT\_RULE\_TYPE optional  
    exporting  
      !EV\_DATE type DATS  
    exceptions  
      EX\_ITEM\_NOT\_PASSED  
      EX\_CONTRACT\_NOT\_PASSED .  
  class-methods CHECK\_BILLING\_ELIGIBILITY  
    importing  
      !IV\_VBELN type VBELN  
    exporting  
      !EV\_BLOCK\_BILLING type CHAR1  
      !EV\_BLOCKED\_CONTRACT type VBELN  
      !ET\_BLOCKED\_ITMS type /CGDC/TT\_CR\_BLOCKED\_ITEMS  
    exceptions  
      NON\_RTB\_DOC .  
  class-methods GENERATE\_HEADER\_DF  
    importing  
      !IS\_BILL\_DATA type VBCO6  
      !IS\_BILL\_HEADER type VBRK  
    changing  
      !CT\_VBFAS\_TAB type VBFAS\_T .  
  class-methods GENERATE\_ITEM\_DF  
    importing  
      !IS\_BILL\_DATA type VBCO6  
      !IS\_BILL\_HEADER type VBRK  
    changing  
      !CT\_VBFAS\_TAB type VBFAS\_T .  
  class-methods ADJUST\_BASEDATE  
    importing  
      !IV\_ADJST\_TYPE type /CGDC/CLRQ\_CUST\_DT\_ADJUSTMENT  
    changing  
      !C\_DATE type SYDATUM .  
  class-methods UPD\_COST\_ELEMENT\_ITEMS  
    importing  
      !IV\_COST\_ELEM\_GRP type /CGDC/CEGRP  
      !IV\_COST\_ELEM type KSTAR optional .  
  class-methods RAISE\_COST\_ELEMENT\_HDR\_MSG .  
  class-methods RTB\_AUTHORIZATION  
    importing  
      !IV\_CLRQY type /CGDC/CLRQT optional  
      !IV\_VKORG type VKORG optional  
      !IV\_SPART type SPART optional  
      !IV\_FNKEY type /CGDC/RTB\_AUTHFNKEY optional  
      !IV\_RTB\_ACTV type ACTIV\_AUTH optional  
      !IV\_VTWEG type VTWEG optional  
      !IV\_AUART type AUART optional  
    exporting  
      !EV\_INVALID type FLAG .  
  class-methods RTB\_DOC\_USAGE\_CHECK  
    importing  
      !I\_VBTYP type VBTYP  
      !I\_ABRVW type ABRVW  
    exporting  
      !E\_RC type NUMC1 .  
  class-methods VALIDATE\_POPEND\_DATE  
    importing  
      value(IV\_VBELN) type VGBEL  
      value(IV\_POSNR) type VGPOS  
    changing  
      !CV\_POP\_END\_DATE type DATUM .  
  class-methods MAP\_CLRQSRC\_DATA  
    importing  
      !IS\_DATA type DATA  
      !IV\_TABNAME type TABNAME  
      !IT\_SRC\_FIELDS type /CGDC/TT\_DD03L  
    changing  
      !CS\_CLRQSRC type /CGDC/DCO\_SRC .  
  class-methods COLLECT\_DATA\_FOR\_MIGO  
    importing  
      !IT\_ACDOCA type FINST\_ACDOCA optional  
      !IT\_MSEG type TY\_T\_MSEG optional .  
  class-methods SET\_COEP  
    importing  
      !IT\_COEP type TT\_COEP .  
  class-methods SET\_ACDOCA  
    importing  
      !IT\_ACDOCA type FINST\_ACDOCA .  
  class-methods SET\_COBK  
    importing  
      !IT\_COBK type TTCOBK .  
  class-methods COPY\_PBD\_CONDITIONS  
    importing  
      !IS\_KOMP type KOMP  
      !IV\_KOMV\_KSCHL type KSCHL  
      !IS\_KOMK type KOMK  
    changing  
      !CV\_KWERT type KWERT .  
  class-methods READ\_PRPS\_PSPHI  
    importing  
      !IV\_PS\_PSP\_PNR type /CGDC/PS\_PSP\_PNR  
    returning  
      value(RV\_PSPHI) type PS\_PSPHI .  
  class-methods READ\_BCPOP  
    importing  
      !IV\_VBELN type VBELN\_VA  
    returning  
      value(RV\_BCPOP) type /CGDC/BCOPOP .  
  class-methods GET\_PERFORMANCE\_DATE\_BRF  
    importing  
      !IS\_VBAP type VBAP optional  
      !IV\_DATERULE type VDREG\_VEDA  
    exporting  
      !EV\_DATE type DATS .  
  class-methods GET\_ERROR\_CONFG  
    exporting  
      value(ET\_CLRQ\_ERR) type GTT\_CLRQ\_ERR .  
  class-methods READ\_CF\_POPSTART\_DATE  
    importing  
      !IV\_VBELN type VBELN\_VA  
      !IV\_POSNR type POSNR\_VA  
    returning  
      value(RV\_FENDAT) type /CGDC/FENDAT .  
  class-methods READ\_CF\_POPEND\_DATE  
    importing  
      !IV\_VBELN type VBELN\_VA  
      !IV\_POSNR type POSNR\_VA  
    returning  
      value(RV\_FENDAT) type /CGDC/FENDAT .  
  class-methods SELECT\_AMOUNTS\_CALC\_GRP\_DIST  
    importing  
      !IS\_CLRQHD type /CGDC/S\_CLRQHD optional  
      !IT\_CRGRRF type /CGDC/TT\_CLRQGRRF optional  
      !IS\_CRGRR type /CGDC/RTB\_CLGRN optional  
      !IS\_DIST type /CGDC/S\_CLRQGRR optional  
      !IS\_CRGCGF type /CGDC/S\_RTBPCGF optional  
      !IS\_CRCGFC type /CGDC/S\_RTBPCGFC optional  
      !I\_KSCHL type KSCHL optional  
    exporting  
      !ET\_RESULT type /CGDC/TT\_CLRQ\_DIST  
    changing  
      !C\_VALUE type KWERT optional .  
  class-methods UPDATE\_MLTIPM\_CLRQ\_DATA  
    importing  
      !IS\_MLPAY\_DOC type /CGDC/S\_MLPAY\_DOC optional  
      !IS\_MLPAYSA\_DOC type /CGDC/S\_MLPAYSA\_DOC optional .  
  class-methods COPY\_CLGRP\_COND  
    importing  
      !IS\_KOMP type KOMP  
      !IV\_KOMV\_KSCHL type KSCHL  
      !IS\_KOMK type KOMK  
    changing  
      !CV\_KWERT type KWERT .  
  class-methods GET\_FIELDINFO  
    importing  
      !I\_TABNAME type TABNAME  
    exporting  
      !ET\_DFIES type DFIES\_TAB .  
  class-methods GET\_CR\_DOCTYPE  
    importing  
      !IS\_VBFA type DOCUMENT\_FLOW\_ALV\_STRUC  
    returning  
      value(RV\_DOCTYPE) type FMRE\_BATXT .  
  class-methods IS\_RTBP\_ON\_CONTR  
    importing  
      !IV\_CLRNO type /CGDC/CLRQNO  
    returning  
      value(RV\_RESULT) type ABAP\_BOOL .  
  class-methods VALIDATE\_CLRQ\_TRGR .  
protected section.  
  
  class-data GS\_TKA01 type TKA01 .  
private section.  
ENDCLASS.  
  
  
  
CLASS /CGDC/CL\_CLRQ\_PROCESS IMPLEMENTATION.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>BUFFER\_COSTSHEET\_CONDITIONS*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IT\_KOMV                        TYPE        KOMV\_ITAB*  
*\* | [--->] IT\_KOMV\_ADD\_DATA               TYPE        /CGDC/TT\_CLRQ\_CS\_CON\_DATA*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD BUFFER\_COSTSHEET\_CONDITIONS.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Buffer Costsheet Conditions*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
    gt\_cost\_conditions = it\_komv.  
    gt\_cost\_conditions\_add = it\_komv\_add\_data.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>SET\_CLRQ\_COSTSHEET\_CONDITIONS*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IT\_KOMV                        TYPE        KOMV\_TAB*  
*\* | [--->] IS\_TKA01                       TYPE        TKA01(optional)*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD set\_clrq\_costsheet\_conditions.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Set clrq costsheet conditions*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
    gt\_clrq\_cs\_komv = it\_komv.  
    gs\_tka01        = is\_tka01.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>SELECT\_AMOUNTS\_CALC\_GRP\_ROW*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IS\_CLRQHD                      TYPE        /CGDC/S\_CLRQHD*  
*\* | [--->] IT\_CRGRRF                      TYPE        /CGDC/TT\_CLRQGRRF*  
*\* | [--->] IS\_CRGRR                       TYPE        /CGDC/S\_CLRQGRR*  
*\* | [--->] I\_KSCHL                        TYPE        KSCHL*  
*\* | [--->] IV\_CONSTR\_DT                   TYPE        DATUM(optional)*  
*\* | [<-->] C\_VALUE                        TYPE        KWERT*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD select\_amounts\_calc\_grp\_row.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Select amounts calc grp row*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
    DATA: BEGIN OF ls\_quals\_key,  
            fcateg TYPE /cgdc/rtb\_qfcateg,  
            fname  TYPE /cgdc/rtb\_qfname,  
          END OF ls\_quals\_key.  
  
    DATA: lwa\_calc\_grp             TYPE /cgdc/s\_rtbpcg,  
          lwa\_calc\_group\_fun\_src   TYPE /cgdc/s\_rtbpcgfc,  
          lwa\_crgrrf               TYPE /cgdc/s\_clrqgrrf,  
          lv\_select\_clrq           TYPE string,  
*\*          lv\_select\_bs             TYPE string,*  
          lv\_select\_bs             TYPE xfeld,  
          lv\_final\_string          TYPE string,  
          lv\_pos\_string            TYPE string,  
          lv\_quals\_string          TYPE string,  
          lv\_neg\_string            TYPE string,  
          lv\_where\_clrq            TYPE string,  
          lv\_where\_for\_src\_as\_tgt  TYPE string,  
*\*          lv\_where\_bs               TYPE string,*  
          lv\_date\_where            TYPE string,  
          lv\_from\_table\_clrq       TYPE string,  
          lv\_from\_table\_bs         TYPE string,  
          ls\_quals\_key\_prev        LIKE ls\_quals\_key,  
          lt\_komv                  TYPE komv\_t,  
          lt\_komv\_bs               TYPE komv\_t,  
          lwa\_komv                 TYPE komv,  
          lwa\_komv\_bs              TYPE komv,  
          lwa\_xref\_fcat            TYPE /cgdc/\_xref\_fct,  
          lt\_source\_cond\_as\_trg    TYPE /cgdc/tt\_rtbpcgfc,  
          lwa\_source\_cond\_as\_trg   TYPE /cgdc/s\_rtbpcgf,  
          lt\_clgrp\_itm\_buffer      TYPE /cgdc/tt\_clrqit,  
          lt\_clgrp\_cnd\_buffer      TYPE /cgdc/tt\_clrqcn,  
          lt\_clgrp\_itm             TYPE /cgdc/tt\_clrqit,  
          lt\_clgrp\_cnd             TYPE /cgdc/tt\_clrqcn,  
          lt\_clrqit\_dfies          TYPE dfies\_tab,  
          lt\_clrq\_dfies            TYPE dfies\_tab,  
          lwa\_cgroup\_fun\_src\_quals TYPE /cgdc/t\_rpcgfsq,  
          lwa\_clgrp\_itm\_buffer     TYPE /cgdc/s\_clrqit,  
          lwa\_clgrp\_cnd\_buffer     TYPE /cgdc/s\_clrqcn,  
          lwa\_clgrp\_itm            TYPE /cgdc/s\_clrqit,  
          lwa\_clgrp\_cnd            TYPE /cgdc/s\_clrqcn,  
          lwa\_calc\_group\_fun       TYPE /cgdc/t\_rtbpcgf.  
  
    CALL METHOD select\_realtime\_billprf\_data  
      EXPORTING  
        i\_rtbpn = is\_clrqhd-rtbpn.  
  
    *"--SOC AMISHARMA - Read on Calc grp instead of condition.*  
*\*    READ TABLE gt\_calc\_group INTO lwa\_calc\_grp*  
*\*    WITH KEY kschl\_base = is\_crgrr-clgrp\_kschl.*  
  
    READ TABLE gt\_calc\_group INTO lwa\_calc\_grp  
    WITH KEY clgrn = is\_crgrr-clgrn.  
    *"--EOC AMISHARMA - Read on Calc grp instead of condition.*  
  
    IF i\_kschl = is\_crgrr-clgrp\_kschl.  
      IF lwa\_calc\_grp-bsmth = 'A'.  
        SELECT SINGLE kbetr FROM konp INTO c\_value  
          WHERE knumh = is\_crgrr-clgrp\_knumh.  
      ELSEIF lwa\_calc\_grp-bsmth = 'B'.  
        SELECT SINGLE \* FROM /cgdc/\_xref\_fct  
          INTO lwa\_xref\_fcat  
          WHERE ctype = is\_crgrr-clgrp\_kschl.  
        CASE lwa\_xref\_fcat-ftype.  
          WHEN 'C'.  
            SELECT SINGLE cost\_fund FROM /cgdc/\_cafdfund  
              INTO c\_value  
*\*              WHERE vbeln = is\_clrqhd-cntno*  
              WHERE vbeln = is\_clrqhd-vgbel  
                AND wbsel = is\_crgrr-wbsel.  
          WHEN 'F'.  
            SELECT SINGLE fix\_fee\_fund FROM /cgdc/\_cafdfund  
              INTO c\_value  
*\*              WHERE vbeln = is\_clrqhd-cntno*  
              WHERE vbeln = is\_clrqhd-vgbel  
                AND wbsel = is\_crgrr-wbsel.  
          WHEN 'I'.  
            SELECT SINGLE inc\_fee\_fund FROM /cgdc/\_cafdfund  
              INTO c\_value  
*\*              WHERE vbeln = is\_clrqhd-cntno*  
              WHERE vbeln = is\_clrqhd-vgbel  
                AND wbsel = is\_crgrr-wbsel.  
          WHEN 'A'.  
            SELECT SINGLE award\_fee\_fund FROM /cgdc/\_cafdfund  
              INTO c\_value  
*\*              WHERE vbeln = is\_clrqhd-cntno*  
              WHERE vbeln = is\_clrqhd-vgbel  
                AND wbsel = is\_crgrr-wbsel.  
        ENDCASE.  
      ENDIF.  
  
    ELSE.  
*\*\*\*DCLM 6686*  
      CLEAR: lt\_clrqit\_dfies[], lt\_clrq\_dfies[].  
  
      CALL METHOD /cgdc/cl\_clrq\_process=>get\_fieldinfo  
        EXPORTING  
          i\_tabname = '/CGDC/\_CLRQIT'  
        IMPORTING  
          et\_dfies  = lt\_clrqit\_dfies.  
  
      CALL METHOD /cgdc/cl\_clrq\_process=>get\_fieldinfo  
        EXPORTING  
          i\_tabname = '/CGDC/\_CLRQIC'  
        IMPORTING  
          et\_dfies  = lt\_clrq\_dfies.  
  
      APPEND LINES OF lt\_clrqit\_dfies TO lt\_clrq\_dfies.  
*\*\*\**  
      DATA: lv\_date TYPE sy-datum.  
      lv\_date = sy-datum.  
  
      CONCATENATE `"/CGDC/\_CLRQHD" AS A `  
        ` INNER JOIN "/CGDC/\_CLRQIT" AS B `  
        ` ON  A.CLRNO = B.CLRNO AND B.CLRSIT != '000000' `  
*\*\*\*DCLM 6686*  
        ` INNER JOIN "/CGDC/\_CLRQIC" AS IC `  
        ` ON  B.CLRNO = IC.CLRNO AND B.CLRIT = IC.CLRIT AND B.CLRSIT = IC.CLRSIT AND B.CLRDIT = IC.CLRDIT `  
*\*\*\**  
        ` INNER JOIN "/CGDC/\_CLRQCN" AS C `  
                     ` ON  B.CLRNO = C.CLRNO `  
                     ` AND B.CLRIT = C.CLRIT `  
                     ` AND B.CLRSIT = C.CLRSIT `  
                     ` AND B.CLRDIT = C.CLRDIT `  
                     ` AND C.KINAK = ' ' `  
        ` INNER JOIN "/CGDC/\_CLRQBI" AS D `  
                     ` ON  B.CLRNO = D.CLRNO `  
                     ` AND B.RTBBINR = D.BILL\_ITEM `  
                     ` AND B.BITYP = D.BITYPE `  
                     ` AND D.FAKSP = ' '`  
*\*                     ` AND D.BLDAT <= CURRENT\_DATE `*  
  
  
                     INTO lv\_from\_table\_clrq.  
  
      IF iv\_constr\_dt is NOT INITIAL.  
*\*        CONCATENATE lv\_from\_table\_clrq ` AND D.BLDAT <= IV\_CONSTR\_DT `*  
        lv\_from\_table\_clrq = | { lv\_from\_table\_clrq }  AND D.BLDAT <= '{ iv\_constr\_dt }' |.  
      ELSE.  
        lv\_from\_table\_clrq = | { lv\_from\_table\_clrq }  AND D.BLDAT <= CURRENT\_DATE |.  
  
      ENDIF.  
  
      CONCATENATE lv\_where\_clrq ` A.CLRNO = '` is\_clrqhd-clrno `'` INTO lv\_where\_clrq.  
      CONCATENATE lv\_where\_clrq ` AND ( (  B.ISTAT = 'A' ) or ( B.RTBBINR IS NOT NULL AND D.BSTAT != ' ' ) ) `  
      INTO lv\_where\_clrq .  
  
      CONCATENATE lv\_where\_for\_src\_as\_tgt ` CLRNO = '` is\_clrqhd-clrno `'` INTO lv\_where\_for\_src\_as\_tgt.  
  
      LOOP AT it\_crgrrf INTO lwa\_crgrrf  
        WHERE clgrn = is\_crgrr-clgrn.  
  
        IF lwa\_crgrrf-fcatg = 'HD'.  
          IF lwa\_crgrrf-fname CS '/'.  
            CONCATENATE lv\_where\_clrq ` AND A."` lwa\_crgrrf-fname `" = '` lwa\_crgrrf-fvalue `' ` INTO lv\_where\_clrq.  
          ELSE.  
            CONCATENATE lv\_where\_clrq ` AND A.` lwa\_crgrrf-fname ` = '` lwa\_crgrrf-fvalue `' ` INTO lv\_where\_clrq.  
          ENDIF.  
  
        ELSEIF lwa\_crgrrf-fcatg = 'IT'.  
*\*\*\*DCLM 6686*  
          READ TABLE lt\_clrq\_dfies WITH KEY fieldname = lwa\_crgrrf-fname INTO DATA(lwa\_clrq\_dfies).  
  
          IF lwa\_clrq\_dfies-tabname EQ '/CGDC/\_CLRQIT'.  
*\*\*\**  
            IF lwa\_crgrrf-fname CS '/'.  
              CONCATENATE lv\_where\_clrq ` AND B."` lwa\_crgrrf-fname `" = '` lwa\_crgrrf-fvalue `' ` INTO lv\_where\_clrq.  
            ELSE.  
              CONCATENATE lv\_where\_clrq ` AND B.` lwa\_crgrrf-fname ` = '` lwa\_crgrrf-fvalue `' ` INTO lv\_where\_clrq.  
            ENDIF.  
*\*\*\*DCLM 6686*  
          ELSEIF lwa\_clrq\_dfies-tabname EQ '/CGDC/\_CLRQIC'.  
  
            IF lwa\_crgrrf-fname CS '/'.  
              CONCATENATE lv\_where\_clrq ` AND IC."` lwa\_crgrrf-fname `" = '` lwa\_crgrrf-fvalue `' ` INTO lv\_where\_clrq.  
            ELSE.  
              CONCATENATE lv\_where\_clrq ` AND IC.` lwa\_crgrrf-fname ` = '` lwa\_crgrrf-fvalue `' ` INTO lv\_where\_clrq.  
            ENDIF.  
  
          ENDIF.  
*\*\*\**  
          CONCATENATE lv\_where\_for\_src\_as\_tgt ` AND ` lwa\_crgrrf-fname ` = '` lwa\_crgrrf-fvalue `' ` INTO lv\_where\_for\_src\_as\_tgt.  
  
        ENDIF.  
        IF lwa\_crgrrf-fname = 'DATAB' OR lwa\_crgrrf-fname = 'DATBI'.  
          IF lwa\_crgrrf-fname = 'DATAB'.  
            CONCATENATE lv\_date\_where ` AND B.PRSDT >= '`  lwa\_crgrrf-fvalue `' `  INTO lv\_date\_where.  
            CONCATENATE lv\_where\_for\_src\_as\_tgt ` AND PRSDT >= '`  lwa\_crgrrf-fvalue `' `  INTO lv\_where\_for\_src\_as\_tgt.  
          ENDIF.  
          IF lwa\_crgrrf-fname = 'DATBI'.  
            CONCATENATE lv\_date\_where ` AND B.PRSDT <= '`  lwa\_crgrrf-fvalue `' `  INTO lv\_date\_where.  
            CONCATENATE lv\_where\_for\_src\_as\_tgt ` AND PRSDT <= '`  lwa\_crgrrf-fvalue `' `  INTO lv\_where\_for\_src\_as\_tgt.  
          ENDIF.  
        ENDIF.  
      ENDLOOP.  
  
      CONCATENATE lv\_where\_clrq ` ` lv\_date\_where INTO lv\_where\_clrq.  
  
      LOOP AT gt\_calc\_group\_fun INTO lwa\_calc\_group\_fun WHERE clgrn = is\_crgrr-clgrn  
                                                          AND proc\_kschl = i\_kschl.  
  
        CLEAR: lv\_select\_clrq, lv\_select\_bs, lv\_final\_string, lv\_pos\_string, lv\_neg\_string, lv\_quals\_string.  
        lv\_final\_string = 'TO\_DECIMAL( SUM( CASE WHEN ( '.  
  
        LOOP AT gt\_calc\_group\_fun\_src INTO lwa\_calc\_group\_fun\_src  
          WHERE clgrn = is\_crgrr-clgrn  
            AND kschl\_func = lwa\_calc\_group\_fun-kschl\_func  
            AND proc\_kschl = i\_kschl.  
  
          CLEAR: lv\_quals\_string, ls\_quals\_key, ls\_quals\_key\_prev.  
          READ TABLE gt\_calc\_group\_fun  
          INTO lwa\_source\_cond\_as\_trg  
          WITH KEY trgt\_kschl = lwa\_calc\_group\_fun\_src-kschl\_src.  
  
          IF sy-subrc IS INITIAL.  
            IF NOT ( lwa\_source\_cond\_as\_trg-clgrn = is\_crgrr-clgrn  ).  
              APPEND lwa\_calc\_group\_fun\_src TO lt\_source\_cond\_as\_trg.  
            ENDIF.  
          ENDIF.  
  
          IF lwa\_calc\_group\_fun\_src-contr EQ '+'  
            OR lwa\_calc\_group\_fun\_src-contr EQ space.  
  
            IF lv\_pos\_string IS INITIAL.  
              CONCATENATE `( C.KSCHL = '` lwa\_calc\_group\_fun\_src-kschl\_src `'`  INTO lv\_pos\_string.  
            ELSE.  
              CONCATENATE lv\_pos\_string ` ) OR ( C.KSCHL = '` lwa\_calc\_group\_fun\_src-kschl\_src `'` INTO lv\_pos\_string.  
            ENDIF.  
  
            LOOP AT gt\_cgroup\_fun\_src\_quals INTO lwa\_cgroup\_fun\_src\_quals WHERE clgrn EQ lwa\_calc\_group\_fun\_src-clgrn  
                                                                            AND kschl\_func EQ lwa\_calc\_group\_fun\_src-kschl\_func  
                                                                            AND proc\_kschl EQ lwa\_calc\_group\_fun\_src-proc\_kschl  
                                                                            AND kschl\_src EQ lwa\_calc\_group\_fun\_src-kschl\_src  
                                                                            AND fname  IS NOT INITIAL.  
  
              MOVE-CORRESPONDING lwa\_cgroup\_fun\_src\_quals TO ls\_quals\_key.  
  
              IF ls\_quals\_key\_prev NE ls\_quals\_key .  
  
                IF lv\_quals\_string IS INITIAL.  
                  IF lwa\_cgroup\_fun\_src\_quals-fname CS '/'.  
                    CONCATENATE ` AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `."` lwa\_cgroup\_fun\_src\_quals-fname `" = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                    INTO lv\_quals\_string.  
                  ELSE.  
                    CONCATENATE ` AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `.` lwa\_cgroup\_fun\_src\_quals-fname ` = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                    INTO lv\_quals\_string.  
                  ENDIF.  
                ELSE.  
                  IF lwa\_cgroup\_fun\_src\_quals-fname CS '/'.  
                    CONCATENATE lv\_quals\_string ` ) AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `."` lwa\_cgroup\_fun\_src\_quals-fname `" = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                    INTO lv\_quals\_string.  
                  ELSE.  
                    CONCATENATE lv\_quals\_string ` ) AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `.` lwa\_cgroup\_fun\_src\_quals-fname ` = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                    INTO lv\_quals\_string.  
                  ENDIF.  
                ENDIF.  
                IF lwa\_cgroup\_fun\_src\_quals-fcateg EQ 'B'.  
                  CONCATENATE lv\_where\_for\_src\_as\_tgt ` AND ` lwa\_cgroup\_fun\_src\_quals-fname ` = '` lwa\_cgroup\_fun\_src\_quals-fvalue `' ` INTO lv\_where\_for\_src\_as\_tgt.  
                ENDIF.  
                ls\_quals\_key\_prev = ls\_quals\_key.  
  
              ELSE.  
  
                IF lwa\_cgroup\_fun\_src\_quals-fname CS '/'.  
                  CONCATENATE lv\_quals\_string ` OR ` lwa\_cgroup\_fun\_src\_quals-fcateg `."` lwa\_cgroup\_fun\_src\_quals-fname `" = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                      INTO lv\_quals\_string.  
                ELSE.  
                  CONCATENATE lv\_quals\_string ` OR ` lwa\_cgroup\_fun\_src\_quals-fcateg `.` lwa\_cgroup\_fun\_src\_quals-fname ` = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                  INTO lv\_quals\_string.  
                ENDIF.  
              ENDIF.  
  
            ENDLOOP.  
  
            IF lv\_quals\_string IS INITIAL.  
              *"++SOC - DCLM 5826 - Handling BSTAT*  
              IF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
                CONCATENATE lv\_quals\_string ` AND ( D.BSTAT >= '` lwa\_calc\_group\_fun\_src-bsfrm  
                `' AND D.BSTAT <= '`  lwa\_calc\_group\_fun\_src-bstto `' ` *"`' ) `*  
                INTO  lv\_quals\_string.  
                lv\_select\_bs = abap\_true.  
              ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
                CONCATENATE lv\_quals\_string ` AND ( D.BSTAT = '` lwa\_calc\_group\_fun\_src-bsfrm `' ` *" `' ) `*  
                INTO lv\_quals\_string.  
                lv\_select\_bs = abap\_true.  
              ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
                lv\_quals\_string = |{ lv\_quals\_string } AND ( D.BSTAT BETWEEN ' ' AND '{ lwa\_calc\_group\_fun\_src-bstto }' |.  
                lv\_select\_bs = abap\_true.  
              ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
                lv\_quals\_string = |{ lv\_quals\_string } AND ( D.BSTAT = ' ' |.  
                lv\_select\_bs = abap\_true.  
              ENDIF.  
  
            ELSE.  
  
              IF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
                CONCATENATE lv\_quals\_string ` ) AND ( D.BSTAT >= '` lwa\_calc\_group\_fun\_src-bsfrm  
                `' AND D.BSTAT <= '`  lwa\_calc\_group\_fun\_src-bstto `' ` *"`' ) `*  
                INTO  lv\_quals\_string.  
                lv\_select\_bs = abap\_true.  
              ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
                CONCATENATE lv\_quals\_string ` ) AND ( D.BSTAT = '` lwa\_calc\_group\_fun\_src-bsfrm `' ` *"`' ) `*  
                INTO lv\_quals\_string.  
                lv\_select\_bs = abap\_true.  
              ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
                lv\_quals\_string = |{ lv\_quals\_string } ) AND ( D.BSTAT BETWEEN ' ' AND '{ lwa\_calc\_group\_fun\_src-bstto }'|.  
                lv\_select\_bs = abap\_true.  
              ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
                lv\_quals\_string = |{ lv\_quals\_string } ) AND ( D.BSTAT = ' '|.  
                lv\_select\_bs = abap\_true.  
              ENDIF.  
              *"++EOC - DCLM 5826 - Handling BSTAT*  
            ENDIF.  
  
            IF NOT lv\_quals\_string IS INITIAL.  
              CONCATENATE lv\_pos\_string lv\_quals\_string ` ) ` INTO lv\_pos\_string SEPARATED BY space.  
            ENDIF.  
  
          ELSEIF lwa\_calc\_group\_fun\_src-contr EQ '-'.  
  
            IF lv\_neg\_string IS INITIAL.  
              CONCATENATE `( C.KSCHL = '` lwa\_calc\_group\_fun\_src-kschl\_src `'`  INTO lv\_neg\_string.  
            ELSE.  
              CONCATENATE lv\_neg\_string ` ) OR ( C.KSCHL = '` lwa\_calc\_group\_fun\_src-kschl\_src `'` INTO lv\_neg\_string.  
            ENDIF.  
  
            LOOP AT gt\_cgroup\_fun\_src\_quals INTO lwa\_cgroup\_fun\_src\_quals WHERE clgrn EQ lwa\_calc\_group\_fun\_src-clgrn  
                                                                            AND kschl\_func EQ lwa\_calc\_group\_fun\_src-kschl\_func  
                                                                            AND proc\_kschl EQ lwa\_calc\_group\_fun\_src-proc\_kschl  
                                                                            AND kschl\_src EQ lwa\_calc\_group\_fun\_src-kschl\_src.  
  
              MOVE-CORRESPONDING lwa\_cgroup\_fun\_src\_quals TO ls\_quals\_key.  
  
              IF ls\_quals\_key\_prev NE ls\_quals\_key.  
  
                IF lv\_quals\_string IS INITIAL.  
                  IF lwa\_cgroup\_fun\_src\_quals-fname CS '/'.  
                    CONCATENATE ` AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `."` lwa\_cgroup\_fun\_src\_quals-fname `" = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                    INTO lv\_quals\_string.  
                  ELSE.  
                    CONCATENATE ` AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `.` lwa\_cgroup\_fun\_src\_quals-fname ` = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                    INTO lv\_quals\_string.  
                  ENDIF.  
                ELSE.  
                  IF lwa\_cgroup\_fun\_src\_quals-fname CS '/'.  
                    CONCATENATE lv\_quals\_string ` ) AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `."` lwa\_cgroup\_fun\_src\_quals-fname `" = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                         INTO lv\_quals\_string.  
                  ELSE.  
                    CONCATENATE lv\_quals\_string ` ) AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `.` lwa\_cgroup\_fun\_src\_quals-fname ` = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                    INTO lv\_quals\_string.  
                  ENDIF.  
  
                ENDIF.  
  
              ELSE.  
                IF lwa\_cgroup\_fun\_src\_quals-fname CS '/'.  
                  CONCATENATE lv\_quals\_string ` OR ` lwa\_cgroup\_fun\_src\_quals-fcateg `."` lwa\_cgroup\_fun\_src\_quals-fname `" = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                       INTO lv\_quals\_string.  
                ELSE.  
                  CONCATENATE lv\_quals\_string ` OR ` lwa\_cgroup\_fun\_src\_quals-fcateg `.` lwa\_cgroup\_fun\_src\_quals-fname ` = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                  INTO lv\_quals\_string.  
                ENDIF.  
  
              ENDIF.  
  
            ENDLOOP.  
  
            IF lv\_quals\_string IS INITIAL.  
  
              IF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
                CONCATENATE lv\_quals\_string ` AND ( D.BSTAT >= '` lwa\_calc\_group\_fun\_src-bsfrm  
                `' AND D.BSTAT <= '`  lwa\_calc\_group\_fun\_src-bstto `' ` *"`' ) `*  
                INTO  lv\_quals\_string.  
                lv\_select\_bs = abap\_true.  
              ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
                CONCATENATE lv\_quals\_string ` AND ( D.BSTAT = '` lwa\_calc\_group\_fun\_src-bsfrm `' ` *"`' ) `*  
                INTO lv\_quals\_string.  
                lv\_select\_bs = abap\_true.  
                *"++SOC - DCLM 5826 - Handling BSTAT*  
              ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
                lv\_quals\_string = |{ lv\_quals\_string } AND ( D.BSTAT BETWEEN ' ' AND '{ lwa\_calc\_group\_fun\_src-bstto }'|.  
                lv\_select\_bs = abap\_true.  
              ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
                lv\_quals\_string = |{ lv\_quals\_string } AND ( D.BSTAT = ' '|.  
                lv\_select\_bs = abap\_true.  
              ENDIF.  
              *"++EOC - DCLM 5826 - Handling BSTAT*  
            ELSE.  
  
              IF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
                CONCATENATE lv\_quals\_string ` ) AND ( D.BSTAT >= '` lwa\_calc\_group\_fun\_src-bsfrm  
                `' AND D.BSTAT <= '`  lwa\_calc\_group\_fun\_src-bstto `' ` *"`' ) `*  
                INTO  lv\_quals\_string.  
                lv\_select\_bs = abap\_true.  
              ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
                CONCATENATE lv\_quals\_string ` ) AND ( D.BSTAT = '` lwa\_calc\_group\_fun\_src-bsfrm `' ` *"`' ) `*  
                INTO lv\_quals\_string.  
                lv\_select\_bs = abap\_true.  
                *"++SOC - DCLM 5826 - Handling BSTAT*  
              ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
                lv\_quals\_string = |{ lv\_quals\_string } ) AND ( D.BSTAT BETWEEN ' ' AND '{ lwa\_calc\_group\_fun\_src-bstto }'|.  
                lv\_select\_bs = abap\_true.  
              ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
                lv\_quals\_string = |{ lv\_quals\_string } ) AND ( D.BSTAT = ' '|.  
                lv\_select\_bs = abap\_true.  
              ENDIF.  
              *"++EOC - DCLM 5826 - Handling BSTAT*  
            ENDIF.  
  
            IF NOT lv\_quals\_string IS INITIAL.  
              CONCATENATE lv\_neg\_string lv\_quals\_string ` ) ` INTO lv\_neg\_string SEPARATED BY space.  
            ENDIF.  
  
          ENDIF.  
  
*\*          IF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.*  
*\*            IF lv\_select\_clrq IS  INITIAL.*  
*\*              CONCATENATE `SUM( CASE WHEN (  C.KSCHL = '` lwa\_calc\_group\_fun\_src-kschl\_src `'`  INTO lv\_select\_clrq.*  
*\*            ELSE.*  
*\*              CONCATENATE lv\_select\_clrq ` OR C.KSCHL = '` lwa\_calc\_group\_fun\_src-kschl\_src `'` INTO lv\_select\_clrq.*  
*\*            ENDIF.*  
*\*          ELSE.*  
*\*            IF lv\_select\_bs IS  INITIAL.*  
*\*              CONCATENATE `SUM( CASE WHEN ( ( C.KSCHL = '` lwa\_calc\_group\_fun\_src-kschl\_src `'`  INTO lv\_select\_bs.*  
*\*            ELSE.*  
*\*              CONCATENATE lv\_select\_bs ` OR ( ( C.KSCHL = '` lwa\_calc\_group\_fun\_src-kschl\_src `'` INTO lv\_select\_bs.*  
*\*            ENDIF.*  
*\**  
*\*            IF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.*  
*\*              CONCATENATE lv\_select\_bs ` AND B.BSTAT => '` lwa\_calc\_group\_fun\_src-bsfrm*  
*\*              `' AND B.BSTAT <= '`  lwa\_calc\_group\_fun\_src-bstto `' ) ` INTO  lv\_select\_bs.*  
*\*            ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL .*  
*\*              CONCATENATE lv\_select\_bs ` AND B.BSTAT = '` lwa\_calc\_group\_fun\_src-bsfrm `' ) ` INTO lv\_select\_bs.*  
*\*            ENDIF.*  
*\*          ENDIF.*  
  
        ENDLOOP.  
        IF lv\_pos\_string IS INITIAL AND lv\_neg\_string IS INITIAL..  
          EXIT.  
        ENDIF.  
*\*\*\*New changes*  
        IF lv\_pos\_string IS NOT INITIAL AND lv\_neg\_string IS NOT INITIAL.  
          CONCATENATE lv\_final\_string ` ` lv\_pos\_string ` ) ) THEN KWERT24 WHEN ( ` lv\_neg\_string ` ) ) THEN -1 \* KWERT24 ` INTO lv\_final\_string.  
        ELSEIF lv\_pos\_string IS NOT INITIAL AND lv\_neg\_string IS INITIAL.  
          CONCATENATE lv\_final\_string ` ` lv\_pos\_string ` ) ) THEN KWERT24 ` INTO lv\_final\_string.  
        ELSE.  
          CONCATENATE lv\_final\_string ` ` lv\_neg\_string ` ) ) THEN -1 \* KWERT24 ` INTO lv\_final\_string.  
        ENDIF.  
  
        CONCATENATE lv\_final\_string `ELSE 0 END ), 13, 2 ) AS KWERT ` INTO lv\_final\_string.  
  
  
        IF lv\_final\_string IS NOT INITIAL  
  
          AND lv\_from\_table\_clrq IS NOT INITIAL  
          AND lv\_where\_clrq IS NOT INITIAL.  
  
  
  
          IF NOT cl\_abap\_dbfeatures=>use\_features(  
                    EXPORTING  
                      requested\_features =  
                        VALUE #( ( cl\_abap\_dbfeatures=>call\_amdp\_method )  
                                 ( cl\_abap\_dbfeatures=>amdp\_table\_function ) ) ).  
            *"Error handling..*  
          ELSE.  
            TRY.  
                NEW /cgdc/cl\_clrq\_process\_amdp( )->get\_clrqsrc\_condition\_value(  
                  EXPORTING  
                    iv\_client          = sy-mandt  
                    iv\_select          = lv\_final\_string  
                    iv\_from\_table      = lv\_from\_table\_clrq  
                    iv\_where\_condition = lv\_where\_clrq  
                  IMPORTING  
                    et\_result          = DATA(lt\_data)  
                ).  
                lt\_komv = CORRESPONDING #( lt\_data ).  
              CATCH cx\_amdp\_error INTO DATA(amdp\_error).  
                *"error handling*  
            ENDTRY.  
  
          ENDIF.  
  
          READ TABLE lt\_komv INTO lwa\_komv INDEX 1.  
          c\_value = lwa\_komv-kwert.  
        ENDIF.  
*\**  
*\*        IF lv\_final\_string IS NOT INITIAL*  
*\*          AND lv\_select\_bs IS NOT INITIAL*  
*\*          AND lv\_from\_table\_bs IS NOT INITIAL*  
*\*          AND lv\_where\_clrq IS NOT INITIAL.*  
*\**  
*\*          SELECT (lv\_final\_string)*  
*\*            FROM (lv\_from\_table\_bs)*  
*\*            INTO CORRESPONDING FIELDS OF TABLE @lt\_komv\_bs*  
*\*            WHERE (lv\_where\_clrq).*  
*\**  
*\*          READ TABLE lt\_komv\_bs INTO lwa\_komv INDEX 1.*  
*\*          c\_value = lwa\_komv-kwert.*  
*\*        ENDIF.*  
  
        CALL FUNCTION '/CGDC/CLRQ\_DATA\_GET'  
          IMPORTING  
            et\_clgrp\_itm\_buffer = lt\_clgrp\_itm\_buffer  
            et\_clgrp\_cnd\_buffer = lt\_clgrp\_cnd\_buffer  
            et\_clgrp\_itm        = lt\_clgrp\_itm  
            et\_clgrp\_cnd        = lt\_clgrp\_cnd.  
  
        IF lt\_source\_cond\_as\_trg IS NOT INITIAL.  
          LOOP AT lt\_source\_cond\_as\_trg INTO lwa\_calc\_group\_fun\_src WHERE bsfrm IS INITIAL.  
  
            lv\_where\_for\_src\_as\_tgt = cl\_abap\_dyn\_prg=>escape\_quotes\_str( lv\_where\_for\_src\_as\_tgt ). *"ATC Correction*  
            LOOP AT lt\_clgrp\_itm\_buffer INTO lwa\_clgrp\_itm\_buffer WHERE (lv\_where\_for\_src\_as\_tgt).  
              LOOP AT lt\_clgrp\_cnd\_buffer INTO lwa\_clgrp\_cnd\_buffer WHERE clrit = lwa\_clgrp\_itm\_buffer-clrit AND kschl = lwa\_calc\_group\_fun\_src-kschl\_src.  
                IF lwa\_calc\_group\_fun\_src-contr = '-'.  
                  c\_value = c\_value - lwa\_clgrp\_cnd\_buffer-kwert.  
                ELSE.  
                  c\_value = c\_value + lwa\_clgrp\_cnd\_buffer-kwert.  
                ENDIF.  
              ENDLOOP.  
            ENDLOOP.  
*\*            READ TABLE lt\_clgrp\_cnd\_buffer INTO lwa\_clgrp\_cnd\_buffer*  
*\*            WITH KEY kschl = lwa\_calc\_group\_fun\_src-kschl\_src.*  
*\*            IF sy-subrc = 0 .*  
*\*            ENDIF.*  
          ENDLOOP.  
        ENDIF.  
  
        *"++SOC DCLM - 6655 Consider release Calc grp items runtime*  
        IF lt\_clgrp\_itm IS NOT INITIAL.  
          LOOP AT gt\_calc\_group\_fun INTO lwa\_calc\_group\_fun WHERE clgrn = is\_crgrr-clgrn  
                                                            AND   proc\_kschl = i\_kschl.  
            LOOP AT gt\_calc\_group\_fun\_src INTO lwa\_calc\_group\_fun\_src  
                                          WHERE clgrn = is\_crgrr-clgrn  
                                          AND kschl\_func = lwa\_calc\_group\_fun-kschl\_func  
                                          AND proc\_kschl = i\_kschl  
                                          AND bsfrm IS INITIAL.  
  
              lv\_where\_for\_src\_as\_tgt = cl\_abap\_dyn\_prg=>escape\_quotes\_str( lv\_where\_for\_src\_as\_tgt ). *"ATC Correction*  
              LOOP AT lt\_clgrp\_itm INTO lwa\_clgrp\_itm WHERE (lv\_where\_for\_src\_as\_tgt).  
                LOOP AT lt\_clgrp\_cnd INTO lwa\_clgrp\_cnd WHERE clrit = lwa\_clgrp\_itm-clrit AND  
                                                              kschl = lwa\_calc\_group\_fun\_src-kschl\_src.  
                  IF lwa\_calc\_group\_fun\_src-contr = '-'.  
                    c\_value = c\_value - lwa\_clgrp\_cnd-kwert.  
                  ELSE.  
                    c\_value = c\_value + lwa\_clgrp\_cnd-kwert.  
                  ENDIF.  
                ENDLOOP.  
              ENDLOOP.  
            ENDLOOP.  
          ENDLOOP.  
  
        ENDIF.  
        *"++EOC DCLM - 6655 Consider release Calc grp items runtime*  
  
        CLEAR:lwa\_komv,  
              lt\_komv.  
      ENDLOOP.  
  
    ENDIF.  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>GET\_COSTSHEET\_CONDITIONS*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [<---] ET\_KOMV                        TYPE        KOMV\_ITAB*  
*\* | [<---] ET\_KOMV\_ADD\_DATA               TYPE        /CGDC/TT\_CLRQ\_CS\_CON\_DATA*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD GET\_COSTSHEET\_CONDITIONS.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Get costsheet condition*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
   et\_komv = gt\_cost\_conditions.  
   et\_komv\_add\_data = gt\_cost\_conditions\_add.  
   CLEAR: gt\_cost\_conditions,  
          gt\_cost\_conditions\_add.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>GET\_CALC\_GRP\_PROC\_COND\_VAL*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] I\_KSCHL                        TYPE        KSCHL*  
*\* | [<---] E\_KWERT                        TYPE        KWERT*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD get\_calc\_grp\_proc\_cond\_val.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Get Calc grp proc cond val*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
    DATA: ls\_clrqhd    TYPE /cgdc/s\_clrqhd,  
          ls\_clrqit    TYPE /cgdc/s\_clrqit,  
          ls\_clrqgrr   TYPE /cgdc/s\_clrqgrr,  
          lt\_crgrrf    TYPE /cgdc/tt\_clrqgrrf,  
          lv\_constr\_dt TYPE sy-datum.  
  
    CALL FUNCTION '/CGDC/CLRQ\_DATA\_GET'  
      IMPORTING  
        es\_clrhd     = ls\_clrqhd  
        es\_clrit     = ls\_clrqit  
        es\_crgrr     = ls\_clrqgrr  
        et\_crgrrf    = lt\_crgrrf  
        ev\_constr\_dt = lv\_constr\_dt.  
    IF sy-subrc <> 0.  
    ENDIF.  
  
    CALL METHOD /cgdc/cl\_clrq\_process=>select\_amounts\_calc\_grp\_row  
      EXPORTING  
        is\_clrqhd    = ls\_clrqhd  
        it\_crgrrf    = lt\_crgrrf  
        is\_crgrr     = ls\_clrqgrr  
        i\_kschl      = i\_kschl  
        iv\_constr\_dt = lv\_constr\_dt  
      CHANGING  
        c\_value      = e\_kwert.  
  
  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>CHECK\_CALC\_GRP\_PROC\_COND*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] I\_KSCHL                        TYPE        KSCHL*  
*\* | [<---] E\_SUBRC                        TYPE        SYSUBRC*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD CHECK\_CALC\_GRP\_PROC\_COND.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Check calulation group conditions*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
    data: ls\_clrhd TYPE /cgdc/s\_clrqhd,  
          ls\_crgrr TYPE /cgdc/s\_clrqgrr.  
  
    e\_subrc = 4.  
  
    CALL FUNCTION '/CGDC/CLRQ\_DATA\_GET'  
     IMPORTING  
       ES\_CLRHD        = ls\_clrhd  
*\*       ES\_CLRIT        =*  
       ES\_CRGRR        = ls\_crgrr  
*\*       ET\_CRGRRF       =*  
              .  
    IF sy-subrc <> 0.  
    ENDIF.  
  
    IF ls\_clrhd-rtbpn IS NOT INITIAL.  
      CALL METHOD /cgdc/cl\_clrq\_process=>select\_realtime\_billprf\_data  
        EXPORTING  
          i\_rtbpn = ls\_clrhd-rtbpn  
          .  
      READ TABLE gt\_calc\_group\_fun  
      TRANSPORTING NO FIELDS  
      WITH KEY clgrn = ls\_crgrr-clgrn  
               proc\_kschl = i\_kschl.  
  
      e\_subrc = sy-subrc.  
    ENDIF.  
    IF sy-subrc NE 0.  
      READ TABLE gt\_calc\_group  
      TRANSPORTING NO FIELDS  
      WITH KEY clgrn = ls\_crgrr-clgrn  
               kschl\_base  = i\_kschl.  
  
      e\_subrc = sy-subrc.  
    ENDIF.  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>CHECK\_COSTING\_SHEET\_INTG*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] I\_OBJNR                        TYPE        OBJPO*  
*\* | [--->] I\_RTBPN                        TYPE        /CGDC/RTBPN*  
*\* | [<---] E\_STATUS                       TYPE        /CGDC/CLRQ\_CSIST*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD check\_costing\_sheet\_intg.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Check costing sheet intg*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*    data: lv\_costingsheet TYPE kalsm,*  
*\*          lwa\_prps TYPE prps,*  
*\*          lv\_subrc TYPE sy-subrc,*  
*\*          lv\_tabix TYPE sy-tabix,*  
*\*          lv\_status.*  
*\*    e\_status = 'X'.*  
*\*    IF i\_rtbpn IS NOT INITIAL.*  
*\*      SELECT SINGLE achin FROM /cgdc/t\_rtbp into lv\_status*  
*\*        WHERE rtbpn = i\_rtbpn.*  
*\*      IF lv\_status IS INITIAL.*  
*\*        e\_status = ' '. "No active costing sheet integration at RTB profile*  
*\*      ENDIF.*  
*\*    ENDIF.*  
*\**  
*\*    CHECK e\_status IS NOT INITIAL.*  
*\**  
*\*    READ TABLE gt\_prps*  
*\*    INTO lwa\_prps*  
*\*    WITH KEY objnr = i\_objnr*  
*\*    BINARY SEARCH.*  
*\*    IF sy-subrc NE 0.*  
*\*      lv\_subrc = sy-subrc.*  
*\*      lv\_tabix = sy-tabix.*  
*\**  
*\*      SELECT SINGLE \* FROM prps*  
*\*      INTO lwa\_prps*  
*\*      WHERE objnr = i\_objnr.*  
*\**  
*\*      IF lv\_subrc = 8.*  
*\*        append lwa\_prps to gt\_prps.*  
*\*      ELSEif lv\_subrc = 4  .*  
*\*        INSERT lwa\_prps into gt\_prps INDEX lv\_tabix.*  
*\*      ENDIF.*  
*\*    ENDIF.*  
*\**  
*\*    SELECT SINGLE \* FROM /cgdc/\_wbsmcs*  
*\*      INTO lwa\_wbsmcs*  
*\*      WHERE objnr = i\_objnr*  
*\*      AND ( begdat LE i\_ovrhd  AND enddat GE i\_ovrhd ).*  
*\**  
*\*    IF lwa\_wbsmcs-csheet IS NOT INITIAL.*  
*\*      lv\_costingsheet = lwa\_wbsmcs-csheet.*  
*\*    ELSE.*  
*\*      SELECT SINGLE \* FROM /cgdc/\_rtbclnmcs INTO lwa\_rtbclnmcs*  
*\*         WHERE vbeln = i\_vbeln*  
*\*           AND posnr = i\_posnr*  
*\*           AND ( begdat LE i\_ovrhd  AND enddat GE i\_ovrhd ).*  
*\**  
*\*      IF lwa\_rtbclnmcs-csheet IS NOT INITIAL.*  
*\*        lv\_costingsheet = lwa\_rtbclnmcs-csheet.*  
*\*      ELSE.*  
*\*        lv\_costingsheet  = lwa\_prps-kalsm.*  
*\*      ENDIF.*  
*\*    ENDIF.*  
*\**  
*\**  
*\*    IF lv\_costingsheet IS NOT INITIAL.*  
*\*      e\_status = 'A'.   "Active costing sheet integration at RTB profile and WBS.*  
*\*    ELSE.*  
*\*      e\_status = 'B'.   "No costing sheet at WBS*  
*\*    ENDIF.*  
*\**  
*\*\*    ELSE.*  
*\*\*      IF e\_status = 'C'.*  
*\*\*        e\_status = 'E'.   "No Active costing sheet integration at RTB profile and WBS*  
*\*\*      ELSE.*  
*\*\*        e\_status = 'B'. "No costing sheet at WBS*  
*\*\*      ENDIF.*  
*\*\*    ENDIF.*  
*\**  
*\**  
*\**  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>GET\_CLRQ\_COSTSHEET\_CONDITIONS*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] I\_KSCHL                        TYPE        KSCHL*  
*\* | [<---] ES\_KOMV                        TYPE        KOMV*  
*\* | [<---] ES\_TKA01                       TYPE        TKA01*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD get\_clrq\_costsheet\_conditions.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: CIS-AD: Get clrq costsheet conditions*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
    CLEAR: es\_komv, es\_tka01.  
    READ TABLE gt\_clrq\_cs\_komv INTO es\_komv  
    WITH KEY kschl = i\_kschl.  
  
    es\_tka01 = gs\_tka01.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>GET\_FEE\_VALUE*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] I\_KZWI1                        TYPE        KZWIS*  
*\* | [--->] I\_KZWI4                        TYPE        KZWIS*  
*\* | [--->] I\_KZWI5                        TYPE        KZWIS*  
*\* | [<-->] C\_XKWERT                       TYPE        KWERT*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD get\_fee\_value.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: CIS-AD: Get fee value*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
    Data: lv\_xkwert TYPE kwert.  
    lv\_xkwert = i\_kzwi1 \* ( i\_kzwi4 / i\_kzwi5 ).  
    c\_xkwert = lv\_xkwert.  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>PROCESS\_REALTIME\_DATA*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IT\_COEP                        TYPE        TT\_COEP(optional)*  
*\* | [--->] IT\_COBK                        TYPE        TTCOBK(optional)*  
*\* | [--->] IS\_BKPF                        TYPE        BKPF(optional)*  
*\* | [--->] IT\_ACDOCA                      TYPE        FINST\_ACDOCA(optional)*  
*\* | [--->] IV\_RTB\_EVENT                   TYPE        /CGDC/RTB\_EVENT(optional)*  
*\* | [--->] IT\_CLRQSRC\_DATA                TYPE        /CGDC/TT\_CLRQSRC\_DATA(optional)*  
*\* | [<---] ET\_RETURN                      TYPE        /CGDC/T\_MESSAGE*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD process\_realtime\_data.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Process COEP*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
    DATA : lt\_acdoca  TYPE TABLE OF acdoca,  
           lt\_coep    LIKE it\_coep,  
           lwa\_acdoca TYPE acdoca,  
           lwa\_coep   LIKE LINE OF it\_coep,  
           lt\_prps    TYPE TABLE OF prps,  
           lwa\_prps   TYPE prps.  
  
    DATA: lwa\_clrqsrc\_data TYPE /cgdc/s\_clrqsrc\_data,  
          lt\_clrqsrc\_data  TYPE /cgdc/tt\_clrqsrc\_data.  
  
    *"Source Replacement 5144: Validate Data against source conditions.*  
    TYPES: BEGIN OF lty\_data,  
             rtbpn TYPE /cgdc/rtbpn,  
             data  TYPE finst\_acdoca,  
           END OF lty\_data.  
  
    FIELD-SYMBOLS: <lt\_data\_tab>  TYPE table,  
                   <lt\_src\_valid> TYPE table,  
                   <lt\_input>     TYPE table.  
  
    DATA:  
      lt\_objnr         TYPE tt\_obj\_num,  
      lt\_acdoca\_vld    TYPE finst\_acdoca,  
      lt\_sources\_tab   TYPE /cgdc/tt\_source\_data,  
      lref\_valid\_lines TYPE REF TO data.  
  
  
    FIELD-SYMBOLS : <lt\_srcdata>  TYPE ANY TABLE,  
                    <lwa\_srcdata> TYPE any.  
  
    CONSTANTS : lc\_kb21n TYPE sytcode VALUE 'KB21N',  
                lc\_kb15n TYPE sytcode VALUE 'KB15N',  
                lc\_fb50  TYPE sytcode VALUE 'FB50',  
                lc\_migo  TYPE sytcode VALUE 'MIGO',  
                lc\_miro  TYPE sytcode VALUE 'MIRO'.  
  
*\*      "assign imported data for processing and set prim tab for processing.*  
*\*      "to be replaced with IT\_CLRQSRC\_DATA*  
*\*      READ TABLE it\_clrqsrc\_data INTO DATA(ls\_clr\_src) INDEX 1.*  
*\*      IF sy-subrc = 0.*  
*\*        ASSIGN ls\_clr\_src-data->\* TO <lt\_input>.*  
*\*        DATA(lv\_primtab) = CONV viewname( ls\_clr\_src-tabname ).*  
*\**  
*\*        IF <lt\_input> IS ASSIGNED.  "Source Replacement Validate Data against source conditions.*  
*\**  
*\*          CALL METHOD /cgdc/cl\_rtb\_source\_util=>find\_contr\_info\_realtime*  
*\*            EXPORTING*  
*\*              it\_data          = <lt\_input>                "replace with importing param*  
*\*              iv\_primtab       = lv\_primtab*  
*\*            IMPORTING*  
*\*              et\_result        = DATA(lt\_result)*  
*\*              et\_billelem      = DATA(lt\_billelem)*  
*\*              et\_contract      = DATA(lt\_contract)*  
*\*              et\_prps          = DATA(lt\_prps\_tab)*  
*\*              ev\_rel\_type      = DATA(lv\_rel\_type)*  
*\*              et\_realtime\_data = DATA(lt\_realtime\_data).*  
*\**  
*\*          " filter data by RTB profile*  
*\*          CREATE DATA lref\_valid\_lines TYPE TABLE OF (lv\_primtab).        "replace*  
*\*          ASSIGN lref\_valid\_lines->\* TO <lt\_src\_valid>.*  
*\**  
*\*          LOOP AT lt\_realtime\_data ASSIGNING FIELD-SYMBOL(<lfs\_realtime>)*  
*\*                                   WHERE rtbpn IS NOT INITIAL AND       "records can't be validated without RTB profile*  
*\*                                         vbeln IS NOT INITIAL AND*  
*\*                                         posnr IS NOT INITIAL.*  
*\**  
*\*            ASSIGN <lfs\_realtime>-data->\* TO <lt\_data\_tab>.*  
*\*            IF <lt\_data\_tab> IS ASSIGNED.*  
*\*            CALL METHOD /cgdc/cl\_rtb\_source\_util=>filter\_realtime\_data*  
*\*              EXPORTING*  
*\*                iv\_rtbpn    = <lfs\_realtime>-rtbpn*  
*\*                iv\_vbeln    = <lfs\_realtime>-vbeln*  
*\*                iv\_posnr    = <lfs\_realtime>-posnr*  
*\*                iv\_primtab  = lv\_primtab*  
*\*                it\_context\_data = it\_clrqsrc\_data*  
*\*              IMPORTING*  
*\*                et\_return       = DATA(lt\_return)*  
*\*                et\_sources      = DATA(lt\_sources)*  
*\*              CHANGING*  
*\*                ct\_src\_data     = <lt\_data\_tab>.*  
*\**  
*\*              IF <lt\_data\_tab> IS NOT INITIAL.*  
*\*                APPEND LINES OF <lt\_data\_tab> TO <lt\_src\_valid>.*  
*\*              ENDIF.*  
*\*              IF lt\_sources IS NOT INITIAL.*  
*\*                APPEND LINES OF lt\_sources TO lt\_sources\_tab.*  
*\*              ENDIF.*  
*\**  
*\*              UNASSIGN: <lt\_data\_tab>.*  
*\*              CLEAR: lt\_sources.*  
*\*            ENDIF.*  
*\*          ENDLOOP.*  
*\*        ENDIF.*  
*\**  
*\*        IF /cgdc/cl\_rtb\_event=>get\_event\_status( iv\_rtb\_tcode = sy-tcode ) = 'R'.*  
*\*          IF lt\_sources\_tab IS NOT INITIAL.*  
*\*            CALL FUNCTION '/CGDC/CLRQ\_RTBC\_PROCESS'*  
*\*              EXPORTING*  
*\*\*               i\_vbeln              =*  
*\*                it\_src\_table         = <lt\_src\_valid>*  
*\*                i\_process            = 'B'*  
*\*                i\_commit\_work        = ' '*  
*\*                it\_realtime\_billelem = lt\_billelem*  
*\*                it\_realtime\_prps     = lt\_prps\_tab*  
*\*                it\_realtime\_contr    = lt\_contract*  
*\*                i\_realtime\_primtab   = lv\_primtab*  
*\*                it\_realtime\_sources  = lt\_sources\_tab*  
*\*              IMPORTING*  
*\*                et\_message           = et\_return*  
*\*              EXCEPTIONS*  
*\*                calc\_req\_locked      = 1.*  
*\*            IF sy-subrc <> 0.*  
*\*            ENDIF.*  
*\*          ENDIF.*  
*\*        ENDIF.*  
*\*      ENDIF.*  
  
  
    *""COPIED*  
    *"assign imported data for processing and set prim tab for processing.*  
    *"to be replaced with IT\_CLRQSRC\_DATA*  
    LOOP AT it\_clrqsrc\_data INTO DATA(ls\_clr\_src). *" INDEX 1.*  
      ASSIGN ls\_clr\_src-data->\* TO <lt\_input>.  
      DATA(lv\_primtab) = CONV viewname( ls\_clr\_src-tabname ).  
  
      IF <lt\_input> IS ASSIGNED.  *"Source Replacement Validate Data against source conditions.*  
  
        CALL METHOD /cgdc/cl\_rtb\_source\_util=>find\_contr\_info\_realtime  
          EXPORTING  
            it\_data          = <lt\_input>                *"replace with importing param*  
            iv\_primtab       = lv\_primtab  
          IMPORTING  
            et\_result        = DATA(lt\_result)  
            et\_billelem      = DATA(lt\_billelem)  
            et\_contract      = DATA(lt\_contract)  
            et\_prps          = DATA(lt\_prps\_tab)  
            ev\_rel\_type      = DATA(lv\_rel\_type)  
            et\_realtime\_data = DATA(lt\_realtime\_data).  
  
        *" filter data by RTB profile*  
        CREATE DATA lref\_valid\_lines TYPE TABLE OF (lv\_primtab).        *"replace*  
        ASSIGN lref\_valid\_lines->\* TO <lt\_src\_valid>.  
  
        LOOP AT lt\_realtime\_data ASSIGNING FIELD-SYMBOL(<lfs\_realtime>)  
                                 WHERE rtbpn IS NOT INITIAL AND       *"records can't be validated without RTB profile*  
                                       vbeln IS NOT INITIAL AND  
                                       posnr IS NOT INITIAL.  
  
          ASSIGN <lfs\_realtime>-data->\* TO <lt\_data\_tab>.  
          IF <lt\_data\_tab> IS ASSIGNED.  
            CALL METHOD /cgdc/cl\_rtb\_source\_util=>filter\_realtime\_data01 *"" CREATED DUPLICARE*  
              EXPORTING  
                iv\_rtbpn        = <lfs\_realtime>-rtbpn  
                iv\_vbeln        = <lfs\_realtime>-vbeln  
                iv\_posnr        = <lfs\_realtime>-posnr  
                iv\_primtab      = lv\_primtab  
                it\_context\_data = it\_clrqsrc\_data *"PASS ACDOCA/COEP Directly mentioned in Call*  
              IMPORTING  
                et\_return       = DATA(lt\_return)  
                et\_sources      = DATA(lt\_sources)  
              CHANGING  
                ct\_src\_data     = <lt\_data\_tab>.  
  
            IF <lt\_data\_tab> IS NOT INITIAL.  
              APPEND LINES OF <lt\_data\_tab> TO <lt\_src\_valid>.  
            ENDIF.  
            IF lt\_sources IS NOT INITIAL.  
              APPEND LINES OF lt\_sources TO lt\_sources\_tab.  
            ENDIF.  
  
            UNASSIGN: <lt\_data\_tab>.  
            CLEAR: lt\_sources.  
          ENDIF.  
        ENDLOOP.  
      ENDIF.  
  
      IF /cgdc/cl\_rtb\_event=>get\_event\_status( iv\_rtb\_tcode = sy-tcode ) = 'R'.  
        IF lt\_sources\_tab IS NOT INITIAL.  
          CALL FUNCTION '/CGDC/CLRQ\_RTBC\_PROCESS'  
            EXPORTING  
*\*             i\_vbeln              =*  
              it\_src\_table         = <lt\_src\_valid>  
              i\_process            = 'B'  
              i\_commit\_work        = ' '  
              it\_realtime\_billelem = lt\_billelem  
              it\_realtime\_prps     = lt\_prps\_tab  
              it\_realtime\_contr    = lt\_contract  
              i\_realtime\_primtab   = lv\_primtab  
              it\_realtime\_sources  = lt\_sources\_tab  
            IMPORTING  
              et\_message           = et\_return  
            EXCEPTIONS  
              calc\_req\_locked      = 1.  
          IF sy-subrc <> 0.  
          ENDIF.  
        ENDIF.  
      ENDIF.  
    ENDLOOP.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>SELECT\_REALTIME\_BILLPRF\_DATA*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] I\_RTBPN                        TYPE        /CGDC/RTBPN*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD select\_realtime\_billprf\_data.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Select realtime billing profile data*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
    CHECK i\_rtbpn NE gs\_rtbp-rtbpn.  
  
    SELECT SINGLE \* FROM /cgdc/t\_rtbp  
      INTO gs\_rtbp  
      WHERE rtbpn = i\_rtbpn.  
  
    SELECT a~\* FROM  
      /cgdc/t\_rtbpcg AS a INNER JOIN /cgdc/t\_asgncg AS b  
      ON a~clgrn = b~clgrn  
      INTO TABLE @gt\_calc\_group  
      WHERE b~rtbpn = @i\_rtbpn.  
  
    SELECT a~\* FROM  
      /cgdc/t\_rtbpcgf AS a INNER JOIN /cgdc/t\_asgncg AS b  
      ON a~clgrn = b~clgrn  
      INTO TABLE @gt\_calc\_group\_fun  
      WHERE b~rtbpn = @i\_rtbpn.  
  
    SELECT a~\* FROM  
      /cgdc/t\_rtbpcgfc AS a INNER JOIN /cgdc/t\_asgncg AS b  
      ON a~clgrn = b~clgrn  
      INTO TABLE @gt\_calc\_group\_fun\_src  
      WHERE b~rtbpn = @i\_rtbpn.  
  
    IF sy-subrc EQ 0.  
  
      SELECT a~\* FROM  
        /cgdc/t\_rpcgfsq AS a INNER JOIN /cgdc/t\_asgncg AS b  
        ON a~clgrn = b~clgrn  
        INTO TABLE @gt\_cgroup\_fun\_src\_quals  
        FOR ALL ENTRIES IN @gt\_calc\_group\_fun\_src  
        WHERE b~rtbpn EQ @i\_rtbpn  
          AND a~clgrn EQ @gt\_calc\_group\_fun\_src-clgrn  
          AND a~kschl\_func EQ @gt\_calc\_group\_fun\_src-kschl\_func  
          AND a~proc\_kschl EQ @gt\_calc\_group\_fun\_src-proc\_kschl  
          AND a~kschl\_src EQ @gt\_calc\_group\_fun\_src-kschl\_src.  
  
    ENDIF.  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>SUBSEQUENT\_DOC\_DATA\_UPDATE*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IT\_VBRK                        TYPE        VBRKVB\_T*  
*\* | [--->] IT\_VBRP                        TYPE        VBRPVB\_T*  
*\* | [--->] IV\_PROCESS                     TYPE        CHAR1(optional)*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD subsequent\_doc\_data\_update.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Workflow of BDR Update*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*\*\*\*Purpose of IV\_PROCESS valus*  
*\*\*\*\*'D' BDR is getting deleted or rejected so make calc request item Open*  
*\*\*\*\*'E' PBD is getting deleted or rejected so make calc request item Open*  
*\*\*\*\*When others PBD and Invoice creation*  
    CONSTANTS : lc\_pbd(3) VALUE 'PBD'.  
    CONSTANTS : lc\_vf11(4) VALUE 'VF11'.  
  
    DATA: lt\_clrqdlf TYPE TABLE OF /cgdc/\_clrqdf.  
    DATA: lwa\_clrqdlf TYPE /cgdc/\_clrqdf.  
    DATA: lwa\_clrqdoc TYPE /cgdc/s\_clrq\_doc,  
          lt\_critm    TYPE /cgdc/tt\_clrqit.  
    DATA: lt\_clrqdoc TYPE TABLE OF /cgdc/s\_clrq\_doc.  
    DATA: lwa\_vbrk         LIKE LINE OF it\_vbrk,  
          lt\_vbrp          LIKE it\_vbrp,  
          lt\_dfies\_clrqbsi TYPE dfies\_tab,  
          lt\_dfies\_clrqbsc TYPE dfies\_tab,  
*\*          lt\_clrqbsi       TYPE /cgdc/tt\_clrqbsi,*  
          lwa\_vbrp         LIKE LINE OF it\_vbrp,  
          lwa\_dfies        TYPE dfies,  
*\*          lwa\_clrqbsi      TYPE /cgdc/s\_clrqbsi,*  
*\*          ls\_clrqbs\_doc    TYPE /cgdc/s\_clrqbs\_doc,*  
          lv\_prev\_vbeln    TYPE vbeln\_va,  
          lv\_tabix         TYPE sy-tabix.  
  
    DATA: BEGIN OF ls\_temp\_no ,  
            id(1) TYPE c  VALUE '$',  
            nr(9) TYPE n,  
          END OF ls\_temp\_no.  
  
  
    DATA: lt\_clrqdf\_new TYPE TABLE OF /cgdc/s\_clrqdf.  
    FIELD-SYMBOLS : <lwa\_clrqdf> TYPE /cgdc/s\_clrqdf.  
    FIELD-SYMBOLS : <lv\_value>           TYPE any,  
                    <lwa\_clrqit>         TYPE /cgdc/s\_clrqit,  
*\*                    <lwa\_clrqbsi>        TYPE /cgdc/s\_clrqbsi,*  
*\*                    <lwa\_clrqbsc>        TYPE /cgdc/s\_clrqbsc,*  
                    <lwa\_clrqit\_0subitm> TYPE /cgdc/s\_clrqit,  
                    <lwa\_clrqbi>         TYPE /cgdc/s\_clrqbi.  
  
    CHECK it\_vbrk IS NOT INITIAL.  
  
    DATA(lv\_vbeln) = VALUE #( it\_vbrk[ vbtyp = lc\_pbd ]-vbeln OPTIONAL ).  
  
    SELECT SINGLE pbd\_status, apm\_approval\_status  
     FROM vbrk  
     WHERE vbeln = @lv\_vbeln  
     AND vbtyp = @lc\_pbd  
     INTO (@DATA(lv\_pbdstat), @DATA(lv\_apr\_stat)) .  
  
    SELECT \*  
      FROM /cgdc/\_clrqdf  
      INTO TABLE lt\_clrqdlf  
      FOR ALL ENTRIES IN it\_vbrk  
      WHERE sbqdc = it\_vbrk-vbeln.  
  
    DATA(lv\_rfbsk) = VALUE #( it\_vbrk[ 1 ]-rfbsk OPTIONAL ).  
  
  
    IF sy-subrc = 0.  
  
      SELECT \*  
        FROM /cgdc/\_clrqbi  
        FOR ALL ENTRIES IN @lt\_clrqdlf  
        WHERE clrno     = @lt\_clrqdlf-clrno  
        AND   bill\_item = @lt\_clrqdlf-rtbbinr  
        INTO TABLE @DATA(lt\_clrqbi).  
  
        MOVE-CORRESPONDING lt\_clrqbi TO lwa\_clrqdoc-x-crbinr.  
  
    ENDIF.  
    IF sy-subrc = 0.  
  
*\*      SELECT \**  
*\*        FROM /cgdc/\_clrqbsi*  
*\*        INTO TABLE ls\_clrqbs\_doc-x-clrqbsi*  
*\*        FOR ALL ENTRIES IN it\_vbrp*  
*\*        WHERE xrfdn EQ it\_vbrp-vgbel*  
*\*          AND xrfin EQ it\_vbrp-vgpos.*  
*\**  
*\*      SELECT \**  
*\*        FROM /cgdc/\_clrqit*  
*\*        INTO CORRESPONDING FIELDS OF TABLE lwa\_clrqdoc-x-critm*  
*\*        FOR ALL ENTRIES IN lt\_clrqdlf*  
*\*        WHERE clrno = lt\_clrqdlf-clrno*  
*\*        AND clrit = lt\_clrqdlf-clrit*  
*\*        AND clrsit = lt\_clrqdlf-clrsit*  
*\*        AND clrdit = lt\_clrqdlf-clrdit.*  
*\**  
*\*\*\*\*New status changes*  
*\*      SELECT \**  
*\*        FROM /cgdc/\_clrqit*  
*\*        INTO CORRESPONDING FIELDS OF TABLE lwa\_clrqdoc-x-critm*  
*\*        FOR ALL ENTRIES IN lt\_clrqdlf*  
*\*        WHERE clrno = lt\_clrqdlf-clrno*  
*\*        AND clrit = lt\_clrqdlf-clrit*  
*\*        AND clrsit = lt\_clrqdlf-clrsit*  
*\*        AND clrdit = lt\_clrqdlf-clrdit.*  
  
*\*      SELECT \**  
*\*        FROM /cgdc/\_clrqit*  
*\*        APPENDING CORRESPONDING FIELDS OF TABLE lt\_critm*  
*\*        FOR ALL ENTRIES IN lt\_clrqdlf*  
*\*        WHERE clrno = lt\_clrqdlf-clrno*  
*\*        AND clrit = lt\_clrqdlf-clrit*  
*\*        AND clrsit = '000000'.*  
*\*\**  
      CASE   iv\_process.  
        WHEN 'D'.  
*\*          LOOP AT lwa\_clrqdoc-x-critm ASSIGNING <lwa\_clrqit>.*  
*\*\*\*\*New status changes*  
*\*\*            <lwa\_clrqit>-bstat = ' '.*  
*\*            <lwa\_clrqit>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-open.*  
*\*\*\*\**  
*\*            <lwa\_clrqit>-updkz = 'U'.*  
*\*            UNASSIGN: <lwa\_clrqit\_0subitm>.*  
*\*            READ TABLE lt\_critm ASSIGNING <lwa\_clrqit\_0subitm>*  
*\*            WITH KEY  clrno = <lwa\_clrqit>-clrno*  
*\*                      clrit = <lwa\_clrqit>-clrit*  
*\*                      clrsit = '000000'.*  
*\*            IF <lwa\_clrqit\_0subitm> IS ASSIGNED.*  
*\*              <lwa\_clrqit\_0subitm>-billed\_amt = <lwa\_clrqit\_0subitm>-billed\_amt - <lwa\_clrqit>-netwr.*  
*\*              <lwa\_clrqit\_0subitm>-billed\_qty = <lwa\_clrqit\_0subitm>-billed\_qty - <lwa\_clrqit>-zmeng.*  
*\*              <lwa\_clrqit\_0subitm>-updkz = 'U'.*  
*\*            ENDIF.*  
*\*          ENDLOOP.*  
*\*\*\*New status changes*  
          LOOP AT lwa\_clrqdoc-x-crbinr ASSIGNING <lwa\_clrqbi>.  
            <lwa\_clrqbi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-open.  
            <lwa\_clrqbi>-updkz = 'U'.  
          ENDLOOP.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Begin: Commented by Yunus - Billing item Changes\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*          IF NOT ls\_clrqbs\_doc-x-clrqbsi IS INITIAL.*  
*\**  
*\*            READ TABLE ls\_clrqbs\_doc-x-clrqbsi INTO lwa\_clrqbsi INDEX 1.*  
*\**  
*\*            IF sy-subrc EQ 0.*  
*\**  
*\*              CALL FUNCTION 'DDIF\_FIELDINFO\_GET'*  
*\*                EXPORTING*  
*\*                  tabname        = '/CGDC/\_CLRQBSI'*  
*\*                  langu          = sy-langu*  
*\*                  all\_types      = abap\_true*  
*\*                TABLES*  
*\*                  dfies\_tab      = lt\_dfies\_clrqbsi*  
*\*                EXCEPTIONS*  
*\*                  not\_found      = 1*  
*\*                  internal\_error = 2*  
*\*                  OTHERS         = 3.*  
*\**  
*\*              CALL FUNCTION 'DDIF\_FIELDINFO\_GET'*  
*\*                EXPORTING*  
*\*                  tabname        = '/CGDC/\_CLRQBSC'*  
*\*                  langu          = sy-langu*  
*\*                  all\_types      = abap\_true*  
*\*                TABLES*  
*\*                  dfies\_tab      = lt\_dfies\_clrqbsc*  
*\*                EXCEPTIONS*  
*\*                  not\_found      = 1*  
*\*                  internal\_error = 2*  
*\*                  OTHERS         = 3.*  
*\**  
*\*              SELECT SINGLE \**  
*\*                FROM /cgdc/\_clrqbsh*  
*\*                INTO ls\_clrqbs\_doc-x-clrqbsh*  
*\*                WHERE clrno EQ lwa\_clrqbsi-clrno*  
*\*                  AND blsno EQ lwa\_clrqbsi-blsno.*  
*\**  
*\*              SELECT \**  
*\*                FROM /cgdc/\_clrqbsc*  
*\*                INTO TABLE ls\_clrqbs\_doc-x-clrqbsc*  
*\*                FOR ALL ENTRIES IN ls\_clrqbs\_doc-x-clrqbsi*  
*\*                WHERE clrno EQ ls\_clrqbs\_doc-x-clrqbsi-clrno*  
*\*                  AND blsno EQ ls\_clrqbs\_doc-x-clrqbsi-blsno*  
*\*                  AND xrfdn EQ ls\_clrqbs\_doc-x-clrqbsi-xrfdn*  
*\*                  AND xrfin EQ ls\_clrqbs\_doc-x-clrqbsi-xrfin.*  
*\**  
*\*              ls\_clrqbs\_doc-x-clrqbsh-billcount = ls\_clrqbs\_doc-x-clrqbsh-billcount + 1.*  
*\*              ls\_clrqbs\_doc-x-clrqbsh-negval = abap\_true.*  
*\*              ls\_clrqbs\_doc-x-clrqbsh-updkz = 'I'.*  
*\*              cl\_system\_uuid=>if\_system\_uuid\_static~create\_uuid\_c22( RECEIVING uuid = ls\_clrqbs\_doc-x-clrqbsh-blsno ).*  
*\*              CLEAR lt\_clrqbsi[].*  
*\**  
*\*              LOOP AT lwa\_clrqdoc-x-critm ASSIGNING <lwa\_clrqit>.*  
*\*                <lwa\_clrqit>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-open.*  
*\*                <lwa\_clrqit>-updkz = 'U'.*  
*\*              ENDLOOP.*  
*\**  
*\*              LOOP AT ls\_clrqbs\_doc-x-clrqbsi ASSIGNING <lwa\_clrqbsi>.*  
*\*                <lwa\_clrqbsi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-open.*  
*\*                <lwa\_clrqbsi>-updkz = 'U'.*  
*\*              ENDLOOP.*  
*\**  
*\*              LOOP AT ls\_clrqbs\_doc-x-clrqbsi INTO lwa\_clrqbsi.*  
*\**  
*\*                LOOP AT lt\_dfies\_clrqbsi INTO lwa\_dfies.*  
*\*                  CHECK lwa\_dfies-datatype EQ 'CURR' OR lwa\_dfies-datatype EQ 'QUAN'.*  
*\*                  UNASSIGN <lv\_value>.*  
*\*                  ASSIGN COMPONENT lwa\_dfies-fieldname OF STRUCTURE lwa\_clrqbsi TO <lv\_value>.*  
*\*                  CHECK <lv\_value> IS ASSIGNED.*  
*\*                  <lv\_value> = -1 \* <lv\_value>.*  
*\*                ENDLOOP.*  
*\**  
*\*                lwa\_clrqbsi-blsno = ls\_clrqbs\_doc-x-clrqbsh-blsno.*  
*\*                lwa\_clrqbsi-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-open.*  
*\*                lwa\_clrqbsi-updkz = 'I'.*  
*\*                APPEND lwa\_clrqbsi TO lt\_clrqbsi.*  
*\**  
*\*              ENDLOOP.*  
*\**  
*\*              IF NOT lt\_clrqbsi IS INITIAL.*  
*\*                APPEND LINES OF lt\_clrqbsi TO ls\_clrqbs\_doc-x-clrqbsi.*  
*\*              ENDIF.*  
*\**  
*\*              LOOP AT ls\_clrqbs\_doc-x-clrqbsc ASSIGNING <lwa\_clrqbsc>.*  
*\**  
*\*                LOOP AT lt\_dfies\_clrqbsc INTO lwa\_dfies.*  
*\*                  CHECK lwa\_dfies-datatype EQ 'CURR' OR lwa\_dfies-datatype EQ 'QUAN'.*  
*\*                  UNASSIGN <lv\_value>.*  
*\*                  ASSIGN COMPONENT lwa\_dfies-fieldname OF STRUCTURE <lwa\_clrqbsc> TO <lv\_value>.*  
*\*                  CHECK <lv\_value> IS ASSIGNED.*  
*\*                  <lv\_value> = -1 \* <lv\_value>.*  
*\*                ENDLOOP.*  
*\**  
*\*                <lwa\_clrqbsc>-blsno = ls\_clrqbs\_doc-x-clrqbsh-blsno.*  
*\*                <lwa\_clrqbsc>-updkz = 'I'.*  
*\**  
*\*              ENDLOOP.*  
*\**  
*\*            ENDIF.*  
*\**  
*\*          ENDIF.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End: Commented by Yunus - Billing item Changes\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*\*\**  
        WHEN 'E'.  
*\*\*New status changes*  
*\*          LOOP AT lwa\_clrqdoc-x-critm ASSIGNING <lwa\_clrqit>.*  
*\*\*\*New status changes*  
*\*\*            <lwa\_clrqit>-bstat = 'Q'.*  
*\*            <lwa\_clrqit>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-billing\_req.*  
*\*\*\*\**  
*\*            <lwa\_clrqit>-updkz = 'U'.*  
*\*          ENDLOOP.*  
*\**  
*\*          LOOP AT ls\_clrqbs\_doc-x-clrqbsi ASSIGNING <lwa\_clrqbsi>.*  
*\*            <lwa\_clrqbsi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-billing\_req.*  
*\*            <lwa\_clrqbsi>-updkz = 'U'.*  
*\*          ENDLOOP.*  
  
          LOOP AT lwa\_clrqdoc-x-crbinr ASSIGNING <lwa\_clrqbi>.  
            <lwa\_clrqbi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-billing\_req.  
            <lwa\_clrqbi>-updkz = 'U'.  
          ENDLOOP.  
*\*\*\**  
        WHEN 'R'.  
  
  
          CLEAR lwa\_vbrk.  
          READ TABLE it\_vbrk INTO lwa\_vbrk INDEX 1.  
  
          LOOP AT lwa\_clrqdoc-x-crbinr ASSIGNING <lwa\_clrqbi>.  
  
            <lwa\_clrqbi>-bstat = COND #( WHEN lwa\_vbrk-rfbsk CA ' AB'  
                                         THEN /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-invoice\_created  
                                         WHEN lwa\_vbrk-rfbsk CA 'CDH'  
                                         THEN /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-invoice\_posted  
                                         WHEN lwa\_vbrk-rfbsk CA 'FGIK'  
                                         THEN /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-erroneous\_invoice  ) .  
            <lwa\_clrqbi>-updkz = 'U'.  
          ENDLOOP.  
  
        WHEN OTHERS.  
          lt\_vbrp = it\_vbrp.  
          SORT lt\_vbrp BY vgbel vbeln.  
          DELETE ADJACENT DUPLICATES FROM  lt\_vbrp COMPARING vgbel vbeln.  
          LOOP AT lt\_vbrp INTO lwa\_vbrp .  
            ls\_temp\_no = lwa\_vbrp-vbeln.  
*\*\*\*New status changes*  
            IF ls\_temp\_no-id = '$'.  
*\*\*\**  
              lv\_tabix = ls\_temp\_no-nr.  
              READ TABLE it\_vbrk INTO lwa\_vbrk INDEX lv\_tabix.  
             DATA(lv\_sbqdc) = cond #( WHEN lwa\_vbrp-pbd\_id IS INITIAL  
                                      THEN lwa\_vbrp-vgbel ELSE lwa\_vbrp-pbd\_id ).  
              LOOP AT lt\_clrqdlf INTO lwa\_clrqdlf WHERE sbqdc = lv\_sbqdc.  
                APPEND INITIAL LINE TO lwa\_clrqdoc-x-crdfl ASSIGNING <lwa\_clrqdf>.  
                IF <lwa\_clrqdf> IS ASSIGNED.  
                  MOVE-CORRESPONDING lwa\_clrqdlf TO <lwa\_clrqdf>.  
                  <lwa\_clrqdf>-prcdc = lwa\_clrqdlf-sbqdc.  
                  <lwa\_clrqdf>-prcit = lwa\_clrqdlf-sbqit.  
                  <lwa\_clrqdf>-sbqdc = lwa\_vbrk-vbeln.  
                  <lwa\_clrqdf>-sbqty = lwa\_vbrk-fkart.  
                  <lwa\_clrqdf>-updkz = 'I'.  
                  <lwa\_clrqdf>-erdat = sy-datum.  
                  <lwa\_clrqdf>-ernam = sy-uname.  
                  <lwa\_clrqdf>-erzet = sy-uzeit.  
  
*\*                  UNASSIGN <lwa\_clrqbsi>.  "clrqbsi deleted*  
*\*                  READ TABLE ls\_clrqbs\_doc-x-clrqbsi*  
*\*                  ASSIGNING <lwa\_clrqbsi>*  
*\*                  WITH KEY clrno = <lwa\_clrqdf>-clrno*  
*\*                           xrfdn = <lwa\_clrqdf>-xrfdn*  
*\*                           xrfin = <lwa\_clrqdf>-xrfin.*  
*\*\*\*New status changes*  
                  IF <lwa\_clrqbi> IS ASSIGNED.  
                    UNASSIGN <lwa\_clrqbi>.  
                  ENDIF.  
                  READ TABLE lwa\_clrqdoc-x-crbinr  
                  ASSIGNING <lwa\_clrqbi>  
                  WITH KEY clrno     = <lwa\_clrqdf>-clrno  
                           bill\_item = <lwa\_clrqdf>-rtbbinr.  
*\*                  UNASSIGN <lwa\_clrqit> .*  
*\*                  READ TABLE lwa\_clrqdoc-x-critm ASSIGNING <lwa\_clrqit> WITH KEY clrno = <lwa\_clrqdf>-clrno*  
*\*                                                                         clrit = <lwa\_clrqdf>-clrit*  
*\*                                                                         clrsit = <lwa\_clrqdf>-clrsit.*  
*\*\*\**  
                  CASE lwa\_vbrk-vbtyp.  
*\*                  WHEN 'EBDR'.*  
*\*                    <lwa\_clrqdf>-sbdct = 'BDR'.*  
                    WHEN 'PBD'.  
                      <lwa\_clrqdf>-sbdct = 'PBD'.  
*\*\*\*New status changes*  
*\*\*                      <lwa\_clrqbsi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-pbd\_created.*  
*\*\*                      <lwa\_clrqbsi>-updkz = 'U'.*  
                      <lwa\_clrqbi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-pbd\_created.  
                      <lwa\_clrqbi>-updkz = 'U'.  
*\*\*                    <lwa\_clrqit>-bstat = '2'.*  
*\*                      <lwa\_clrqit>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-pbd\_created.*  
*\*\*\*\**  
*\*                      <lwa\_clrqit>-updkz = 'U'.*  
                    WHEN OTHERS.  
                      <lwa\_clrqdf>-sbdct = 'BD'.  
*\*\*\*\*\*New status changes*  
*\*                      <lwa\_clrqbi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-invoice\_created. "commented by Vinit*  
  
  
                      IF sy-tcode EQ 'VF11'.  
                        LOOP AT lwa\_clrqdoc-x-crbinr ASSIGNING <lwa\_clrqbi>.  
  
                          <lwa\_clrqbi>-bstat = COND #( WHEN lv\_pbdstat = 'A'  
                                                       THEN /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-pbd\_created  
                                                       WHEN lv\_pbdstat = 'B' AND lv\_apr\_stat = ' '  
                                                       THEN /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-pbd\_finalized  
                                                       WHEN lv\_pbdstat = 'B' AND lv\_apr\_stat = 'A'  
                                                       THEN /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-pbd\_finalized  
                                                       WHEN lv\_pbdstat = 'B' AND lv\_apr\_stat = 'B'  
                                                       THEN /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-pbd\_approved  
                                                       WHEN lv\_pbdstat = 'B' AND lv\_apr\_stat = 'C'  
                                                       THEN ' '  
                                                       WHEN lv\_pbdstat = 'B' AND lv\_apr\_stat = 'D'  
                                                       THEN /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-pbd\_in\_rework  
                                                       WHEN lv\_pbdstat = 'C'  
                                                       THEN /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-pbd\_approved  ) .  
                          <lwa\_clrqbi>-updkz = 'U'.  
                        ENDLOOP.  
  
                      ELSE.  
  
  
                        LOOP AT lwa\_clrqdoc-x-crbinr ASSIGNING <lwa\_clrqbi>.  
  
                          <lwa\_clrqbi>-bstat = COND #( WHEN lwa\_vbrk-rfbsk CA ' AB'  
                                                       THEN /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-invoice\_created  
                                                       WHEN lwa\_vbrk-rfbsk CA 'CDH'  
                                                       THEN /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-invoice\_posted  
                                                       WHEN lwa\_vbrk-rfbsk CA 'FGIK'  
                                                       THEN /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-erroneous\_invoice  ).  
                          <lwa\_clrqbi>-updkz = 'U'.  
                        ENDLOOP.  
                      ENDIF.  
*\*\*                    <lwa\_clrqit>-bstat = 'B'.*  
*\*                      <lwa\_clrqit>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-invoice\_created.*  
*\*\*\*\**  
*\*                      <lwa\_clrqit>-updkz = 'U'.*  
                  ENDCASE.  
  
*\*                IF sy-subrc = 0.*  
*\*                  <lwa\_clrqit>-updkz = 'U'.*  
*\*                  <lwa\_clrqit>-bstat = '2'.*  
*\*                ENDIF.*  
                ENDIF.  
              ENDLOOP.  
*\*\*\*New status changes*  
            ELSE.  
  
              READ TABLE it\_vbrk INTO lwa\_vbrk WITH KEY vbeln = lwa\_vbrp-vbeln.  
  
              IF ( ( sy-subrc EQ 0 ) AND ( lwa\_vbrk-vbtyp EQ 'M' OR lwa\_vbrk-vbtyp EQ 'O' OR lwa\_vbrk-vbtyp EQ 'P'  
                OR lwa\_vbrk-vbtyp EQ 'U' OR lwa\_vbrk-vbtyp EQ '5' OR lwa\_vbrk-vbtyp EQ '6' OR lwa\_vbrk-vbtyp EQ 'PBD' )  
                AND ( lv\_prev\_vbeln NE lwa\_vbrp-vbeln ) ).  
  
                lv\_prev\_vbeln = lwa\_vbrp-vbeln.  
  
                LOOP AT lt\_clrqdlf INTO lwa\_clrqdlf WHERE prcdc = lwa\_vbrp-vgbel.  
  
*\*                  READ TABLE lwa\_clrqdoc-x-critm*  
*\*                  ASSIGNING <lwa\_clrqit>*  
*\*                  WITH KEY clrno = lwa\_clrqdlf-clrno*  
*\*                           clrit = lwa\_clrqdlf-clrit*  
*\*                           clrsit = lwa\_clrqdlf-clrsit.*  
*\**  
*\*                  IF sy-subrc EQ 0*  
*\*                    AND <lwa\_clrqit> IS ASSIGNED.*  
*\**  
*\*                    IF lwa\_vbrk-vf\_status EQ 'A'.*  
*\*                      <lwa\_clrqit>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-invoice\_posted.*  
*\*                    ELSEIF lwa\_vbrk-vf\_status EQ 'B' OR lwa\_vbrk-vf\_status EQ 'D'.*  
*\*                      <lwa\_clrqit>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-erroneous\_invoice.*  
*\*                    ENDIF.*  
*\**  
*\*                    <lwa\_clrqit>-updkz = 'U'.*  
*\**  
*\*                  ENDIF.*  
  
*\*                  UNASSIGN <lwa\_clrqbsi>. " clrqbsi deleted*  
*\*                  READ TABLE ls\_clrqbs\_doc-x-clrqbsi*  
*\*                  ASSIGNING <lwa\_clrqbsi>*  
*\*                  WITH KEY clrno = lwa\_clrqdlf-clrno*  
*\*                           xrfdn = lwa\_clrqdlf-xrfdn*  
*\*                           xrfin = lwa\_clrqdlf-xrfin.*  
*\**  
*\*                  IF sy-subrc EQ 0*  
*\*                    AND <lwa\_clrqbsi> IS ASSIGNED.*  
*\**  
*\*                    IF lwa\_vbrk-vf\_status EQ 'A'.*  
*\*                      <lwa\_clrqbsi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-invoice\_posted.*  
*\*                    ELSEIF lwa\_vbrk-vf\_status EQ 'B' OR lwa\_vbrk-vf\_status EQ 'D'.*  
*\*                      <lwa\_clrqbsi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-erroneous\_invoice.*  
*\*                    ENDIF.*  
*\**  
*\*                    <lwa\_clrqbsi>-updkz = 'U'.*  
*\**  
*\*                  ENDIF.*  
  
                  IF <lwa\_clrqbi> IS ASSIGNED.  
                    UNASSIGN <lwa\_clrqbi>.  
                  ENDIF.  
                  READ TABLE lwa\_clrqdoc-x-crbinr  
                  ASSIGNING <lwa\_clrqbi>  
                  WITH KEY clrno     = lwa\_clrqdlf-clrno  
                           bill\_item = lwa\_clrqdlf-rtbbinr.  
                  IF sy-subrc EQ 0 AND <lwa\_clrqbi> IS ASSIGNED.  
                    IF lwa\_vbrk-vf\_status EQ 'A'.  
                      <lwa\_clrqbi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-invoice\_posted.  
                    ELSEIF lwa\_vbrk-vf\_status EQ 'B' OR lwa\_vbrk-vf\_status EQ 'D'.  
                      <lwa\_clrqbi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-erroneous\_invoice.  
                    ELSEIF lwa\_vbrk-pbd\_status = 'B' AND lwa\_vbrk-apm\_approval\_status = ' '.  
                      <lwa\_clrqbi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-pbd\_finalized.  
                    ELSEIF lwa\_vbrk-pbd\_status = 'B' AND lwa\_vbrk-apm\_approval\_status = 'A'.  
                      <lwa\_clrqbi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-pbd\_finalized.  
                    ELSEIF lwa\_vbrk-pbd\_status = 'B' AND lwa\_vbrk-apm\_approval\_status = 'B'.  
                      <lwa\_clrqbi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-pbd\_approved.  
                    ELSEIF lwa\_vbrk-pbd\_status = 'B' AND lwa\_vbrk-apm\_approval\_status = 'C'.  
                      CLEAR <lwa\_clrqbi>-bstat.  
                    ELSEIF lwa\_vbrk-pbd\_status = 'B' AND lwa\_vbrk-apm\_approval\_status = 'D'.  
                      <lwa\_clrqbi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-pbd\_in\_rework.  
*\*                      ELSEIF lwa\_vbrk-pbd\_status = 'B' AND lwa\_vbrk-APM\_APPROVAL\_STATUS = 'C'.*  
*\*                      <lwa\_clrqbi>-bstat = /cgdc/cl\_clrq\_process=>gc\_itm\_bill\_status-pbd\_reject.*  
                    ENDIF.  
                    <lwa\_clrqbi>-updkz = 'U'.  
                  ENDIF.  
  
  
                ENDLOOP.  
  
              ENDIF.  
  
            ENDIF.  
*\*\*\**  
          ENDLOOP.  
  
      ENDCASE.  
  
*\*      APPEND LINES OF lt\_critm TO lwa\_clrqdoc-x-critm.*  
  
      IF NOT lwa\_clrqdoc IS INITIAL.  
  
        APPEND lwa\_clrqdoc TO lt\_clrqdoc.  
  
        CALL FUNCTION '/CGDC/CLRQ\_SAVE'  
          EXPORTING  
            i\_commit\_work = ' '  
          CHANGING  
            ct\_clrq\_doc   = lt\_clrqdoc  
          EXCEPTIONS  
            no\_change     = 1  
            OTHERS        = 2.  
        IF sy-subrc <> 0.  
        ENDIF.  
  
      ENDIF.  
  
*\*      IF NOT ls\_clrqbs\_doc IS INITIAL. "clrqbs deleted*  
*\**  
*\*        CALL FUNCTION '/CGDC/CLRQBS\_SAVE'*  
*\*          EXPORTING*  
*\*            i\_commit\_work = space*  
*\*          CHANGING*  
*\*            cs\_clrqbs\_doc = ls\_clrqbs\_doc.*  
*\*        IF sy-subrc <> 0.*  
*\*        ENDIF.*  
*\**  
*\*      ENDIF.*  
  
    ENDIF.  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>BDR\_REJECT*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IT\_PREBILLING                  TYPE        IF\_SD\_BIL\_TYPE\_DEF=>TT\_DRAFT\_DOCUMENT*  
*\* | [<---] CV\_SUBRC                       TYPE        SY-SUBRC*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD bdr\_reject.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: CIS-AD: BDR reject*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
    DATA : it\_docnumber          TYPE bapiebdrrequestextbilldocreq\_t,  
           lwa\_docnumber         TYPE bapiebdrrequestextbilldocreq,  
           lt\_rejdocnumber       TYPE bapiebdrrequestextbilldocreq\_t,  
           lt\_rejfaileddocnumber TYPE bapiebdrrequestextbilldocreq\_t,  
           lt\_msg                TYPE bapiebdrrequestmsg\_t,  
           lt\_return             TYPE bapiret2\_t.  
  
    IF  it\_prebilling IS NOT INITIAL.  
      LOOP AT it\_prebilling INTO DATA(lwa\_pbd).  
        SELECT SINGLE \* FROM /cgdc/\_clrqdf  
                        INTO @DATA(lwa\_bdr)  
                         WHERE sbqdc = @lwa\_pbd-vbeln  
                         AND  sbdct  = 'PBD'.  
        IF lwa\_bdr-prcdc IS NOT INITIAL.  
          lwa\_docnumber-extbillingdocrequest  = lwa\_bdr-prcdc.  
          APPEND lwa\_docnumber TO it\_docnumber.  
          CALL FUNCTION 'BAPI\_EBDR\_REJECTMULTIPLE'  
            EXPORTING  
              it\_docnumber             = it\_docnumber  
              iv\_commit\_mode           = '1'  
*\*             TESTRUN                  = ' '*  
            IMPORTING  
              et\_rejecteddocnumber     = lt\_rejdocnumber  
              et\_rejectfaileddocnumber = lt\_rejfaileddocnumber  
              et\_message               = lt\_msg  
              return                   = lt\_return.  
          IF lt\_rejfaileddocnumber IS NOT INITIAL.  
            cv\_subrc = '4'.  
          ELSE.  
            cv\_subrc = '0'.  
          ENDIF.  
        ENDIF.  
        CLEAR : it\_docnumber , lwa\_bdr , lwa\_docnumber.  
      ENDLOOP.  
    ENDIF.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>BDR\_REJECT\_CHECK*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IM\_KOMKBV3                     TYPE        KOMKBV3*  
*\* | [<---] EX\_SUBRC                       TYPE        SYST\_SUBRC*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD bdr\_reject\_check.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: CIS-AD: BDR reject check*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
    IF im\_komkbv3-vbtyp = 'PBD'.  
      SELECT SINGLE \* FROM /cgdc/\_clrqdf INTO @DATA(lwa\_clrqdf) WHERE sbqdc = @im\_komkbv3-vbeln.  
      IF sy-subrc EQ 0.  
        SELECT SINGLE \* FROM vbrk INTO @DATA(lwa\_vbrk)  
                        WHERE vbeln = @lwa\_clrqdf-sbqdc  
                        AND   pbd\_status EQ 'D'.  
        IF sy-subrc EQ 0.  
          SELECT SINGLE \* FROM vbrk INTO @DATA(lwa\_vbrk1)  
                  WHERE vbeln = @lwa\_clrqdf-prcdc  
                  AND   bdr\_status = 'D'.  
          IF sy-subrc EQ 0.  
            ex\_subrc   = 4.  
          ELSE.  
            ex\_subrc   = 0.  
          ENDIF.  
        ELSE.  
          ex\_subrc   = 4.  
        ENDIF.  
      ELSE.  
        ex\_subrc   = 4.  
      ENDIF.  
    ELSE.  
      ex\_subrc = 4.  
    ENDIF.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>DETERMINE\_CALCREQ\_AND\_CONTRACT*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IS\_VBRK                        TYPE        VBRK*  
*\* | [<---] ES\_CLRQHD                      TYPE        /CGDC/S\_CLRQHD*  
*\* | [<---] ES\_VBAK                        TYPE        VBAK*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD determine\_calcreq\_and\_contract.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: CIS-AD: Determine cal req and contract*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
    DATA: ls\_clrqdf TYPE /cgdc/\_clrqdf.  
  
    select single \* FROM /cgdc/\_clrqdf  
      into ls\_clrqdf  
      WHERE sbqdc = is\_vbrk-vbeln.  
  
    IF ls\_clrqdf IS NOT INITIAL.  
      select SINGLE \* FROM /cgdc/\_clrqhd  
        INTO  es\_clrqhd  
        WHERE clrno = ls\_clrqdf-clrno.  
  
      SELECT SINGLE \* FROM vbak  
        INTO es\_vbak  
        WHERE vbeln = es\_clrqhd-vgbel.*"es\_clrqhd-cntno.*  
    ENDIF.  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Instance Public Method /CGDC/CL\_CLRQ\_PROCESS->MESSAGE\_HANDLING*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IT\_ERROR                       TYPE        /CGDC/T\_CLRQ\_ERR*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  method MESSAGE\_HANDLING.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Message handling*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*- Update Item table with the error message*  
  
  
  endmethod.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>ADJUST\_BASEDATE*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IV\_ADJST\_TYPE                  TYPE        /CGDC/CLRQ\_CUST\_DT\_ADJUSTMENT*  
*\* | [<-->] C\_DATE                         TYPE        SYDATUM*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD adjust\_basedate.  
    CASE iv\_adjst\_type.  
      WHEN 'BW'.  
        CALL FUNCTION 'GET\_WEEK\_INFO\_BASED\_ON\_DATE'  
          EXPORTING  
            date   = c\_date  
          IMPORTING  
*\*           WEEK   =*  
            monday = c\_date  
*\*           SUNDAY =*  
          .  
        IF sy-subrc <> 0.  
        ENDIF.  
  
      WHEN 'EW'.  
        CALL FUNCTION 'GET\_WEEK\_INFO\_BASED\_ON\_DATE'  
          EXPORTING  
            date   = c\_date  
          IMPORTING  
*\*           WEEK   =*  
*\*           MONDAY = c\_date*  
            sunday = c\_date.  
        IF sy-subrc <> 0.  
        ENDIF.  
      WHEN 'BM'.  
        CALL FUNCTION 'HR\_JP\_MONTH\_BEGIN\_END\_DATE'  
          EXPORTING  
            iv\_date             = c\_date  
          IMPORTING  
            ev\_month\_begin\_date = c\_date  
*\*           EV\_MONTH\_END\_DATE   =*  
          .  
        IF sy-subrc <> 0.  
        ENDIF.  
      WHEN 'EM'.  
        CALL FUNCTION 'HR\_JP\_MONTH\_BEGIN\_END\_DATE'  
          EXPORTING  
            iv\_date           = c\_date  
          IMPORTING  
*\*           EV\_MONTH\_BEGIN\_DATE       = c\_date*  
            ev\_month\_end\_date = c\_date.  
        IF sy-subrc <> 0.  
        ENDIF.  
      WHEN 'BQ'.  
        CALL FUNCTION 'BKK\_GET\_QUARTER\_DATE'  
          EXPORTING  
            i\_date          = c\_date  
*\*           I\_PERIOD        =*  
          IMPORTING  
            e\_quarter\_start = c\_date  
*\*           E\_QUARTER\_END   =*  
*\*           E\_QUARTER\_ADD   =*  
          .  
        IF sy-subrc <> 0.  
        ENDIF.  
  
      WHEN 'EQ'.  
        CALL FUNCTION 'BKK\_GET\_QUARTER\_DATE'  
          EXPORTING  
            i\_date        = c\_date  
*\*           I\_PERIOD      =*  
          IMPORTING  
*\*           E\_QUARTER\_START       = c\_date*  
            e\_quarter\_end = c\_date  
*\*           E\_QUARTER\_ADD =*  
          .  
        IF sy-subrc <> 0.  
        ENDIF.  
      WHEN 'BY'.  
        c\_date+4(4) = 0101.  
  
      WHEN 'EY'.  
        c\_date+4(4) = 1231.  
      WHEN OTHERS.  
    ENDCASE.  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>CHECK\_BILLING\_ELIGIBILITY*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IV\_VBELN                       TYPE        VBELN*  
*\* | [<---] EV\_BLOCK\_BILLING               TYPE        CHAR1*  
*\* | [<---] EV\_BLOCKED\_CONTRACT            TYPE        VBELN*  
*\* | [<---] ET\_BLOCKED\_ITMS                TYPE        /CGDC/TT\_CR\_BLOCKED\_ITEMS*  
*\* | [EXC!] NON\_RTB\_DOC*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD check\_billing\_eligibility.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Check billing eligibility*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
    *"" check if billing doc created via RTB*  
    SELECT SINGLE \* FROM /cgdc/\_clrqdf  
      INTO @DATA(ls\_clrqdf)  
      WHERE sbqdc = @iv\_vbeln.  
    IF sy-subrc = 0.  
      *"" get corresponding sales contract*  
      SELECT SINGLE vgbel FROM /cgdc/\_clrqhd  
        INTO @DATA(lv\_contract)  
        WHERE clrno = @ls\_clrqdf-clrno.  
      IF sy-subrc = 0.  
        *"" check billing block on header*  
        SELECT SINGLE faksk FROM vbak  
          INTO @DATA(lv\_faksk)  
          WHERE vbeln = @lv\_contract.  
        IF lv\_faksk IS NOT INITIAL.  
          ev\_blocked\_contract = lv\_contract.  
  
          APPEND INITIAL LINE TO et\_blocked\_itms ASSIGNING FIELD-SYMBOL(<lfs\_blocked\_items>).  
          <lfs\_blocked\_items>-vbeln = lv\_contract.  
          <lfs\_blocked\_items>-posnr = '00000'.  
          <lfs\_blocked\_items>-billing\_block = lv\_faksk.  
        ENDIF.  
  
        *"" check billing block on items w.r.t PBD Billing items*  
        SELECT \*  
          FROM vbap  
          INTO TABLE @DATA(lt\_vbap)  
          WHERE vbeln = @lv\_contract.  
        IF sy-subrc = 0.  
  
        ENDIF.  
  
        SELECT \* FROM /cgdc/\_clrqdf  
          INTO TABLE @DATA(lt\_clrqdf)  
          WHERE sbqdc = @iv\_vbeln.  
        IF sy-subrc = 0.  
  
          SORT lt\_clrqdf BY rtbbinr.  
          DELETE ADJACENT DUPLICATES FROM lt\_clrqdf COMPARING rtbbinr.  
  
          SELECT \* FROM /cgdc/\_clrqbi  
            FOR ALL ENTRIES IN @lt\_clrqdf  
            WHERE bill\_item = @lt\_clrqdf-rtbbinr  
            INTO TABLE @DATA(lt\_clrqbi).  
          IF sy-subrc = 0.  
            LOOP AT lt\_clrqbi INTO DATA(ls\_clrqbi).  
  
              DATA(lv\_faksp) = VALUE #( lt\_vbap[ ps\_psp\_pnr = ls\_clrqbi-ps\_psp\_pnr ]-faksp OPTIONAL ).  
  
              IF lv\_faksp IS NOT INITIAL.  
                APPEND INITIAL LINE TO et\_blocked\_itms ASSIGNING <lfs\_blocked\_items>.  
                <lfs\_blocked\_items>-vbeln = lv\_contract.  
                <lfs\_blocked\_items>-posnr = VALUE #( lt\_vbap[ ps\_psp\_pnr = ls\_clrqbi-ps\_psp\_pnr ]-posnr OPTIONAL ).  
                <lfs\_blocked\_items>-billing\_block = lv\_faksp.  
              ENDIF.  
            ENDLOOP.  
          ENDIF.  
        ENDIF.  
      ENDIF.  
  
    ELSE.  
      RAISE non\_rtb\_doc.  
    ENDIF.  
  
    IF et\_blocked\_itms IS NOT INITIAL.  
      ev\_block\_billing = abap\_true.  
    ENDIF.  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>COLLECT\_DATA\_FOR\_MIGO*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IT\_ACDOCA                      TYPE        FINST\_ACDOCA(optional)*  
*\* | [--->] IT\_MSEG                        TYPE        TY\_T\_MSEG(optional)*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  method COLLECT\_DATA\_FOR\_MIGO.  
  endmethod.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>COPY\_PBD\_CONDITIONS*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IS\_KOMP                        TYPE        KOMP*  
*\* | [--->] IV\_KOMV\_KSCHL                  TYPE        KSCHL*  
*\* | [--->] IS\_KOMK                        TYPE        KOMK*  
*\* | [<-->] CV\_KWERT                       TYPE        KWERT*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD copy\_pbd\_conditions.  
    DATA: lt\_bdr\_cond TYPE bapiebdrrequestcond\_t.  
    DATA lwa\_bdr\_cond TYPE bapiebdrrequestcond.  
    CHECK is\_komk-trtyp = 'H'.  
    CALL FUNCTION '/CGDC/CLRQ\_DATA\_GET'  
      IMPORTING  
*\*       ES\_CLRHD     =*  
*\*       ES\_CLRIT     =*  
*\*       ES\_CRGRR     =*  
*\*       ET\_CRGRRF    =*  
*\*       ET\_CLGRP\_ITM\_BUFFER       =*  
*\*       ET\_CLGRP\_CND\_BUFFER       =*  
        et\_pbd\_conds = lt\_bdr\_cond.  
  
*\*    LOOP AT ct\_komv ASSIGNING FIELD-SYMBOL(<lwa\_komv>).*  
    READ TABLE lt\_bdr\_cond INTO lwa\_bdr\_cond WITH KEY conditiontype = iv\_komv\_kschl  
                                                      precedingdocument = is\_komp-vgbel  
                                                      precedingdocumentitem = is\_komp-vgpos.  
    IF sy-subrc = 0.  
      cv\_kwert = lwa\_bdr\_cond-conditionratevalue.  
    ELSE.  
      IF IS\_KOMk-VBTYP EQ 'EBDR'..  
        SELECT a~knumv,a~kposn,a~kwert,a~kschl,b~vbeln  
          FROM prcd\_elements AS a  
          INNER JOIN vbrk AS b ON a~knumv = b~knumv  
          INTO TABLE @DATA(lt\_bdr\_cond\_sel)  
          WHERE a~knumv = @is\_komk-knumv.  
  
        READ TABLE lt\_bdr\_cond\_sel INTO DATA(lwa\_pbd\_cond\_sel) WITH KEY knumv = is\_komk-knumv  
                                                                kposn = is\_komp-kposn  
                                                                kschl = iv\_komv\_kschl.  
        IF sy-subrc = 0.  
          cv\_kwert = lwa\_pbd\_cond\_sel-kwert.  
        ENDIF.  
      ELSE.  
        SELECT a~knumv,a~kposn,a~kwert,a~kschl,b~vbeln  
          FROM prcd\_elements AS a  
          INNER JOIN vbrk AS b ON a~knumv = b~knumv  
          INTO TABLE @DATA(lt\_pbd\_cond)  
          WHERE b~vbeln = @is\_komp-vgbel.  
  
        READ TABLE lt\_pbd\_cond INTO DATA(lwa\_pbd\_cond) WITH KEY vbeln = is\_komp-vgbel  
                                                                kposn = is\_komp-vgpos  
                                                                kschl = iv\_komv\_kschl.  
        IF sy-subrc = 0.  
          cv\_kwert = lwa\_pbd\_cond-kwert.  
        ENDIF.  
      ENDIF.  
  
    ENDIF.  
  
  
*\*    ENDLOOP.*  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>GENERATE\_HEADER\_DF*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IS\_BILL\_DATA                   TYPE        VBCO6*  
*\* | [--->] IS\_BILL\_HEADER                 TYPE        VBRK*  
*\* | [<-->] CT\_VBFAS\_TAB                   TYPE        VBFAS\_T*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD generate\_header\_df.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Generate header df*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
    DATA: lwa\_vbfa    TYPE vbfas,  
          lt\_vbfa\_tab TYPE TABLE OF vbfas,  
          lwa\_vbrf\_df LIKE lwa\_vbfa,  
          lwa\_vbrk\_p  TYPE vbrk,  
          lwa\_vbrk\_s  TYPE vbrk,  
          lv\_stufe    TYPE stufe\_vbfa VALUE '1'.  
  
    IF is\_bill\_data-posnr IS INITIAL.  
      SELECT \* FROM vbak INTO TABLE @DATA(lt\_vbak)  
                         WHERE vbeln = @is\_bill\_data-vbeln  
                         AND  /cgdc/rtbpn <> 0.  
      IF sy-subrc NE 0.  
        SELECT SINGLE a~vgbel, b~rtbbinr  
          FROM /cgdc/\_clrqhd AS a  
          INNER JOIN /cgdc/\_clrqdf AS b  
          ON  b~clrno = a~clrno  
          WHERE b~sbqdc = @is\_bill\_data-vbeln  
            INTO ( @DATA(lv\_vbeln), @DATA(lv\_rtbbinr) ).  
        IF sy-subrc = 0 .  
          DATA(lv\_vbfa) = abap\_true.  
          CLEAR ct\_vbfas\_tab[].  
          SELECT \* FROM vbak INTO TABLE @lt\_vbak  
                   WHERE vbeln = @lv\_vbeln  
                   AND  /cgdc/rtbpn <> 0.  
        ENDIF.  
  
      ENDIF.  
  
      IF sy-subrc EQ 0.  
        *"Contract*  
        lv\_vbeln = COND #( WHEN lv\_vbfa = abap\_false THEN is\_bill\_data-vbeln  
                                                   ELSE lv\_vbeln ).  
  
        *"parent node*  
        ct\_vbfas\_tab = CORRESPONDING #( lt\_vbak EXCEPT vbelv posnv ).  
        IF line\_exists( ct\_vbfas\_tab[ 1 ] ).  
          ct\_vbfas\_tab[ 1 ]-vbtyp\_n = VALUE #( lt\_vbak[ vbeln = lv\_vbeln ]-vbtyp OPTIONAL ) .  
        ENDIF.  
  
        SELECT \* FROM /cgdc/\_clrqhd INTO TABLE @DATA(lt\_clrqhd)  
                           FOR ALL ENTRIES IN @lt\_vbak  
*\*                             WHERE cntno = @lt\_vbak-vbeln.*  
                           WHERE vgbel = @lt\_vbak-vbeln.  
        IF sy-subrc EQ 0.  
          LOOP AT lt\_clrqhd INTO DATA(lwa\_clrqhd).  
            MOVE-CORRESPONDING lwa\_clrqhd TO lwa\_vbfa.  
            lwa\_vbfa-vbeln   = lwa\_clrqhd-clrno.  
            lwa\_vbfa-mandt   = sy-mandt.  
            lwa\_vbfa-stufe   = lv\_stufe.  
            lwa\_vbfa-vbtyp\_n = lwa\_clrqhd-clrqt.  
            IF lv\_vbfa IS INITIAL.  
              READ TABLE ct\_vbfas\_tab INTO DATA(lwa\_vbfa\_tab)  INDEX 1.  
              IF sy-subrc = 0.  
                lwa\_vbfa-vbelv   = lwa\_vbfa\_tab-vbeln.  
                lwa\_vbfa-vbtyp\_v = lwa\_vbfa\_tab-vbtyp\_n.  
                lwa\_vbfa-erdat   = lwa\_clrqhd-erdat.  
                lwa\_vbfa-erzet   = lwa\_clrqhd-erzet.  
              ENDIF.  
            ELSE.  
              DATA(ls\_vbak) = VALUE #( lt\_vbak[ vbeln = lv\_vbeln ] OPTIONAL ).  
              IF ls\_vbak IS NOT INITIAL.  
                lwa\_vbfa = VALUE #( BASE lwa\_vbfa  
                                   vbelv   = ls\_vbak-vbeln  
                                   vbtyp\_v = ls\_vbak-vbtyp  
                                   erdat   = ls\_vbak-erdat  
                                   erzet   = ls\_vbak-erzet  ).  
              ENDIF.  
            ENDIF.  
            APPEND lwa\_vbfa TO ct\_vbfas\_tab.  
            CLEAR : lwa\_vbfa.  
          ENDLOOP.  
  
*\*          IF sy-uname <> 'AMISHARMA'.*  
*\*            SELECT \* FROM /cgdc/\_clrqdf INTO TABLE @DATA(lt\_clrqdf)*  
*\*                           FOR ALL ENTRIES IN @lt\_clrqhd*  
*\*                           WHERE clrno = @lt\_clrqhd-clrno.*  
*\*          ELSE.*  
          IF lv\_rtbbinr IS NOT INITIAL.  
            SELECT DISTINCT df~\* FROM vbap AS vbap  
            INNER JOIN /cgdc/\_clrqbi AS bi ON  
             bi~vgbel = vbap~vbeln AND  
             bi~vgpos = vbap~posnr  
            INNER JOIN /cgdc/\_clrqdf AS df ON  
             df~clrno   = bi~clrno AND  
             df~rtbbinr = @lv\_rtbbinr  
            WHERE vbap~vbeln = @lv\_vbeln  *"is\_bill\_data-vbeln*  
            INTO TABLE @DATA(lt\_clrqdf).  
          ELSE.  
            SELECT DISTINCT df~\* FROM vbap AS vbap  
            INNER JOIN /cgdc/\_clrqbi AS bi ON  
             bi~vgbel = vbap~vbeln AND  
             bi~vgpos = vbap~posnr  
            INNER JOIN /cgdc/\_clrqdf AS df ON  
             df~clrno   = bi~clrno  
            *"AND*  
             *"df~rtbbinr = bi~bill\_item*  
            WHERE vbap~vbeln = @lv\_vbeln  *"is\_bill\_data-vbeln*  
            INTO TABLE @lt\_clrqdf.  
          ENDIF.  
  
*\*          ENDIF.*  
  
          IF lt\_clrqdf IS NOT INITIAL.  
            SORT lt\_clrqdf BY clrno rtbbinr prcdc prcit sbqdc sbqit.     *"create right seq. BDR->PBD->BD*  
*\*            DELETE ADJACENT DUPLICATES FROM lt\_clrqdf.*  
            SELECT \* FROM vbrk INTO TABLE @DATA(lt\_vbrk\_df)  
              FOR ALL ENTRIES IN @lt\_clrqdf  
              WHERE vbeln = @lt\_clrqdf-sbqdc.  
          ENDIF.  
          IF sy-subrc EQ 0.  
            lv\_stufe  =   lv\_stufe + 1.  
            LOOP AT lt\_clrqdf INTO DATA(lwa\_clrqdf).  
*\*          READ TABLE vbfa\_tab INTO lwa\_vbfa INDEX 1.*  
              IF lwa\_clrqdf-prcdc IS NOT INITIAL .  
                READ TABLE lt\_vbrk\_df INTO lwa\_vbrk\_p  
                WITH KEY vbeln = lwa\_clrqdf-prcdc.  
              ENDIF.  
              IF lwa\_clrqdf-sbqdc IS NOT INITIAL .  
                READ TABLE lt\_vbrk\_df INTO lwa\_vbrk\_s  
                WITH KEY vbeln = lwa\_clrqdf-sbqdc.  
              ENDIF.  
              IF lwa\_clrqdf-prcdc IS INITIAL.  
                READ TABLE ct\_vbfas\_tab  INTO lwa\_vbrf\_df  
                WITH KEY vbeln = lwa\_clrqdf-clrno.  
  
                lwa\_vbfa-vbelv = lwa\_clrqdf-clrno.  
                lwa\_vbfa-vbeln = lwa\_clrqdf-sbqdc.  
                lwa\_vbfa-vbtyp\_v = lwa\_clrqhd-clrqt.  
                lwa\_vbfa-vbtyp\_n = lwa\_vbrk\_s-vbtyp.  
                lwa\_vbfa-erdat = lwa\_clrqdf-erdat.  
                lwa\_vbfa-erzet = lwa\_clrqdf-erzet.  
                lwa\_vbfa-rfwrt = lwa\_clrqdf-netwr.  
                lwa\_vbfa-waers = lwa\_clrqdf-waerk.  
                lwa\_vbfa-stufe = lwa\_vbrf\_df-stufe + 1.  
              ELSE.  
                READ TABLE ct\_vbfas\_tab  INTO lwa\_vbrf\_df  
                WITH KEY vbeln = lwa\_clrqdf-prcdc.  
  
                lwa\_vbfa-vbelv = lwa\_clrqdf-prcdc.  
                lwa\_vbfa-vbeln = lwa\_clrqdf-sbqdc.  
                lwa\_vbfa-vbtyp\_v = lwa\_vbrk\_p-vbtyp.  
                lwa\_vbfa-vbtyp\_n = lwa\_vbrk\_s-vbtyp.  
                lwa\_vbfa-erdat = lwa\_clrqdf-erdat.  
                lwa\_vbfa-erzet = lwa\_clrqdf-erzet.  
                lwa\_vbfa-rfwrt = lwa\_clrqdf-netwr.  
                lwa\_vbfa-waers = lwa\_clrqdf-waerk.  
                lwa\_vbfa-stufe = lwa\_vbrf\_df-stufe + 1.  
  
              ENDIF.  
*\*          lwa\_vbfa-stufe = lv\_stufe.*  
  
              lwa\_vbfa-mandt  = sy-mandt.  
              APPEND lwa\_vbfa TO ct\_vbfas\_tab.  
              CLEAR : lwa\_vbfa.  
            ENDLOOP.  
            CHECK ct\_vbfas\_tab IS NOT INITIAL.  
            IF ct\_vbfas\_tab IS NOT INITIAL.  
              SELECT \* FROM vbrk INTO TABLE @DATA(lt\_vbrk)  
                                 FOR ALL ENTRIES IN @ct\_vbfas\_tab  
                                 WHERE vbeln = @ct\_vbfas\_tab-vbeln.  
              IF sy-subrc EQ 0.  
                LOOP AT ct\_vbfas\_tab ASSIGNING FIELD-SYMBOL(<lfs\_vbfa>).  
                  READ TABLE lt\_vbrk INTO DATA(lwa\_vbrk) WITH KEY vbeln = <lfs\_vbfa>-vbeln.  
                  IF sy-subrc EQ 0 AND <lfs\_vbfa>-vbtyp\_n IS INITIAL.  
                    <lfs\_vbfa>-vbtyp\_n  = lwa\_vbrk-vbtyp.  
                  ENDIF.  
                ENDLOOP.  
              ENDIF.  
            ENDIF.  
          ENDIF.  
        ENDIF.  
      ELSEIF ( is\_bill\_header IS NOT INITIAL AND is\_bill\_data-posnr IS INITIAL ).  
  
        lv\_stufe = 1.  
  
        SELECT SINGLE \* FROM /cgdc/\_clrqdf INTO lwa\_clrqdf  
                           WHERE prcdc = is\_bill\_data-vbeln.  
        IF sy-subrc EQ 0.  
          SELECT SINGLE \* FROM /cgdc/\_clrqhd INTO  lwa\_clrqhd  
                           WHERE clrno = lwa\_clrqdf-clrno.  
          IF sy-subrc EQ 0.  
            SELECT SINGLE \* FROM vbak INTO  @DATA(lwa\_vbak)  
*\*                            WHERE vbeln = @lwa\_clrqhd-cntno.*  
                          WHERE vbeln = @lwa\_clrqhd-vgbel.  
  
          ENDIF.  
  
          SELECT SINGLE \* FROM /cgdc/\_clrqdf INTO lwa\_clrqdf  
                         WHERE prcdc = is\_bill\_data-vbeln.  
  
          lwa\_vbfa-vbelv = ''.  
          lwa\_vbfa-vbeln = lwa\_vbak-vbeln.  
          lwa\_vbfa-vbtyp\_v = ' '.  
          lwa\_vbfa-vbtyp\_n = 'G'.  
          lwa\_vbfa-erdat = lwa\_clrqdf-erdat.  
          lwa\_vbfa-erzet = lwa\_clrqdf-erzet.  
          lwa\_vbfa-rfwrt = lwa\_clrqdf-netwr.  
          lwa\_vbfa-waers = lwa\_clrqdf-waerk.  
          lwa\_vbfa-mandt  = sy-mandt.  
          APPEND lwa\_vbfa TO lt\_vbfa\_tab.  
          CLEAR : lwa\_vbfa.  
  
          lwa\_vbfa-vbelv   = lwa\_vbak-vbeln.  
          lwa\_vbfa-vbeln   = lwa\_clrqdf-clrno.  
          lwa\_vbfa-vbtyp\_v = 'G'.  
          lwa\_vbfa-vbtyp\_n = lwa\_clrqhd-clrqt.  
          lwa\_vbfa-erdat   = lwa\_clrqdf-erdat.  
          lwa\_vbfa-erzet   = lwa\_clrqdf-erzet.  
          lwa\_vbfa-rfwrt   = lwa\_clrqdf-netwr.  
          lwa\_vbfa-waers   = lwa\_clrqdf-waerk.  
          lwa\_vbfa-stufe   =   lv\_stufe.  
          lwa\_vbfa-mandt   = sy-mandt.  
          APPEND lwa\_vbfa TO lt\_vbfa\_tab.  
          CLEAR : lwa\_vbfa.  
        ENDIF.  
  
*\*        IF sy-uname <> 'AMISHARMA'.*  
*\*          SELECT \* FROM /cgdc/\_clrqdf INTO TABLE lt\_clrqdf*  
*\*                   WHERE clrno = lwa\_clrqhd-clrno.*  
*\*        ELSE.*  
        SELECT DISTINCT df~\* FROM vbap AS vbap  
            INNER JOIN /cgdc/\_clrqbi AS bi ON  
             bi~vgbel = vbap~vbeln AND  
             bi~vgpos = vbap~posnr  
            INNER JOIN /cgdc/\_clrqdf AS df ON  
             df~clrno   = bi~clrno  
          *"AND*  
           *"  df~rtbbinr = bi~bill\_item*  
            WHERE vbap~vbeln = @lv\_vbeln  *"is\_bill\_data-vbeln*  
            INTO TABLE @lt\_clrqdf.  
*\*        ENDIF.*  
  
        IF lt\_clrqdf IS NOT INITIAL.  
          SORT lt\_clrqdf BY clrno rtbbinr.  
          DELETE ADJACENT DUPLICATES FROM lt\_clrqdf.  
          SELECT \* FROM vbrk INTO TABLE lt\_vbrk\_df  
            FOR ALL ENTRIES IN lt\_clrqdf  
            WHERE vbeln = lt\_clrqdf-sbqdc.  
        ENDIF.  
        IF sy-subrc EQ 0.  
          lv\_stufe  =   lv\_stufe + 1.  
          READ TABLE lt\_clrqdf INTO lwa\_clrqdf WITH KEY sbqdc = is\_bill\_data-vbeln.  
          IF sy-subrc EQ 0.  
            DATA(lv\_index) = sy-tabix.  
          ENDIF.  
          LOOP AT lt\_clrqdf INTO lwa\_clrqdf FROM lv\_index.  
  
            IF lwa\_clrqdf-prcdc IS NOT INITIAL .  
              READ TABLE lt\_vbrk\_df INTO lwa\_vbrk\_p  
              WITH KEY vbeln = lwa\_clrqdf-prcdc.  
            ENDIF.  
            IF lwa\_clrqdf-sbqdc IS NOT INITIAL .  
              READ TABLE lt\_vbrk\_df INTO lwa\_vbrk\_s  
              WITH KEY vbeln = lwa\_clrqdf-sbqdc.  
            ENDIF.  
  
            IF lwa\_clrqdf-prcdc IS INITIAL AND lwa\_clrqdf-sbqdc = is\_bill\_data-vbeln.  
              READ TABLE ct\_vbfas\_tab  INTO lwa\_vbrf\_df  
              WITH KEY vbeln = lwa\_clrqdf-clrno.  
  
              lwa\_vbfa-vbelv = lwa\_clrqdf-clrno.  
              lwa\_vbfa-vbeln = lwa\_clrqdf-sbqdc.  
              lwa\_vbfa-vbtyp\_v = lwa\_clrqhd-clrqt.  
              lwa\_vbfa-vbtyp\_n = lwa\_vbrk\_s-vbtyp.  
              lwa\_vbfa-erdat = lwa\_clrqdf-erdat.  
              lwa\_vbfa-erzet = lwa\_clrqdf-erzet.  
              lwa\_vbfa-rfwrt = lwa\_clrqdf-netwr.  
              lwa\_vbfa-waers = lwa\_clrqdf-waerk.  
              lwa\_vbfa-stufe = lwa\_vbrf\_df-stufe + 1.  
              DATA(lwa\_clrqdf\_tmp) = lwa\_clrqdf.  
            ELSEIF ( lwa\_clrqdf\_tmp-sbqdc EQ lwa\_clrqdf-prcdc ).  
              READ TABLE ct\_vbfas\_tab  INTO lwa\_vbrf\_df  
              WITH KEY vbeln = lwa\_clrqdf-prcdc.  
  
              lwa\_vbfa-vbelv = lwa\_clrqdf-prcdc.  
              lwa\_vbfa-vbeln = lwa\_clrqdf-sbqdc.  
              lwa\_vbfa-vbtyp\_v = lwa\_vbrk\_p-vbtyp.  
              lwa\_vbfa-vbtyp\_n = lwa\_vbrk\_s-vbtyp.  
              lwa\_vbfa-erdat = lwa\_clrqdf-erdat.  
              lwa\_vbfa-erzet = lwa\_clrqdf-erzet.  
              lwa\_vbfa-rfwrt = lwa\_clrqdf-netwr.  
              lwa\_vbfa-waers = lwa\_clrqdf-waerk.  
              lwa\_vbfa-stufe = lwa\_vbrf\_df-stufe + 1.  
              lwa\_clrqdf\_tmp = lwa\_clrqdf.  
            ELSE.  
              EXIT.  
            ENDIF.  
            lv\_index = lv\_index + 1.  
            lwa\_vbfa-mandt  = sy-mandt.  
            APPEND lwa\_vbfa TO lt\_vbfa\_tab.  
            CLEAR : lwa\_vbfa.  
          ENDLOOP.  
          ct\_vbfas\_tab[] = lt\_vbfa\_tab[].  
        ENDIF.  
      ENDIF.  
    ENDIF.  
    FREE lv\_vbfa.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>GENERATE\_ITEM\_DF*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IS\_BILL\_DATA                   TYPE        VBCO6*  
*\* | [--->] IS\_BILL\_HEADER                 TYPE        VBRK*  
*\* | [<-->] CT\_VBFAS\_TAB                   TYPE        VBFAS\_T*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD generate\_item\_df.  
  
    DATA: lwa\_vbfa    TYPE vbfas,  
          lt\_vbfa\_tab TYPE TABLE OF vbfas,  
          lwa\_vbrf\_df LIKE lwa\_vbfa,  
          lwa\_vbrk\_p  TYPE vbrk,  
          lwa\_vbrk\_s  TYPE vbrk,  
          lv\_stufe    TYPE stufe\_vbfa VALUE '1'.  
  
    IF is\_bill\_data-posnr IS NOT INITIAL.  
  
      SELECT SINGLE vbeln, posnr  
        FROM vbap  
        WHERE vbeln = @is\_bill\_data-vbeln  
          AND posnr = @is\_bill\_data-posnr  
        INTO @DATA(ls\_vbap).  
      IF sy-subrc NE 0.  
  
        SELECT SINGLE bi~vgbel, bi~vgpos, df~rtbbinr  
         FROM /cgdc/\_clrqdf AS df  
        INNER JOIN /cgdc/\_clrqbi AS bi  
        ON  bi~clrno = df~clrno  
        AND bi~bill\_item = df~rtbbinr  
        WHERE df~sbqdc = @is\_bill\_data-vbeln  
         AND df~sbqit = @is\_bill\_data-posnr  
      INTO ( @DATA(lv\_billdoc), @DATA(lv\_posnr), @DATA(lv\_rtbbinr) ).  
        DATA(lv\_vbfa) = abap\_true.  
        CLEAR ct\_vbfas\_tab[].  
      ELSE.  
        lv\_billdoc = ls\_vbap-vbeln.  
        lv\_posnr   = ls\_vbap-posnr.  
      ENDIF.  
  
  
      IF sy-subrc = 0.  
        IF lv\_rtbbinr IS NOT INITIAL.  
          SELECT df~\* FROM vbap AS vbap  
            INNER JOIN /cgdc/\_clrqbi AS bi ON  
*\*             bi~ps\_psp\_pnr = vbap~ps\_psp\_pnr*  
            bi~vgbel = vbap~vbeln AND  
               bi~vgpos = vbap~posnr  
            INNER JOIN /cgdc/\_clrqdf AS df ON  
             df~clrno   = bi~clrno AND  
           df~rtbbinr = bi~bill\_item AND  
             df~rtbbinr = @lv\_rtbbinr  
            WHERE vbap~vbeln = @lv\_billdoc  
              AND vbap~posnr = @lv\_posnr  
            INTO TABLE @DATA(lt\_clrqdf).  
        ELSE.  
                 SELECT df~\* FROM vbap AS vbap  
          INNER JOIN /cgdc/\_clrqbi AS bi ON  
*\*           bi~ps\_psp\_pnr = vbap~ps\_psp\_pnr*  
                   bi~vgbel = vbap~vbeln AND  
               bi~vgpos = vbap~posnr  
          INNER JOIN /cgdc/\_clrqdf AS df ON  
           df~clrno   = bi~clrno AND  
           df~rtbbinr = bi~bill\_item  
*\*           df~rtbbinr = @lv\_rtbbinr*  
          WHERE vbap~vbeln = @lv\_billdoc  
            AND vbap~posnr = @lv\_posnr  
          INTO TABLE @lt\_clrqdf.  
        ENDIF.  
  
  
        SELECT \* FROM vbak INTO TABLE @DATA(lt\_vbak)  
                           WHERE vbeln = @lv\_billdoc  
                           AND  /cgdc/rtbpn <> 0.  
        IF sy-subrc = 0.  
          SELECT \* FROM vbap INTO TABLE @DATA(lt\_vbap)  
                             WHERE vbeln = @lv\_billdoc  
                               AND posnr = @lv\_posnr.  
  
          ct\_vbfas\_tab = CORRESPONDING #( lt\_vbap MAPPING posnn = posnr EXCEPT vbelv posnv ).  
          ct\_vbfas\_tab[ 1 ]-vbtyp\_n = VALUE #( lt\_vbak[ vbeln = lv\_billdoc ]-vbtyp OPTIONAL ) .  
          IF sy-subrc EQ 0.  
            SELECT \* FROM /cgdc/\_clrqhd INTO TABLE @DATA(lt\_clrqhd)  
                               FOR ALL ENTRIES IN @lt\_vbak  
*\*                               WHERE cntno = @lt\_vbak-vbeln.*  
                               WHERE vgbel = @lt\_vbak-vbeln.  
            IF sy-subrc EQ 0.  
              LOOP AT lt\_clrqhd INTO DATA(lwa\_clrqhd).  
                MOVE-CORRESPONDING lwa\_clrqhd TO lwa\_vbfa.  
                lwa\_vbfa-vbeln   = lwa\_clrqhd-clrno.  
                lwa\_vbfa-mandt   = sy-mandt.  
                lwa\_vbfa-stufe   = lv\_stufe.  
                lwa\_vbfa-vbtyp\_n = lwa\_clrqhd-clrqt.  
                IF lv\_vbfa IS INITIAL.  
                  READ TABLE ct\_vbfas\_tab INTO DATA(lwa\_vbfa\_tab)  INDEX 1.  
                  lwa\_vbfa-vbelv   = lwa\_vbfa\_tab-vbeln.  
                  lwa\_vbfa-vbtyp\_v = lwa\_vbfa\_tab-vbtyp\_n.  
                  lwa\_vbfa-erdat   = lwa\_clrqhd-erdat.  
                  lwa\_vbfa-erzet   = lwa\_clrqhd-erzet.  
                ELSE.  
                  ls\_vbap = VALUE #( lt\_vbap[ vbeln = lv\_billdoc  
                                                    posnr = lv\_posnr  ] OPTIONAL ).  
                  DATA(ls\_vbak) = VALUE #( lt\_vbak[ vbeln = lv\_billdoc  ] OPTIONAL ).  
                  IF ls\_vbap IS NOT INITIAL.  
                    lwa\_vbfa = VALUE #( BASE lwa\_vbfa  
                                       vbelv   = ls\_vbap-vbeln  
                                       posnv   = ls\_vbap-posnr  
                                       vbtyp\_v = ls\_vbak-vbtyp ).  
*\*                                       erdat   = ls\_vbak-erdat*  
*\*                                       erzet   = ls\_vbak-erzet  ).*  
                  ENDIF.  
                ENDIF.  
                APPEND lwa\_vbfa TO ct\_vbfas\_tab.  
                CLEAR : lwa\_vbfa.  
  
              ENDLOOP.  
*\*          SELECT \* FROM /cgdc/\_clrqdf INTO TABLE @DATA(lt\_clrqdf)*  
*\*                           FOR ALL ENTRIES IN @lt\_clrqhd*  
*\*                           WHERE clrno = @lt\_clrqhd-clrno.*  
              IF lt\_clrqdf IS NOT INITIAL.  
                SELECT \* FROM vbrk INTO TABLE @DATA(lt\_vbrk\_df)  
                  FOR ALL ENTRIES IN @lt\_clrqdf  
                  WHERE vbeln = @lt\_clrqdf-sbqdc.  
              ENDIF.  
              IF sy-subrc EQ 0.  
                lv\_stufe  =   lv\_stufe + 1.  
                LOOP AT lt\_clrqdf INTO DATA(lwa\_clrqdf).  
*\*          READ TABLE vbfa\_tab INTO lwa\_vbfa INDEX 1.*  
                  IF lwa\_clrqdf-prcdc IS NOT INITIAL .  
                    READ TABLE lt\_vbrk\_df INTO lwa\_vbrk\_p  
                    WITH KEY vbeln = lwa\_clrqdf-prcdc.  
                  ENDIF.  
                  IF lwa\_clrqdf-sbqdc IS NOT INITIAL .  
                    READ TABLE lt\_vbrk\_df INTO lwa\_vbrk\_s  
                    WITH KEY vbeln = lwa\_clrqdf-sbqdc.  
                  ENDIF.  
                  IF lwa\_clrqdf-prcdc IS INITIAL.  
                    READ TABLE ct\_vbfas\_tab  INTO lwa\_vbrf\_df  
                    WITH KEY vbeln = lwa\_clrqdf-clrno.  
  
                    lwa\_vbfa-vbelv = lwa\_clrqdf-clrno.  
                    lwa\_vbfa-vbeln = lwa\_clrqdf-sbqdc.  
                    lwa\_vbfa-vbtyp\_v = lwa\_clrqhd-clrqt.  
                    lwa\_vbfa-vbtyp\_n = lwa\_vbrk\_s-vbtyp.  
                    lwa\_vbfa-erdat = lwa\_clrqdf-erdat.  
                    lwa\_vbfa-erzet = lwa\_clrqdf-erzet.  
                    lwa\_vbfa-rfwrt = lwa\_clrqdf-netwr.  
                    lwa\_vbfa-waers = lwa\_clrqdf-waerk.  
                    lwa\_vbfa-stufe = lwa\_vbrf\_df-stufe + 1.  
                  ELSE.  
                    READ TABLE ct\_vbfas\_tab  INTO lwa\_vbrf\_df  
                    WITH KEY vbeln = lwa\_clrqdf-prcdc.  
  
                    lwa\_vbfa-vbelv = lwa\_clrqdf-prcdc.  
                    lwa\_vbfa-vbeln = lwa\_clrqdf-sbqdc.  
                    lwa\_vbfa-vbtyp\_v = lwa\_vbrk\_p-vbtyp.  
                    lwa\_vbfa-vbtyp\_n = lwa\_vbrk\_s-vbtyp.  
                    lwa\_vbfa-erdat = lwa\_clrqdf-erdat.  
                    lwa\_vbfa-erzet = lwa\_clrqdf-erzet.  
                    lwa\_vbfa-rfwrt = lwa\_clrqdf-netwr.  
                    lwa\_vbfa-waers = lwa\_clrqdf-waerk.  
                    lwa\_vbfa-stufe = lwa\_vbrf\_df-stufe + 1.  
  
                  ENDIF.  
*\*          lwa\_vbfa-stufe = lv\_stufe.*  
  
                  lwa\_vbfa-mandt  = sy-mandt.  
                  APPEND lwa\_vbfa TO ct\_vbfas\_tab.  
                  CLEAR : lwa\_vbfa.  
                ENDLOOP.  
                CHECK ct\_vbfas\_tab IS NOT INITIAL.  
                IF ct\_vbfas\_tab IS NOT INITIAL.  
                  SELECT \* FROM vbrk INTO TABLE @DATA(lt\_vbrk)  
                                     FOR ALL ENTRIES IN @ct\_vbfas\_tab  
                                     WHERE vbeln = @ct\_vbfas\_tab-vbeln.  
                  IF sy-subrc EQ 0.  
                    LOOP AT ct\_vbfas\_tab ASSIGNING FIELD-SYMBOL(<lfs\_vbfa>).  
                      READ TABLE lt\_vbrk INTO DATA(lwa\_vbrk) WITH KEY vbeln = <lfs\_vbfa>-vbeln.  
                      IF sy-subrc EQ 0 AND <lfs\_vbfa>-vbtyp\_n IS INITIAL.  
                        <lfs\_vbfa>-vbtyp\_n  = lwa\_vbrk-vbtyp.  
                      ENDIF.  
                    ENDLOOP.  
                  ENDIF.  
                ENDIF.  
              ENDIF.  
            ENDIF.  
          ELSEIF ( is\_bill\_header IS NOT INITIAL AND is\_bill\_data-posnr IS INITIAL ).  
  
            lv\_stufe = 1.  
  
            SELECT SINGLE \* FROM /cgdc/\_clrqdf INTO lwa\_clrqdf  
                               WHERE prcdc = is\_bill\_data-vbeln.  
            IF sy-subrc EQ 0.  
              SELECT SINGLE \* FROM /cgdc/\_clrqhd INTO  lwa\_clrqhd  
                               WHERE clrno = lwa\_clrqdf-clrno.  
              IF sy-subrc EQ 0.  
                SELECT SINGLE \* FROM vbak INTO  @DATA(lwa\_vbak)  
*\*                              WHERE vbeln = @lwa\_clrqhd-cntno.*  
                              WHERE vbeln = @lwa\_clrqhd-vgbel.  
  
              ENDIF.  
  
              SELECT SINGLE \* FROM /cgdc/\_clrqdf INTO lwa\_clrqdf  
                             WHERE prcdc = is\_bill\_data-vbeln.  
  
              lwa\_vbfa-vbelv = ''.  
              lwa\_vbfa-vbeln = lwa\_vbak-vbeln.  
              lwa\_vbfa-vbtyp\_v = ' '.  
              lwa\_vbfa-vbtyp\_n = 'G'.  
              lwa\_vbfa-erdat = lwa\_clrqdf-erdat.  
              lwa\_vbfa-erzet = lwa\_clrqdf-erzet.  
              lwa\_vbfa-rfwrt = lwa\_clrqdf-netwr.  
              lwa\_vbfa-waers = lwa\_clrqdf-waerk.  
              lwa\_vbfa-mandt  = sy-mandt.  
              APPEND lwa\_vbfa TO lt\_vbfa\_tab.  
              CLEAR : lwa\_vbfa.  
  
              lwa\_vbfa-vbelv   = lwa\_vbak-vbeln.  
              lwa\_vbfa-vbeln   = lwa\_clrqdf-clrno.  
              lwa\_vbfa-vbtyp\_v = 'G'.  
              lwa\_vbfa-vbtyp\_n = lwa\_clrqhd-clrqt.  
              lwa\_vbfa-erdat   = lwa\_clrqdf-erdat.  
              lwa\_vbfa-erzet   = lwa\_clrqdf-erzet.  
              lwa\_vbfa-rfwrt   = lwa\_clrqdf-netwr.  
              lwa\_vbfa-waers   = lwa\_clrqdf-waerk.  
              lwa\_vbfa-stufe   =   lv\_stufe.  
              lwa\_vbfa-mandt   = sy-mandt.  
              APPEND lwa\_vbfa TO lt\_vbfa\_tab.  
              CLEAR : lwa\_vbfa.  
            ENDIF.  
  
*\*        SELECT \* FROM /cgdc/\_clrqdf INTO TABLE lt\_clrqdf*  
*\*                           WHERE clrno = lwa\_clrqhd-clrno.*  
            IF lt\_clrqdf IS NOT INITIAL.  
              SELECT \* FROM vbrk INTO TABLE lt\_vbrk\_df  
                FOR ALL ENTRIES IN lt\_clrqdf  
                WHERE vbeln = lt\_clrqdf-sbqdc.  
            ENDIF.  
            IF sy-subrc EQ 0.  
              lv\_stufe  =   lv\_stufe + 1.  
              READ TABLE lt\_clrqdf INTO lwa\_clrqdf WITH KEY sbqdc = is\_bill\_data-vbeln.  
              IF sy-subrc EQ 0.  
                DATA(lv\_index) = sy-tabix.  
              ENDIF.  
              LOOP AT lt\_clrqdf INTO lwa\_clrqdf FROM lv\_index.  
  
                IF lwa\_clrqdf-prcdc IS NOT INITIAL .  
                  READ TABLE lt\_vbrk\_df INTO lwa\_vbrk\_p  
                  WITH KEY vbeln = lwa\_clrqdf-prcdc.  
                ENDIF.  
                IF lwa\_clrqdf-sbqdc IS NOT INITIAL .  
                  READ TABLE lt\_vbrk\_df INTO lwa\_vbrk\_s  
                  WITH KEY vbeln = lwa\_clrqdf-sbqdc.  
                ENDIF.  
  
                IF lwa\_clrqdf-prcdc IS INITIAL AND lwa\_clrqdf-sbqdc = is\_bill\_data-vbeln.  
                  READ TABLE ct\_vbfas\_tab  INTO lwa\_vbrf\_df  
                  WITH KEY vbeln = lwa\_clrqdf-clrno.  
  
                  lwa\_vbfa-vbelv = lwa\_clrqdf-clrno.  
                  lwa\_vbfa-vbeln = lwa\_clrqdf-sbqdc.  
                  lwa\_vbfa-vbtyp\_v = lwa\_clrqhd-clrqt.  
                  lwa\_vbfa-vbtyp\_n = lwa\_vbrk\_s-vbtyp.  
                  lwa\_vbfa-erdat = lwa\_clrqdf-erdat.  
                  lwa\_vbfa-erzet = lwa\_clrqdf-erzet.  
                  lwa\_vbfa-rfwrt = lwa\_clrqdf-netwr.  
                  lwa\_vbfa-waers = lwa\_clrqdf-waerk.  
                  lwa\_vbfa-stufe = lwa\_vbrf\_df-stufe + 1.  
                  DATA(lwa\_clrqdf\_tmp) = lwa\_clrqdf.  
                ELSEIF ( lwa\_clrqdf\_tmp-sbqdc EQ lwa\_clrqdf-prcdc ).  
                  READ TABLE ct\_vbfas\_tab  INTO lwa\_vbrf\_df  
                  WITH KEY vbeln = lwa\_clrqdf-prcdc.  
  
                  lwa\_vbfa-vbelv = lwa\_clrqdf-prcdc.  
                  lwa\_vbfa-vbeln = lwa\_clrqdf-sbqdc.  
                  lwa\_vbfa-vbtyp\_v = lwa\_vbrk\_p-vbtyp.  
                  lwa\_vbfa-vbtyp\_n = lwa\_vbrk\_s-vbtyp.  
                  lwa\_vbfa-erdat = lwa\_clrqdf-erdat.  
                  lwa\_vbfa-erzet = lwa\_clrqdf-erzet.  
                  lwa\_vbfa-rfwrt = lwa\_clrqdf-netwr.  
                  lwa\_vbfa-waers = lwa\_clrqdf-waerk.  
                  lwa\_vbfa-stufe = lwa\_vbrf\_df-stufe + 1.  
                  lwa\_clrqdf\_tmp = lwa\_clrqdf.  
                ELSE.  
                  EXIT.  
                ENDIF.  
                lv\_index = lv\_index + 1.  
                lwa\_vbfa-mandt  = sy-mandt.  
                APPEND lwa\_vbfa TO lt\_vbfa\_tab.  
                CLEAR : lwa\_vbfa.  
              ENDLOOP.  
              ct\_vbfas\_tab[] = lt\_vbfa\_tab[].  
            ENDIF.  
          ENDIF.  
        ENDIF.  
      ENDIF.  
    ENDIF.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>GET\_ERROR\_CONFG*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [<---] ET\_CLRQ\_ERR                    TYPE        GTT\_CLRQ\_ERR*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD get\_error\_confg.  
  
    IF gt\_clrq\_err IS INITIAL.  
      SELECT \* FROM /cgdc/t\_clrq\_err  
           INTO TABLE gt\_clrq\_err.  
    ENDIF.  
  
    et\_clrq\_err = gt\_clrq\_err.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>GET\_PERFORMANCE\_DATE*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IS\_CLRQHDR                     TYPE        /CGDC/S\_CLRQHD(optional)*  
*\* | [--->] IS\_CLRQIT                      TYPE        /CGDC/S\_CLRQIT(optional)*  
*\* | [--->] IS\_CLRQBI                      TYPE        /CGDC/S\_CLRQBI(optional)*  
*\* | [--->] IV\_OBJTY                       TYPE        CHAR1*  
*\* | [--->] IS\_CLRQGRR                     TYPE        /CGDC/S\_CLRQGRR(optional)*  
*\* | [--->] IV\_DATE\_TY                     TYPE        /CGDC/CLRQ\_CUST\_DT\_RULE\_TYPE(optional)*  
*\* | [<---] EV\_DATE                        TYPE        DATS*  
*\* | [EXC!] EX\_ITEM\_NOT\_PASSED*  
*\* | [EXC!] EX\_CONTRACT\_NOT\_PASSED*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD get\_performance\_date.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: GET\_PERFORMANCE\_DATE*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
    DATA lwa\_critm TYPE /cgdc/\_clrqit.  
    DATA lwa\_clrbi TYPE /cgdc/\_clrqbi.  
    DATA lwa\_clrhd TYPE /cgdc/\_clrqhd.  
    DATA  lt\_clrgrr TYPE TABLE OF /cgdc/\_clrqgrr.  
    DATA  lwa\_clrgrr TYPE /cgdc/\_clrqgrr.  
    DATA  lt\_clrbi TYPE TABLE OF /cgdc/\_clrqbi.  
    DATA  lt\_vbfa TYPE TABLE OF vbfa.  
    DATA lwa\_vbfa TYPE vbfa.  
    DATA: lwa\_dtrule TYPE /cgdc/t\_dtrule,  
          lv\_signum,  
          lwa\_bsdate TYPE /cgdc/t\_basdat,  
          lwa\_billty TYPE /cgdc/t\_bilitty.  
    DATA: lv\_holiday\_error TYPE flag.  
    DATA : lwa\_t5a4a TYPE t5a4a.  
  
    DATA : lt\_ptab  TYPE abap\_parmbind\_tab,  
           lwa\_ptab LIKE LINE OF lt\_ptab.  
    DATA: ex\_ref TYPE REF TO cx\_sy\_dyn\_call\_error.  
  
    DATA : lt\_data TYPE REF TO data.  
    FIELD-SYMBOLS : <lwa\_date> TYPE /cgdc/s\_clrq\_popdate.  
    CREATE DATA lt\_data TYPE /cgdc/s\_clrq\_popdate.  
    ASSIGN lt\_data->\* TO <lwa\_date>.  
  
*\*    lwa\_ptab-kind = cl\_abap\_objectdescr=>changing.*  
*\*    lwa\_ptab-name = 'GS\_PERFORMANCE\_DATE'.*  
*\*    lwa\_ptab-value = gs\_performance\_date.*  
*\*    APPEND lwa\_ptab TO lt\_ptab.*  
  
*\*    gs\_clrit = is\_clrqit.   "gs\_clrit kumar to review*  
    CASE iv\_objty.  
      WHEN 'R'.  
        MOVE-CORRESPONDING is\_clrqit TO gs\_performance\_date.  
        gs\_performance\_date-vbeln = is\_clrqit-vgbel.  
        gs\_performance\_date-posnr = is\_clrqit-vgpos.  
        gs\_performance\_date-pspnr = is\_clrqit-ps\_psp\_pnr.  
        gs\_performance\_date-psphi = is\_clrqbi-ps\_psp\_pnr.  
        SELECT SINGLE psphi  
        FROM  prps  
        INTO gs\_performance\_date-psphi  
        WHERE pspnr = is\_clrqbi-ps\_psp\_pnr.  
        IF is\_clrqbi IS NOT INITIAL.  
*\*          lwa\_clrbi = is\_clrqbi.*  
          MOVE-CORRESPONDING is\_clrqbi TO lwa\_clrbi.  
        ELSE.  
          SELECT SINGLE \*  
           FROM /cgdc/\_clrqbi  
           INTO lwa\_clrbi  
           WHERE bitype = is\_clrqit-bityp AND bill\_item = is\_clrqit-rtbbinr.  
        ENDIF.  
  
  
      WHEN 'G'.  
        IF is\_clrqhdr IS INITIAL.  
          SELECT SINGLE vgbel  
            FROM /cgdc/\_clrqhd  
            INTO gs\_performance\_date-vbeln  
            WHERE clrno = is\_clrqgrr-clrno.  
        ELSE.  
          gs\_performance\_date-vbeln = is\_clrqhdr-vgbel.  
        ENDIF.  
      WHEN 'B'.  
  
        gs\_performance\_date-belnr = is\_clrqit-belnr.  
        gs\_performance\_date-buzei = is\_clrqit-buzei.  
*\*        gs\_performance\_date-gjahr = is\_clrqit-gjahr.*  
        gs\_performance\_date-bukrs = is\_clrqhdr-bukrs.  
        MOVE-CORRESPONDING is\_clrqit TO gs\_performance\_date.  
*\*        lwa\_clrbi = is\_clrqbi.*  
        MOVE-CORRESPONDING is\_clrqbi TO lwa\_clrbi.  
        gs\_performance\_date-vbeln = is\_clrqbi-vgbel.  
        gs\_performance\_date-posnr = is\_clrqbi-vgpos.  
        gs\_performance\_date-pspnr = is\_clrqbi-ps\_psp\_pnr.  
  
        *"++SOC - Performance optimization*  
*\*        SELECT SINGLE psphi*  
*\*          FROM  prps*  
*\*          INTO gs\_performance\_date-psphi*  
*\*          WHERE pspnr = is\_clrqbi-ps\_psp\_pnr.*  
        gs\_performance\_date-psphi = read\_prps\_psphi( iv\_ps\_psp\_pnr = is\_clrqbi-ps\_psp\_pnr ).  
        *"++EOC - Performance optimization*  
  
      WHEN 'C'.  
        gs\_performance\_date-vbeln = is\_clrqit-vgbel.  
      WHEN 'P'.  
*\*        SELECT \**  
*\*             FROM vbfa*  
*\*             INTO TABLE lt\_vbfa*  
*\*             WHERE vbeln = iv\_invoice AND vbtyp\_v = 'G'.*  
*\*        IF sy-subrc = 0.*  
*\*          READ TABLE lt\_vbfa INTO lwa\_vbfa INDEX 1.*  
*\*          gs\_performance\_date-vbeln = lwa\_vbfa-vbelv.*  
*\*        ENDIF.*  
      WHEN 'I'.  
  
    ENDCASE.  
  
  
  
  
  
    SELECT SINGLE \* FROM /cgdc/t\_bilitty  
      INTO lwa\_billty  
      WHERE bilitmtyp = lwa\_clrbi-bitype.  
    IF sy-subrc = 0.  
      CASE iv\_date\_ty.  
        WHEN 'B'.  
          SELECT SINGLE \* FROM /cgdc/t\_dtrule  
            INTO lwa\_dtrule  
            WHERE regel = lwa\_billty-billing\_date\_rule.  
        WHEN 'S'.  
          SELECT SINGLE \* FROM /cgdc/t\_dtrule  
        INTO lwa\_dtrule  
        WHERE regel = lwa\_billty-pop\_rule\_beg.  
        WHEN 'E'.  
          SELECT SINGLE \* FROM /cgdc/t\_dtrule  
       INTO lwa\_dtrule  
       WHERE regel = lwa\_billty-pop\_rule\_end.  
      ENDCASE.  
    ENDIF.  
    IF lwa\_dtrule IS NOT INITIAL.  
      SELECT SINGLE \*  
        FROM /cgdc/t\_basdat  
        INTO lwa\_bsdate  
        WHERE basdat = lwa\_dtrule-basdat.  
      IF sy-subrc = 0.  
*\*if fixed date is maintaned pass it to ev\_date rather than calling class and menthods*  
        IF lwa\_bsdate-fixeddate IS INITIAL.                                     *"+DCLM-5651*  
*\*        MOVE-CORRESPONDING gs\_performance\_date TO /cgdc/cl\_clrq\_baseline\_date\_ru=>gs\_performance\_date.*  
*\*        CALL METHOD (lwa\_bsdate-class)=>(lwa\_bsdate-method).*  
  
          <lwa\_date> = gs\_performance\_date.  
          lwa\_ptab-kind = cl\_abap\_objectdescr=>changing.  
          lwa\_ptab-name = 'GS\_PERFORMANCE\_DATE'.  
          lwa\_ptab-value = lt\_data.  
          INSERT lwa\_ptab INTO TABLE lt\_ptab.  
          TRY.  
              CALL METHOD (lwa\_bsdate-class)=>(lwa\_bsdate-method)  
                PARAMETER-TABLE  
                lt\_ptab.  
  
            CATCH cx\_sy\_dyn\_call\_error INTO ex\_ref.  
*\*            MESSAGE exc\_ref->get\_text( ) TYPE 'I'.*  
          ENDTRY.  
          ev\_date = <lwa\_date>-pop\_date.  
  
          CALL METHOD /cgdc/cl\_clrq\_process=>adjust\_basedate *"adjustment Date*  
            EXPORTING  
              iv\_adjst\_type = lwa\_bsdate-adjust  
            CHANGING  
              c\_date        = ev\_date.  
  
          IF lwa\_dtrule-zeitr LE 0.  
            lv\_signum = '-'.  
          ELSE.  
            lv\_signum = '+'.  
          ENDIF.  
  
          CASE lwa\_dtrule-zeite.  
            WHEN 1.  
              lwa\_t5a4a-dlydy = lwa\_dtrule-zeitr.  
  
              CALL FUNCTION 'RP\_CALC\_DATE\_IN\_INTERVAL'  
                EXPORTING  
                  date      = ev\_date  
                  days      = lwa\_t5a4a-dlydy  
                  months    = 00  
                  signum    = lv\_signum  
                  years     = 00  
                IMPORTING  
                  calc\_date = ev\_date.  
              *"Mathematically adjusting the date to not go over by 1 day.*  
              IF iv\_date\_ty EQ 'E'.  
                ev\_date = ev\_date - 1.  
              ENDIF.  
            WHEN 2.  
*\*            ev\_date = ev\_date \* ( lwa\_dtrule-zeitr \* 7 ).*  
              ev\_date = ev\_date + ( lwa\_dtrule-zeitr \* 7 ).  
              *"Mathematically adjusting the date to not go over by 1 day.*  
              IF iv\_date\_ty EQ 'E'.  
                ev\_date = ev\_date - 1.  
              ENDIF.  
            WHEN 3.  
              lwa\_t5a4a-dlymo = lwa\_dtrule-zeitr.  
              CALL FUNCTION 'RP\_CALC\_DATE\_IN\_INTERVAL'  
                EXPORTING  
                  date      = ev\_date  
                  days      = 00  
                  months    = lwa\_t5a4a-dlymo  
                  signum    = lv\_signum  
                  years     = 00  
                IMPORTING  
                  calc\_date = ev\_date.  
              *"Mathematically adjusting the date to not go over by 1 day.*  
              IF iv\_date\_ty EQ 'E'.  
                ev\_date = ev\_date - 1.  
              ENDIF.  
            WHEN 4.  
              lwa\_t5a4a-dlyyr = lwa\_dtrule-zeitr.  
              CALL FUNCTION 'RP\_CALC\_DATE\_IN\_INTERVAL'  
                EXPORTING  
                  date      = ev\_date  
                  days      = 00  
                  months    = 00  
                  signum    = lv\_signum  
                  years     = lwa\_t5a4a-dlyyr  
                IMPORTING  
                  calc\_date = ev\_date.  
  
              *"Mathematically adjusting the date to not go over by 1 day.*  
              IF iv\_date\_ty EQ 'E'.  
                ev\_date = ev\_date - 1.  
              ENDIF.  
  
          ENDCASE.  
          DATA : lv\_calid TYPE wfcid.  
          lv\_calid = lwa\_dtrule-perca.  
          CALL FUNCTION 'DATE\_CHECK\_WORKINGDAY'  
            EXPORTING  
              date                       = ev\_date  
              factory\_calendar\_id        = lv\_calid  
              message\_type               = 'I'  
            EXCEPTIONS  
              date\_after\_range           = 1  
              date\_before\_range          = 2  
              date\_invalid               = 3  
              date\_no\_workingday         = 4  
              factory\_calendar\_not\_found = 5  
              message\_type\_invalid       = 6  
              OTHERS                     = 7.  
          IF sy-subrc <> 0.  
            CASE lwa\_dtrule-wrkdy.  
              WHEN ''.  
                DO 7 TIMES.  
                  ev\_date = ev\_date + 1.  
                  CALL FUNCTION 'DATE\_CHECK\_WORKINGDAY'  
                    EXPORTING  
                      date                       = ev\_date  
                      factory\_calendar\_id        = lv\_calid  
                      message\_type               = 'I'  
                    EXCEPTIONS  
                      date\_after\_range           = 1  
                      date\_before\_range          = 2  
                      date\_invalid               = 3  
                      date\_no\_workingday         = 4  
                      factory\_calendar\_not\_found = 5  
                      message\_type\_invalid       = 6  
                      OTHERS                     = 7.  
                  IF sy-subrc = 0.  
                    EXIT.  
                  ENDIF.  
                ENDDO.  
              WHEN 'P'.  
                DO 7 TIMES.  
                  ev\_date = ev\_date - 1.  
                  CALL FUNCTION 'DATE\_CHECK\_WORKINGDAY'  
                    EXPORTING  
                      date                       = ev\_date  
                      factory\_calendar\_id        = lwa\_dtrule-perca  
                      message\_type               = 'I'  
                    EXCEPTIONS  
                      date\_after\_range           = 1  
                      date\_before\_range          = 2  
                      date\_invalid               = 3  
                      date\_no\_workingday         = 4  
                      factory\_calendar\_not\_found = 5  
                      message\_type\_invalid       = 6  
                      OTHERS                     = 7.  
                  IF sy-subrc = 0.  
                    EXIT.  
                  ENDIF.  
                ENDDO.  
              WHEN 'N'.  
            ENDCASE.  
  
          ENDIF.  
        ELSE.                                                             *"+DCLM-5651*  
          ev\_date = lwa\_bsdate-fixeddate.                                 *"+DCLM-5651*  
        ENDIF.                                                            *"+DCLM-5651*  
  
      ENDIF.  
  
  
    ENDIF.  
  
*\*//BOC-DCLM 4872*  
    IF ev\_date IS NOT INITIAL AND  
       iv\_objty EQ 'B' AND  
       iv\_date\_ty EQ 'E'.  
      validate\_popend\_date( EXPORTING iv\_vbeln        = is\_clrqit-/cgdc/orgh  
                                      iv\_posnr        = is\_clrqit-/cgdc/orgi  
                            CHANGING  cv\_pop\_end\_date = ev\_date ).  
    ENDIF.  
*\*//EOC-DCLM 4872*  
  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>GET\_PERFORMANCE\_DATE\_BRF*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IS\_VBAP                        TYPE        VBAP(optional)*  
*\* | [--->] IV\_DATERULE                    TYPE        VDREG\_VEDA*  
*\* | [<---] EV\_DATE                        TYPE        DATS*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD get\_performance\_date\_brf.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2022] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-211COINCGDC (CIS-AD S/4 Hana 2021)*  
*\**  
*\* DESCRIPTION: DCLM 5651-Get start and end date from date rule from BRF*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
    FIELD-SYMBOLS : <lwa\_date> TYPE /cgdc/s\_clrq\_popdate.  
    DATA : lt\_ptab  TYPE abap\_parmbind\_tab,  
           lwa\_ptab LIKE LINE OF lt\_ptab.  
    DATA: ex\_ref TYPE REF TO cx\_sy\_dyn\_call\_error.  
    DATA : lt\_data   TYPE REF TO data,  
           lwa\_t5a4a TYPE t5a4a.  
  
    CREATE DATA lt\_data TYPE /cgdc/s\_clrq\_popdate.  
    ASSIGN lt\_data->\* TO <lwa\_date>.  
  
    gs\_performance\_date-vbeln = is\_vbap-vbeln.  
    gs\_performance\_date-posnr = is\_vbap-posnr.  
    gs\_performance\_date-pspnr = is\_vbap-ps\_psp\_pnr.  
    gs\_performance\_date-psphi = read\_prps\_psphi( iv\_ps\_psp\_pnr = is\_vbap-ps\_psp\_pnr ).  
  
*\*\*Fetch Baseline Date from date rule to fetch the class and Method name*  
    SELECT SINGLE \*  
      FROM /cgdc/t\_dtrule  
      INTO @DATA(lwa\_dtrule)  
      WHERE regel = @iv\_daterule.  
    IF sy-subrc = 0.  
      SELECT SINGLE \*  
       FROM /cgdc/t\_basdat  
       INTO @DATA(lwa\_basdat)  
       WHERE basdat = @lwa\_dtrule-basdat.  
      IF sy-subrc = 0.  
*\*Pass the Fixed date if that is having any value*  
        IF lwa\_basdat-fixeddate IS INITIAL.  
  
          <lwa\_date> = gs\_performance\_date.  
          lwa\_ptab-kind = cl\_abap\_objectdescr=>changing.  
          lwa\_ptab-name = 'GS\_PERFORMANCE\_DATE'.  
          lwa\_ptab-value = lt\_data.  
          INSERT lwa\_ptab INTO TABLE lt\_ptab.  
          TRY.  
              CALL METHOD (lwa\_basdat-class)=>(lwa\_basdat-method)  
                PARAMETER-TABLE  
                lt\_ptab.  
  
            CATCH cx\_sy\_dyn\_call\_error INTO ex\_ref.  
*\*            MESSAGE exc\_ref->get\_text( ) TYPE 'I'.*  
          ENDTRY.  
          ev\_date = <lwa\_date>-pop\_date.  
  
          IF ev\_date IS NOT INITIAL.  
*\*\*Call the respective method of respective class*  
            CALL METHOD /cgdc/cl\_clrq\_process=>adjust\_basedate *"adjustment Date*  
              EXPORTING  
                iv\_adjst\_type = lwa\_basdat-adjust  
              CHANGING  
                c\_date        = ev\_date.  
  
            CASE lwa\_dtrule-zeite.  
              WHEN 1.  
                lwa\_t5a4a-dlydy = lwa\_dtrule-zeitr.  
                CALL FUNCTION 'RP\_CALC\_DATE\_IN\_INTERVAL'  
                  EXPORTING  
                    date      = ev\_date  
                    days      = lwa\_t5a4a-dlydy  
                    months    = 00  
*\*                   SIGNUM    = '+'*  
                    years     = 00  
                  IMPORTING  
                    calc\_date = ev\_date.  
              WHEN 2.  
                ev\_date = ev\_date + ( lwa\_dtrule-zeitr \* 7 ).  
  
              WHEN 3.  
                lwa\_t5a4a-dlymo = lwa\_dtrule-zeitr.  
                CALL FUNCTION 'RP\_CALC\_DATE\_IN\_INTERVAL'  
                  EXPORTING  
                    date      = ev\_date  
                    days      = 00  
                    months    = lwa\_t5a4a-dlymo  
*\*                   SIGNUM    = '+'*  
                    years     = 00  
                  IMPORTING  
                    calc\_date = ev\_date.  
  
              WHEN 4.  
                lwa\_t5a4a-dlyyr = lwa\_dtrule-zeitr.  
                CALL FUNCTION 'RP\_CALC\_DATE\_IN\_INTERVAL'  
                  EXPORTING  
                    date      = ev\_date  
                    days      = 00  
                    months    = 00  
*\*                   SIGNUM    = '+'*  
                    years     = lwa\_t5a4a-dlyyr  
                  IMPORTING  
                    calc\_date = ev\_date.  
            ENDCASE.  
            DATA : lv\_calid TYPE wfcid.  
            lv\_calid = lwa\_dtrule-perca.  
            CALL FUNCTION 'DATE\_CHECK\_WORKINGDAY'  
              EXPORTING  
                date                       = ev\_date  
                factory\_calendar\_id        = lv\_calid  
                message\_type               = 'I'  
              EXCEPTIONS  
                date\_after\_range           = 1  
                date\_before\_range          = 2  
                date\_invalid               = 3  
                date\_no\_workingday         = 4  
                factory\_calendar\_not\_found = 5  
                message\_type\_invalid       = 6  
                OTHERS                     = 7.  
            IF sy-subrc <> 0.  
              CASE lwa\_dtrule-wrkdy.  
                WHEN ''.  
                  DO 7 TIMES.  
                    ev\_date = ev\_date + 1.  
                    CALL FUNCTION 'DATE\_CHECK\_WORKINGDAY'  
                      EXPORTING  
                        date                       = ev\_date  
                        factory\_calendar\_id        = lv\_calid  
                        message\_type               = 'I'  
                      EXCEPTIONS  
                        date\_after\_range           = 1  
                        date\_before\_range          = 2  
                        date\_invalid               = 3  
                        date\_no\_workingday         = 4  
                        factory\_calendar\_not\_found = 5  
                        message\_type\_invalid       = 6  
                        OTHERS                     = 7.  
                    IF sy-subrc = 0.  
                      EXIT.  
                    ENDIF.  
                  ENDDO.  
                WHEN 'P'.  
                  DO 7 TIMES.  
                    ev\_date = ev\_date - 1.  
                    CALL FUNCTION 'DATE\_CHECK\_WORKINGDAY'  
                      EXPORTING  
                        date                       = ev\_date  
                        factory\_calendar\_id        = lwa\_dtrule-perca  
                        message\_type               = 'I'  
                      EXCEPTIONS  
                        date\_after\_range           = 1  
                        date\_before\_range          = 2  
                        date\_invalid               = 3  
                        date\_no\_workingday         = 4  
                        factory\_calendar\_not\_found = 5  
                        message\_type\_invalid       = 6  
                        OTHERS                     = 7.  
                    IF sy-subrc = 0.  
                      EXIT.  
                    ENDIF.  
                  ENDDO.  
                WHEN 'N'.  
              ENDCASE.  
  
            ENDIF.  
          ENDIF.  
        ELSE.  
          ev\_date = lwa\_basdat-fixeddate.  
        ENDIF.  
      ENDIF.  
    ENDIF.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>GET\_PREVIOUSLY\_BILLED*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IV\_BILL\_MATERIAL               TYPE        MATNR(optional)*  
*\* | [--->] IV\_BILL\_DOC                    TYPE        VBELN*  
*\* | [--->] IV\_SPLIT\_BY\_BIL\_ELEMENT        TYPE        CHAR1 (default =SPACE)*  
*\* | [<---] ET\_PREV\_BILL\_DETAILS           TYPE        /CGDC/T\_KONV\_DET*  
*\* | [<---] ET\_PREV\_BILL\_SUMMARY           TYPE        /CGDC/T\_VBRP\_DET*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD get\_previously\_billed.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Get Previously Billed BDR Value*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
    TYPES :BEGIN OF ty\_vbrp,  
             knumv TYPE v\_konv-knumv,  
             blelm TYPE ps\_psp\_pnr.  
             INCLUDE TYPE vbrp.  
    TYPES : END OF ty\_vbrp.  
  
    DATA : lt\_vbrp          TYPE  TABLE OF ty\_vbrp,  
           lt\_vbrp\_collect  TYPE  /cgdc/t\_vbrp\_det,  
           lwa\_vbrp         TYPE ty\_vbrp,  
           lwa\_vbrp\_collect TYPE LINE OF /cgdc/t\_vbrp\_det,  
           lwa\_konv\_det     TYPE /cgdc/s\_konv\_det,  
           it\_matnr         TYPE RANGE OF mara-matnr,  
           lwa\_bill\_summ    LIKE LINE OF et\_prev\_bill\_summary,  
           lv\_bdr\_found     TYPE abap\_bool.  
    FIELD-SYMBOLS: <lwa\_vbrp\_collect> TYPE LINE OF /cgdc/t\_vbrp\_det.  
    IF iv\_bill\_material IS NOT INITIAL.  
      it\_matnr = VALUE #( ( sign = 'I'  
                            option = 'EQ'  
                            low = iv\_bill\_material ) )  .  
    ENDIF.  
  
    CLEAR lv\_bdr\_found.  
    SELECT SINGLE \*  
      FROM /cgdc/\_clrqdf  
      WHERE sbqdc = @iv\_bill\_doc  
      INTO @DATA(lwa\_clrqdf) .  
  
    IF lwa\_clrqdf-sbdct = 'BDR'.  
      lv\_bdr\_found = abap\_true.  
    ELSE.  
      SELECT \* FROM /cgdc/\_clrqdf  
        WHERE clrno = @lwa\_clrqdf-clrno  
        ORDER BY PRIMARY KEY  
        INTO TABLE @DATA(lt\_clrqdf) .  
      IF sy-subrc IS INITIAL.  
        WHILE lv\_bdr\_found IS INITIAL .  
          lwa\_clrqdf = VALUE #( lt\_clrqdf[ sbqdc = lwa\_clrqdf-prcdc ] OPTIONAL ).  
          IF lwa\_clrqdf-sbdct = 'BDR'.  
            lv\_bdr\_found = abap\_true.  
          ENDIF.  
        ENDWHILE.  
      ENDIF.  
    ENDIF.  
  
    SELECT DISTINCT c~vbeln ,  
           c~knumv,  
           c~bdr\_status  
*\*           b~ps\_psp\_pnr*  
       FROM /cgdc/\_clrqdf AS a  
      INNER JOIN vbrk AS c ON  
      c~vbeln EQ a~sbqdc   AND  
      c~bdr\_status NE 'D'  
*\*      INNER JOIN /cgdc/\_clrqbi as b*  
*\*      ON a~clrno = b~clrno*  
*\*      AND a~rtbbinr = b~bill\_item*  
      WHERE a~sbdct EQ 'BDR'              AND  
            a~clrno EQ @lwa\_clrqdf-clrno  AND  
            a~erdat LE @lwa\_clrqdf-erdat  AND  
            a~erzet LT @lwa\_clrqdf-erzet  
      GROUP BY c~vbeln ,  
               c~knumv,  
               c~bdr\_status  
*\*               b~ps\_psp\_pnr*  
      INTO TABLE @DATA(lt\_vbrk).  
  
    SELECT  b~knumv, a~\* , d~ps\_psp\_pnr AS blelm  
      FROM vbrp AS a  
      INNER JOIN @lt\_vbrk AS b ON  
      b~vbeln = a~vbeln  
  
      INNER JOIN /cgdc/\_clrqdf AS c  
      ON a~vbeln EQ c~sbqdc  
      AND a~posnr EQ c~sbqit  
*\*      a~bdr\_status NE 'D'*  
  
      INNER JOIN /cgdc/\_clrqbi AS d  
      ON c~clrno = d~clrno  
      AND c~rtbbinr = d~bill\_item  
  
      WHERE a~matnr IN @it\_matnr  
      ORDER BY a~matnr , d~ps\_psp\_pnr  
      INTO CORRESPONDING FIELDS OF TABLE @lt\_vbrp.  
  
    SELECT *"a~knumv,a~kposn,*  
      b~matnr,b~ps\_psp\_pnr, a~kschl,a~kwert, a~kstat, a~kntyp, b~vbeln  
      FROM v\_konv AS a  
      INNER JOIN @lt\_vbrp AS b ON  
      b~knumv = a~knumv        AND  
      b~posnr = a~kposn  
      ORDER BY b~matnr, b~ps\_psp\_pnr  
      INTO TABLE @DATA(lt\_konv).  
*\*      IF iv\_split\_by\_bil\_element IS INITIAL.*  
*\*        SELECT ROW\_NUMBER( ) OVER( ORDER BY matnr  ) AS kposn ,*  
*\*          a~matnr ,*  
*\*          SUM( a~netwr ) AS netwr*  
*\*          FROM @lt\_vbrp AS a*  
*\*          GROUP BY a~matnr*  
*\*          ORDER BY a~matnr*  
*\*          INTO TABLE @lt\_vbrp\_collect.*  
*\*      ELSE.*  
*\*        SELECT ROW\_NUMBER( ) OVER( ORDER BY matnr, ps\_psp\_pnr  ) AS kposn ,*  
*\*          a~matnr ,*  
*\*          a~ps\_psp\_pnr,*  
*\*          SUM( a~netwr ) AS netwr*  
*\*          FROM @lt\_vbrp AS a*  
*\*          GROUP BY a~matnr, a~ps\_psp\_pnr*  
*\*          ORDER BY a~matnr, a~ps\_psp\_pnr*  
*\*          INTO TABLE @lt\_vbrp\_collect.*  
*\*      ENDIF.*  
    LOOP AT lt\_vbrp INTO lwa\_vbrp.  
      lwa\_vbrp\_collect-matnr = lwa\_vbrp-matnr.  
      IF iv\_split\_by\_bil\_element IS NOT INITIAL.  
        lwa\_vbrp\_collect-ps\_psp\_pnr = lwa\_vbrp-blelm.  
      ENDIF.  
      lwa\_vbrp\_collect-netwr = lwa\_vbrp-netwr.  
      lwa\_vbrp\_collect-waerk = lwa\_vbrp-waerk.  
      lwa\_vbrp\_collect-fkimg = lwa\_vbrp-fkimg. *"ZTMB LINE ITEM DATA*  
      lwa\_vbrp\_collect-/cgdc/incrd\_yr = lwa\_vbrp-/cgdc/incrd\_yr.  
      COLLECT lwa\_vbrp\_collect INTO lt\_vbrp\_collect.  
    ENDLOOP.  
    LOOP AT lt\_vbrp\_collect ASSIGNING <lwa\_vbrp\_collect>.  
      <lwa\_vbrp\_collect>-kposn = sy-tabix.  
    ENDLOOP.  
    et\_prev\_bill\_summary = CORRESPONDING #( lt\_vbrp\_collect ).  
  
    LOOP AT lt\_konv INTO DATA(lwa\_konv) .  
  
*\*      wa\_konv\_det-kposn =  wa\_konv\_det-kposn + 1 .*  
      lwa\_konv\_det-kschl = lwa\_konv-kschl.  
      lwa\_konv\_det-kstat = lwa\_konv-kstat.  
      lwa\_konv\_det-kntyp = lwa\_konv-kntyp.  
      lwa\_konv\_det-kwert = lwa\_konv-kwert.  
  
*\*      AT NEW matnr.*  
*\*        lwa\_konv\_det-kposn = lwa\_konv\_det-kposn + 1.*  
*\*      ENDAT.*  
      IF iv\_split\_by\_bil\_element IS INITIAL.  
        READ TABLE lt\_vbrp\_collect INTO DATA(lwa\_vbrp\_det)  
        WITH KEY matnr = lwa\_konv-matnr.  
*\*                  ps\_psp\_pnr = lwa\_konv-ps\_psp\_pnr.*  
        lwa\_konv\_det-kposn = lwa\_vbrp\_det-kposn.  
      ELSE.  
        READ TABLE lt\_vbrp\_collect INTO DATA(lwa\_vbrp\_det\_2)  
        WITH KEY matnr = lwa\_konv-matnr  
                 ps\_psp\_pnr = lwa\_konv-ps\_psp\_pnr.  
        lwa\_konv\_det-kposn = lwa\_vbrp\_det\_2-kposn.  
      ENDIF.  
      COLLECT lwa\_konv\_det INTO et\_prev\_bill\_details.  
    ENDLOOP.  
  
  
  
  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>MAP\_CLRQSRC\_DATA*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IS\_DATA                        TYPE        DATA*  
*\* | [--->] IV\_TABNAME                     TYPE        TABNAME*  
*\* | [--->] IT\_SRC\_FIELDS                  TYPE        /CGDC/TT\_DD03L*  
*\* | [<-->] CS\_CLRQSRC                     TYPE        /CGDC/DCO\_SRC*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD map\_clrqsrc\_data.  
    DATA: lt\_dd03l\_data  TYPE  TABLE OF dd03l,  
          lwa\_dd03l\_data TYPE dd03l.  
  
    FIELD-SYMBOLS: <lwa\_data>    TYPE any,  
                   <lwa\_value\_d> TYPE any,  
                   <lwa\_value\_s> TYPE any.  
  
    ASSIGN is\_data TO <lwa\_data>.  
    MOVE-CORRESPONDING <lwa\_data> TO cs\_clrqsrc.  
  
    CASE iv\_tabname.  
      WHEN 'MSEG'.  
  
    ENDCASE.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>RAISE\_COST\_ELEMENT\_HDR\_MSG*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD raise\_cost\_element\_hdr\_msg.  
    FIELD-SYMBOLS : <lwa\_clrqhd> TYPE /cgdc/s\_clrqhd.  
    FIELD-SYMBOLS : <lwa\_clrq\_doc> TYPE /cgdc/s\_clrq\_doc.  
    DATA : lt\_clrqhd   TYPE TABLE OF /cgdc/s\_clrqhd,  
           lt\_clrq\_doc TYPE TABLE OF /cgdc/s\_clrq\_doc.  
    SELECT \*  
      FROM /cgdc/\_clrqhd  
      INTO TABLE lt\_clrqhd  
      FOR ALL ENTRIES IN gt\_ce\_clrit\_key  
      WHERE clrno = gt\_ce\_clrit\_key-clrno.  
    IF sy-subrc = 0.  
      LOOP AT lt\_clrqhd ASSIGNING <lwa\_clrqhd>.  
        <lwa\_clrqhd>-msgcode = 'V'.  
        <lwa\_clrqhd>-updkz = 'U'.  
        APPEND INITIAL LINE TO lt\_clrq\_doc ASSIGNING <lwa\_clrq\_doc>.  
        MOVE-CORRESPONDING <lwa\_clrqhd> TO <lwa\_clrq\_doc>-x-crhdr.  
      ENDLOOP.  
      CALL FUNCTION '/CGDC/CLRQ\_SAVE'  
        EXPORTING  
          i\_set\_update\_task = 'X'  
          i\_commit\_work     = ' '  
        CHANGING  
          ct\_clrq\_doc       = lt\_clrq\_doc  
*\*         CS\_CLRQBS\_DOC     =*  
        EXCEPTIONS  
          no\_change         = 1  
          OTHERS            = 2.  
      IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
      ENDIF.  
    ENDIF.  
  
    CLEAR : gt\_ce\_clrit\_key.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>READ\_BCPOP*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IV\_VBELN                       TYPE        VBELN\_VA*  
*\* | [<-()] RV\_BCPOP                       TYPE        /CGDC/BCOPOP*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD read\_bcpop.  
  
    DATA(ls\_bcpop) = VALUE #( gt\_bcpop[ vbeln = iv\_vbeln ] OPTIONAL ).  
    IF ls\_bcpop IS INITIAL.  
      *"read from DB*  
      SELECT SINGLE /cgdc/bcopop  
        FROM vbak  
        INTO @rv\_bcpop  
        WHERE vbeln = @iv\_vbeln.  
      IF sy-subrc = 0.  
        APPEND INITIAL LINE TO gt\_bcpop ASSIGNING FIELD-SYMBOL(<lfs\_bcpop>).  
        <lfs\_bcpop>-vbeln = iv\_vbeln.  
        <lfs\_bcpop>-bcpop = rv\_bcpop.  
      ENDIF.  
    ELSE.  
      rv\_bcpop = ls\_bcpop-bcpop.  
    ENDIF.  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>READ\_CF\_POPEND\_DATE*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IV\_VBELN                       TYPE        VBELN\_VA*  
*\* | [--->] IV\_POSNR                       TYPE        POSNR\_VA*  
*\* | [<-()] RV\_FENDAT                      TYPE        /CGDC/FENDAT*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD read\_cf\_popend\_date.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Read CF popend date for Contract item*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
    DATA(ls\_pop\_endat) = VALUE #( gt\_cf\_pop\_endat[ vbeln = iv\_vbeln posnr = iv\_posnr ] OPTIONAL ).  
  
    IF ls\_pop\_endat IS INITIAL.  
      SELECT SINGLE vbeln, posnr, /cgdc/fendat ,/cgdc/fstdat  
          FROM /cgdc/cf\_sd\_it  
          INTO @ls\_pop\_endat  
          WHERE vbeln = @iv\_vbeln  
            AND posnr = @iv\_posnr.  
      IF sy-subrc = 0.  
        rv\_fendat = ls\_pop\_endat-fendat.  
        APPEND ls\_pop\_endat TO gt\_cf\_pop\_endat.  
      ENDIF.  
    ELSE.  
      rv\_fendat = ls\_pop\_endat-fendat.  
    ENDIF.  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>READ\_PRPS\_PSPHI*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IV\_PS\_PSP\_PNR                  TYPE        /CGDC/PS\_PSP\_PNR*  
*\* | [<-()] RV\_PSPHI                       TYPE        PS\_PSPHI*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD READ\_PRPS\_PSPHI.  
    DATA: ls\_prps TYPE gty\_prps.  
  
    DATA(lv\_psphi) = VALUE #( gt\_prps\_psphi[ pspnr = iv\_ps\_psp\_pnr ]-psphi OPTIONAL ).  
    IF lv\_psphi IS INITIAL.  
      SELECT SINGLE psphi  
      FROM  prps  
      INTO rv\_psphi  
      WHERE pspnr = iv\_ps\_psp\_pnr.  
      IF sy-subrc = 0.  
        ls\_prps-pspnr = iv\_ps\_psp\_pnr.  
        ls\_prps-psphi = rv\_psphi.  
        INSERT ls\_prps INTO TABLE gt\_prps\_psphi.  
      ENDIF.  
    ELSE.  
      rv\_psphi = lv\_psphi.  
    ENDIF.  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>RTB\_AUTHORIZATION*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IV\_CLRQY                       TYPE        /CGDC/CLRQT(optional)*  
*\* | [--->] IV\_VKORG                       TYPE        VKORG(optional)*  
*\* | [--->] IV\_SPART                       TYPE        SPART(optional)*  
*\* | [--->] IV\_FNKEY                       TYPE        /CGDC/RTB\_AUTHFNKEY(optional)*  
*\* | [--->] IV\_RTB\_ACTV                    TYPE        ACTIV\_AUTH(optional)*  
*\* | [--->] IV\_VTWEG                       TYPE        VTWEG(optional)*  
*\* | [--->] IV\_AUART                       TYPE        AUART(optional)*  
*\* | [<---] EV\_INVALID                     TYPE        FLAG*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD rtb\_authorization.  
    CLEAR ev\_invalid.  
    IF iv\_FNKEY IS INITIAL.  
      AUTHORITY-CHECK OBJECT '/CGDC/CLRQ'  
                 ID '/CGDC/CRTY' FIELD iv\_clrqy  
                 ID 'VKORG' FIELD iv\_vkorg  
                 ID 'VTWEG' FIELD iv\_vtweg  
                 ID 'SPART' FIELD iv\_spart  
                 ID '/CGDC/RTAC' FIELD iv\_rtb\_actv.  
*\*                 ID 'AUART' FIELD iv\_auart.*  
      IF sy-subrc <> 0.  
        ev\_invalid = abap\_true.  
      ENDIF.  
    ELSE.  
      AUTHORITY-CHECK OBJECT '/CGDC/CLRQ'  
                 ID '/CGDC/CRTY' FIELD iv\_clrqy  
                 ID 'VKORG' FIELD iv\_vkorg  
                 ID 'VTWEG' FIELD iv\_vtweg  
                 ID 'SPART' FIELD iv\_spart  
                 ID '/CGDC/RTAC' FIELD iv\_rtb\_actv  
                 ID '/CGDC/CRAK' FIELD iv\_fnkey.  
*\*                 ID 'AUART' FIELD iv\_auart.*  
      IF sy-subrc <> 0.  
        ev\_invalid = abap\_true.  
      ENDIF.  
    ENDIF.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>RTB\_DOC\_USAGE\_CHECK*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] I\_VBTYP                        TYPE        VBTYP*  
*\* | [--->] I\_ABRVW                        TYPE        ABRVW*  
*\* | [<---] E\_RC                           TYPE        NUMC1*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD rtb\_doc\_usage\_check.  
    SELECT SINGLE \*  
      FROM /cgdc/t\_rtbufbd  
      INTO @data(lwa\_rtbufbd)  
      WHERE vbtyp = @i\_vbtyp  
      AND abrvw = @i\_abrvw.  
  
  
      IF sy-subrc NE 0 .  
        e\_rc = 0.  
      ELSE.  
        e\_rc = lwa\_rtbufbd-rcode.  
      ENDIF.  
    ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>SET\_ACDOCA*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IT\_ACDOCA                      TYPE        FINST\_ACDOCA*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  method SET\_ACDOCA.  
    gt\_acdoca = it\_acdoca[].  
  endmethod.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>SET\_COBK*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IT\_COBK                        TYPE        TTCOBK*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD set\_cobk.  
    gt\_cobk = it\_cobk.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>SET\_COEP*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IT\_COEP                        TYPE        TT\_COEP*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD set\_coep.  
    gt\_coep = it\_coep[].  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>UPD\_COST\_ELEMENT\_ITEMS*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IV\_COST\_ELEM\_GRP               TYPE        /CGDC/CEGRP*  
*\* | [--->] IV\_COST\_ELEM                   TYPE        KSTAR(optional)*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD upd\_cost\_element\_items.  
    DATA : lt\_rtbcegrp  TYPE TABLE OF /cgdc/t\_rtbaceg,  
           lwa\_rtbcegrp TYPE /cgdc/t\_rtbaceg.  
    DATA : lt\_clrqit  TYPE TABLE OF /cgdc/\_clrqit,  
           lt\_clrqic  TYPE TABLE OF /cgdc/\_clrqic,  
           lwa\_clrqic TYPE /cgdc/\_clrqic,  
           lwa\_clrqit TYPE /cgdc/\_clrqit.  
    DATA : lt\_clrit\_key TYPE /cgdc/tt\_clrqit\_key.  
    DATA : lwa\_clrit\_key TYPE /cgdc/s\_clrqit\_key.  
    DATA : lt\_clrqhd TYPE TABLE OF /cgdc/\_clrqhd.  
  
    SELECT \*  
      FROM /cgdc/t\_rtbaceg  
      INTO TABLE lt\_rtbcegrp  
      WHERE cegroup = iv\_cost\_elem\_grp.  
    IF sy-subrc = 0.  
      SELECT \*  
        FROM  /cgdc/\_clrqhd  
        INTO TABLE lt\_clrqhd  
        FOR ALL ENTRIES IN lt\_rtbcegrp  
        WHERE rtbpn = lt\_rtbcegrp-rtbpn.  
      IF sy-subrc = 0.  
        SELECT \*  
  FROM  /cgdc/\_clrqic  
  INTO TABLE lt\_clrqic  
  FOR ALL ENTRIES IN lt\_clrqhd  
  WHERE clrno = lt\_clrqhd-clrno.  
        SELECT \*  
     FROM  /cgdc/\_clrqit  
     INTO TABLE lt\_clrqit  
     FOR ALL ENTRIES IN lt\_clrqhd  
     WHERE clrno = lt\_clrqhd-clrno.  
        IF sy-subrc = 0.  
          IF iv\_cost\_elem IS NOT INITIAL.  
*\*            DELETE lt\_clrqit WHERE kstar <> iv\_cost\_elem AND cegroup <> iv\_cost\_elem\_grp.*  
            DELETE lt\_clrqic WHERE kstar <> iv\_cost\_elem AND cegroup <> iv\_cost\_elem\_grp.  
          ELSE.  
*\*            DELETE lt\_clrqit WHERE cegroup <> iv\_cost\_elem\_grp.*  
            DELETE lt\_clrqic WHERE cegroup <> iv\_cost\_elem\_grp.  
          ENDIF.  
*\*          LOOP AT lt\_clrqit INTO lwa\_clrqit.*  
          LOOP AT lt\_clrqic INTO lwa\_clrqic.  
            AT NEW clrno.  
              READ TABLE gt\_ce\_clrit\_key TRANSPORTING NO FIELDS WITH KEY clrno = lwa\_clrqit-clrno.  
              READ TABLE gt\_ce\_clrit\_key TRANSPORTING NO FIELDS WITH KEY clrno = lwa\_clrqic-clrno.  
              IF sy-subrc <> 0.  
*\*                lwa\_clrit\_key-clrno = lwa\_clrqit-clrno.*  
                lwa\_clrit\_key-clrno = lwa\_clrqic-clrno.  
                APPEND lwa\_clrit\_key TO gt\_ce\_clrit\_key.  
              ENDIF.  
            ENDAT.  
          ENDLOOP.  
        ENDIF.  
      ENDIF.  
    ENDIF.  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>VALIDATE\_POPEND\_DATE*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IV\_VBELN                       TYPE        VGBEL*  
*\* | [--->] IV\_POSNR                       TYPE        VGPOS*  
*\* | [<-->] CV\_POP\_END\_DATE                TYPE        DATUM*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD validate\_popend\_date.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: CIS-AD: Labor Material Check*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
    CLEAR: gv\_no\_new\_bill\_item.  
  
    READ TABLE gt\_contract\_itm TRANSPORTING NO FIELDS WITH KEY vbeln = iv\_vbeln.  
    IF sy-subrc NE 0.  
      CLEAR: gt\_contract\_itm.  
  
      SELECT \* FROM vbap  
        INTO CORRESPONDING FIELDS OF TABLE gt\_contract\_itm  
        WHERE vbeln = iv\_vbeln  
          AND uepos = '000000'.  
    ENDIF.  
  
    *"--SOC - Performance optimization*  
*\*    SELECT SINGLE /cgdc/fendat*  
*\*          FROM /cgdc/cf\_sd\_it*  
*\*          INTO @DATA(lv\_cf\_pop\_end)*  
*\*          WHERE vbeln = @iv\_vbeln*  
*\*            AND posnr = @iv\_posnr.*  
    DATA(lv\_cf\_pop\_end) = /cgdc/cl\_clrq\_process=>read\_cf\_popend\_date( iv\_vbeln = iv\_vbeln  
                                                                      iv\_posnr = iv\_posnr ).  
    DATA(lv\_cf\_pop\_stdt) = /cgdc/cl\_clrq\_process=>read\_cf\_popstart\_date( iv\_vbeln = iv\_vbeln  
                                                                      iv\_posnr = iv\_posnr ).  
*\*    SELECT SINGLE /cgdc/bcopop*  
*\*      FROM vbak*  
*\*      INTO @DATA(lv\_header\_bcpop)*  
*\*      WHERE vbeln = @iv\_vbeln.*  
    CHECK lv\_cf\_pop\_end IS NOT INITIAL. *"added 24.05.24*  
    DATA(lv\_header\_bcpop) = /cgdc/cl\_clrq\_process=>read\_bcpop( iv\_vbeln = iv\_vbeln ).  
    *"--EOC - Performance optimization*  
  
    READ TABLE gt\_contract\_itm INTO DATA(lwa\_contract\_itm)  
           WITH KEY vbeln = iv\_vbeln  
                    posnr = iv\_posnr.  
    IF sy-subrc EQ 0.  
      IF ( lwa\_contract\_itm-/cgdc/bcopop IS INITIAL OR  
        lwa\_contract\_itm-/cgdc/bcopop EQ 'N' ) AND  
        ( lv\_header\_bcpop IS INITIAL OR lv\_header\_bcpop = 'N' ) .  
        IF cv\_pop\_end\_date GT lv\_cf\_pop\_end or  
          cv\_pop\_end\_date LT lv\_cf\_pop\_stdt.  
          cv\_pop\_end\_date = lv\_cf\_pop\_end.  
          gv\_no\_new\_bill\_item = abap\_true.  
        ENDIF.  
      ENDIF.  
    ENDIF.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>SELECT\_AMOUNTS\_CALC\_GRP\_DIST*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IS\_CLRQHD                      TYPE        /CGDC/S\_CLRQHD(optional)*  
*\* | [--->] IT\_CRGRRF                      TYPE        /CGDC/TT\_CLRQGRRF(optional)*  
*\* | [--->] IS\_CRGRR                       TYPE        /CGDC/RTB\_CLGRN(optional)*  
*\* | [--->] IS\_DIST                        TYPE        /CGDC/S\_CLRQGRR(optional)*  
*\* | [--->] IS\_CRGCGF                      TYPE        /CGDC/S\_RTBPCGF(optional)*  
*\* | [--->] IS\_CRCGFC                      TYPE        /CGDC/S\_RTBPCGFC(optional)*  
*\* | [--->] I\_KSCHL                        TYPE        KSCHL(optional)*  
*\* | [<---] ET\_RESULT                      TYPE        /CGDC/TT\_CLRQ\_DIST*  
*\* | [<-->] C\_VALUE                        TYPE        KWERT(optional)*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD select\_amounts\_calc\_grp\_dist.  
    DATA: BEGIN OF ls\_quals\_key,  
            fcateg TYPE /cgdc/rtb\_qfcateg,  
            fname  TYPE /cgdc/rtb\_qfname,  
          END OF ls\_quals\_key.  
  
    DATA: lwa\_calc\_grp             TYPE /cgdc/s\_rtbpcg,  
          lwa\_calc\_group\_fun\_src   TYPE /cgdc/s\_rtbpcgfc,  
          lwa\_crgrrf               TYPE /cgdc/s\_clrqgrrf,  
          lv\_select\_clrq           TYPE string,  
*\*          lv\_select\_bs             TYPE string,*  
          lv\_select\_bs             TYPE xfeld,  
          lv\_final\_string          TYPE string,  
          lv\_pos\_string            TYPE string,  
          lv\_quals\_string          TYPE string,  
          lv\_neg\_string            TYPE string,  
          lv\_where\_clrq            TYPE string,  
          lv\_where\_for\_src\_as\_tgt  TYPE string,  
*\*          lv\_where\_bs               TYPE string,*  
          lv\_date\_where            TYPE string,  
          lv\_from\_table\_clrq       TYPE string,  
          lv\_from\_table\_bs         TYPE string,  
          ls\_quals\_key\_prev        LIKE ls\_quals\_key,  
          lt\_komv                  TYPE komv\_t,  
          lt\_komv\_bs               TYPE komv\_t,  
          lwa\_komv                 TYPE komv,  
          lwa\_komv\_bs              TYPE komv,  
          lwa\_xref\_fcat            TYPE /cgdc/\_xref\_fct,  
          lt\_source\_cond\_as\_trg    TYPE /cgdc/tt\_rtbpcgfc,  
          lwa\_source\_cond\_as\_trg   TYPE /cgdc/s\_rtbpcgf,  
          lt\_clgrp\_itm\_buffer      TYPE /cgdc/tt\_clrqit,  
          lt\_clgrp\_cnd\_buffer      TYPE /cgdc/tt\_clrqcn,  
          lwa\_cgroup\_fun\_src\_quals TYPE /cgdc/t\_rpcgfsq,  
          lwa\_clgrp\_itm\_buffer     TYPE /cgdc/s\_clrqit,  
          lwa\_clgrp\_cnd\_buffer     TYPE /cgdc/s\_clrqcn,  
          lwa\_calc\_group\_fun       TYPE /cgdc/t\_rtbpcgf,  
          lv\_dist                  TYPE string.  
  
    SELECT \* FROM /cgdc/t\_rtbpcgfd INTO TABLE @DATA(lt\_dist)  
                                    WHERE clgrn = @is\_dist-clgrn  
                                    AND dist\_base = @is\_dist-dist\_base.  
  
  
    CALL METHOD select\_realtime\_billprf\_data  
      EXPORTING  
        i\_rtbpn = is\_clrqhd-rtbpn.  
  
    DATA: lv\_date TYPE sy-datum.  
    lv\_date = sy-datum.  
    CONCATENATE `"/CGDC/\_CLRQHD" AS A `  
      ` INNER JOIN "/CGDC/\_CLRQIT" AS B `  
      ` ON  A.CLRNO = B.CLRNO AND B.CLRSIT != '000000' `  
      ` INNER JOIN "/CGDC/\_CLRQCN" AS C `  
                   ` ON  B.CLRNO = C.CLRNO `  
                   ` AND B.CLRIT = C.CLRIT `  
                   ` AND B.CLRSIT = C.CLRSIT `  
                   ` AND B.CLRDIT = C.CLRDIT `  
                   ` AND C.KINAK = ' ' `  
      ` INNER JOIN "/CGDC/\_CLRQBI" AS D `  
                   ` ON  B.CLRNO = D.CLRNO `  
                   ` AND B.RTBBINR = D.BILL\_ITEM `  
                   ` AND B.BITYP = D.BITYPE `  
                   ` AND D.FAKSP = ' '`  
                   ` AND D.BLDAT <= CURRENT\_DATE `  
  
  
  
                 INTO lv\_from\_table\_clrq.  
  
  
  
    CONCATENATE lv\_where\_clrq ` A.CLRNO = '` is\_clrqhd-clrno `'` INTO lv\_where\_clrq.  
    CONCATENATE lv\_where\_clrq ` AND ( (  B.ISTAT = 'A' ) or ( B.RTBBINR IS NOT NULL AND D.BSTAT != ' ' ) ) `  
    INTO lv\_where\_clrq .  
  
    CONCATENATE lv\_where\_for\_src\_as\_tgt ` CLRNO = '` is\_clrqhd-clrno `'` INTO lv\_where\_for\_src\_as\_tgt.  
  
    LOOP AT it\_crgrrf INTO lwa\_crgrrf  
      WHERE clgrn = is\_crgrr.  
  
      IF lwa\_crgrrf-fcatg = 'HD'.  
        IF lwa\_crgrrf-fname CS '/'.  
          CONCATENATE lv\_where\_clrq ` AND A."` lwa\_crgrrf-fname `" = '` lwa\_crgrrf-fvalue `' ` INTO lv\_where\_clrq.  
        ELSE.  
          CONCATENATE lv\_where\_clrq ` AND A.` lwa\_crgrrf-fname ` = '` lwa\_crgrrf-fvalue `' ` INTO lv\_where\_clrq.  
        ENDIF.  
  
      ELSEIF lwa\_crgrrf-fcatg = 'IT'.  
        IF lwa\_crgrrf-fname CS '/'.  
          CONCATENATE lv\_where\_clrq ` AND B."` lwa\_crgrrf-fname `" = '` lwa\_crgrrf-fvalue `' ` INTO lv\_where\_clrq.  
        ELSE.  
          CONCATENATE lv\_where\_clrq ` AND B.` lwa\_crgrrf-fname ` = '` lwa\_crgrrf-fvalue `' ` INTO lv\_where\_clrq.  
        ENDIF.  
        CONCATENATE lv\_where\_for\_src\_as\_tgt ` AND ` lwa\_crgrrf-fname ` = '` lwa\_crgrrf-fvalue `' ` INTO lv\_where\_for\_src\_as\_tgt.  
  
      ENDIF.  
      IF lwa\_crgrrf-fname = 'DATAB' OR lwa\_crgrrf-fname = 'DATBI'.  
        IF lwa\_crgrrf-fname = 'DATAB'.  
          CONCATENATE lv\_date\_where ` AND B.PRSDT >= '`  lwa\_crgrrf-fvalue `' `  INTO lv\_date\_where.  
          CONCATENATE lv\_where\_for\_src\_as\_tgt ` AND PRSDT >= '`  lwa\_crgrrf-fvalue `' `  INTO lv\_where\_for\_src\_as\_tgt.  
        ENDIF.  
        IF lwa\_crgrrf-fname = 'DATBI'.  
          CONCATENATE lv\_date\_where ` AND B.PRSDT <= '`  lwa\_crgrrf-fvalue `' `  INTO lv\_date\_where.  
          CONCATENATE lv\_where\_for\_src\_as\_tgt ` AND PRSDT <= '`  lwa\_crgrrf-fvalue `' `  INTO lv\_where\_for\_src\_as\_tgt.  
        ENDIF.  
      ENDIF.  
    ENDLOOP.  
  
    CONCATENATE lv\_where\_clrq ` ` lv\_date\_where INTO lv\_where\_clrq.  
*\*\*concatenate group by value from distribution table*  
    DELETE ADJACENT DUPLICATES FROM lt\_dist COMPARING dist\_by.  
    LOOP AT lt\_dist INTO DATA(ls\_dist).  
      CONCATENATE lv\_dist ls\_dist-dist\_by INTO lv\_dist SEPARATED BY space.  
    ENDLOOP.  
  
    CONCATENATE lv\_where\_clrq 'GROUP BY' lv\_dist INTO lv\_where\_clrq SEPARATED BY space.  
  
  
  
    LOOP AT gt\_calc\_group\_fun INTO lwa\_calc\_group\_fun WHERE clgrn = is\_crgrr  
                                                           AND proc\_kschl = i\_kschl.  
      CLEAR: lv\_select\_clrq, lv\_select\_bs, lv\_final\_string, lv\_pos\_string, lv\_neg\_string, lv\_quals\_string.  
      lv\_final\_string = 'SUM( CASE WHEN ( '.  
  
  
      LOOP AT gt\_calc\_group\_fun\_src INTO lwa\_calc\_group\_fun\_src  
             WHERE clgrn = is\_crgrr  
               AND kschl\_func = lwa\_calc\_group\_fun-kschl\_func  
               AND proc\_kschl = i\_kschl.  
  
        CLEAR: lv\_quals\_string, ls\_quals\_key, ls\_quals\_key\_prev.  
        READ TABLE gt\_calc\_group\_fun  
        INTO lwa\_source\_cond\_as\_trg  
        WITH KEY trgt\_kschl = lwa\_calc\_group\_fun\_src-kschl\_src.  
  
        IF sy-subrc IS INITIAL.  
          IF NOT ( lwa\_source\_cond\_as\_trg-clgrn = is\_crgrr  ).  
            APPEND lwa\_calc\_group\_fun\_src TO lt\_source\_cond\_as\_trg.  
          ENDIF.  
        ENDIF.  
  
        IF lwa\_calc\_group\_fun\_src-contr EQ '+'  
          OR lwa\_calc\_group\_fun\_src-contr EQ space.  
  
          IF lv\_pos\_string IS INITIAL.  
            CONCATENATE `( C.KSCHL = '` lwa\_calc\_group\_fun\_src-kschl\_src `'`  INTO lv\_pos\_string.  
          ELSE.  
            CONCATENATE lv\_pos\_string ` ) OR ( C.KSCHL = '` lwa\_calc\_group\_fun\_src-kschl\_src `'` INTO lv\_pos\_string.  
          ENDIF.  
  
  
          LOOP AT gt\_cgroup\_fun\_src\_quals INTO lwa\_cgroup\_fun\_src\_quals WHERE clgrn EQ lwa\_calc\_group\_fun\_src-clgrn  
                                                                          AND kschl\_func EQ lwa\_calc\_group\_fun\_src-kschl\_func  
                                                                          AND proc\_kschl EQ lwa\_calc\_group\_fun\_src-proc\_kschl  
                                                                          AND kschl\_src EQ lwa\_calc\_group\_fun\_src-kschl\_src  
                                                                          AND fname  IS NOT INITIAL.  
  
            MOVE-CORRESPONDING lwa\_cgroup\_fun\_src\_quals TO ls\_quals\_key.  
  
            IF ls\_quals\_key\_prev NE ls\_quals\_key .  
  
              IF lv\_quals\_string IS INITIAL.  
                IF lwa\_cgroup\_fun\_src\_quals-fname CS '/'.  
                  CONCATENATE ` AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `."` lwa\_cgroup\_fun\_src\_quals-fname `" = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                  INTO lv\_quals\_string.  
                ELSE.  
                  CONCATENATE ` AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `.` lwa\_cgroup\_fun\_src\_quals-fname ` = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                  INTO lv\_quals\_string.  
                ENDIF.  
              ELSE.  
                IF lwa\_cgroup\_fun\_src\_quals-fname CS '/'.  
                  CONCATENATE lv\_quals\_string ` ) AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `."` lwa\_cgroup\_fun\_src\_quals-fname `" = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                  INTO lv\_quals\_string.  
                ELSE.  
                  CONCATENATE lv\_quals\_string ` ) AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `.` lwa\_cgroup\_fun\_src\_quals-fname ` = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                  INTO lv\_quals\_string.  
                ENDIF.  
              ENDIF.  
              IF lwa\_cgroup\_fun\_src\_quals-fcateg EQ 'B'.  
                CONCATENATE lv\_where\_for\_src\_as\_tgt ` AND ` lwa\_cgroup\_fun\_src\_quals-fname ` = '` lwa\_cgroup\_fun\_src\_quals-fvalue `' ` INTO lv\_where\_for\_src\_as\_tgt.  
              ENDIF.  
              ls\_quals\_key\_prev = ls\_quals\_key.  
  
            ELSE.  
  
              IF lwa\_cgroup\_fun\_src\_quals-fname CS '/'.  
                CONCATENATE lv\_quals\_string ` OR ` lwa\_cgroup\_fun\_src\_quals-fcateg `."` lwa\_cgroup\_fun\_src\_quals-fname `" = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                    INTO lv\_quals\_string.  
              ELSE.  
                CONCATENATE lv\_quals\_string ` OR ` lwa\_cgroup\_fun\_src\_quals-fcateg `.` lwa\_cgroup\_fun\_src\_quals-fname ` = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                INTO lv\_quals\_string.  
              ENDIF.  
            ENDIF.  
  
          ENDLOOP.  
  
          IF lv\_quals\_string IS INITIAL.  
            *"++SOC - DCLM 5826 - Handling BSTAT*  
            IF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
              CONCATENATE lv\_quals\_string ` AND ( D.BSTAT >= '` lwa\_calc\_group\_fun\_src-bsfrm  
              `' AND D.BSTAT <= '`  lwa\_calc\_group\_fun\_src-bstto `' ` *"`' ) `*  
              INTO  lv\_quals\_string.  
              lv\_select\_bs = abap\_true.  
            ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
              CONCATENATE lv\_quals\_string ` AND ( D.BSTAT = '` lwa\_calc\_group\_fun\_src-bsfrm `' ` *" `' ) `*  
              INTO lv\_quals\_string.  
              lv\_select\_bs = abap\_true.  
            ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
              lv\_quals\_string = |{ lv\_quals\_string } AND ( D.BSTAT BETWEEN ' ' AND '{ lwa\_calc\_group\_fun\_src-bstto }' |.  
              lv\_select\_bs = abap\_true.  
            ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
              lv\_quals\_string = |{ lv\_quals\_string } AND ( D.BSTAT = ' ' |.  
              lv\_select\_bs = abap\_true.  
            ENDIF.  
  
          ELSE.  
  
            IF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
              CONCATENATE lv\_quals\_string ` ) AND ( D.BSTAT >= '` lwa\_calc\_group\_fun\_src-bsfrm  
              `' AND D.BSTAT <= '`  lwa\_calc\_group\_fun\_src-bstto `' ` *"`' ) `*  
              INTO  lv\_quals\_string.  
              lv\_select\_bs = abap\_true.  
            ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
              CONCATENATE lv\_quals\_string ` ) AND ( D.BSTAT = '` lwa\_calc\_group\_fun\_src-bsfrm `' ` *"`' ) `*  
              INTO lv\_quals\_string.  
              lv\_select\_bs = abap\_true.  
            ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
              lv\_quals\_string = |{ lv\_quals\_string } AND ( D.BSTAT BETWEEN ' ' AND '{ lwa\_calc\_group\_fun\_src-bstto }'|.  
              lv\_select\_bs = abap\_true.  
            ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
              lv\_quals\_string = |{ lv\_quals\_string } AND ( D.BSTAT = ' '|.  
              lv\_select\_bs = abap\_true.  
            ENDIF.  
            *"++EOC - DCLM 5826 - Handling BSTAT*  
          ENDIF.  
  
          IF NOT lv\_quals\_string IS INITIAL.  
            CONCATENATE lv\_pos\_string lv\_quals\_string ` ) ` INTO lv\_pos\_string SEPARATED BY space.  
          ENDIF.  
  
        ELSEIF lwa\_calc\_group\_fun\_src-contr EQ '-'.  
  
          IF lv\_neg\_string IS INITIAL.  
            CONCATENATE `( C.KSCHL = '` lwa\_calc\_group\_fun\_src-kschl\_src `'`  INTO lv\_neg\_string.  
          ELSE.  
            CONCATENATE lv\_neg\_string ` ) OR ( C.KSCHL = '` lwa\_calc\_group\_fun\_src-kschl\_src `'` INTO lv\_neg\_string.  
          ENDIF.  
  
          LOOP AT gt\_cgroup\_fun\_src\_quals INTO lwa\_cgroup\_fun\_src\_quals WHERE clgrn EQ lwa\_calc\_group\_fun\_src-clgrn  
                                                                                    AND kschl\_func EQ lwa\_calc\_group\_fun\_src-kschl\_func  
                                                                                    AND proc\_kschl EQ lwa\_calc\_group\_fun\_src-proc\_kschl  
                                                                                    AND kschl\_src EQ lwa\_calc\_group\_fun\_src-kschl\_src.  
  
            MOVE-CORRESPONDING lwa\_cgroup\_fun\_src\_quals TO ls\_quals\_key.  
  
            IF ls\_quals\_key\_prev NE ls\_quals\_key.  
  
              IF lv\_quals\_string IS INITIAL.  
                IF lwa\_cgroup\_fun\_src\_quals-fname CS '/'.  
                  CONCATENATE ` AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `."` lwa\_cgroup\_fun\_src\_quals-fname `" = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                  INTO lv\_quals\_string.  
                ELSE.  
                  CONCATENATE ` AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `.` lwa\_cgroup\_fun\_src\_quals-fname ` = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                  INTO lv\_quals\_string.  
                ENDIF.  
              ELSE.  
                IF lwa\_cgroup\_fun\_src\_quals-fname CS '/'.  
                  CONCATENATE lv\_quals\_string ` ) AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `."` lwa\_cgroup\_fun\_src\_quals-fname `" = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                       INTO lv\_quals\_string.  
                ELSE.  
                  CONCATENATE lv\_quals\_string ` ) AND ( ` lwa\_cgroup\_fun\_src\_quals-fcateg `.` lwa\_cgroup\_fun\_src\_quals-fname ` = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                  INTO lv\_quals\_string.  
                ENDIF.  
  
              ENDIF.  
  
            ELSE.  
              IF lwa\_cgroup\_fun\_src\_quals-fname CS '/'.  
                CONCATENATE lv\_quals\_string ` OR ` lwa\_cgroup\_fun\_src\_quals-fcateg `."` lwa\_cgroup\_fun\_src\_quals-fname `" = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                     INTO lv\_quals\_string.  
              ELSE.  
                CONCATENATE lv\_quals\_string ` OR ` lwa\_cgroup\_fun\_src\_quals-fcateg `.` lwa\_cgroup\_fun\_src\_quals-fname ` = '` lwa\_cgroup\_fun\_src\_quals-fvalue `'`  
                INTO lv\_quals\_string.  
              ENDIF.  
  
            ENDIF.  
  
          ENDLOOP.  
  
          IF lv\_quals\_string IS INITIAL.  
  
            IF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
              CONCATENATE lv\_quals\_string ` AND ( D.BSTAT >= '` lwa\_calc\_group\_fun\_src-bsfrm  
              `' AND D.BSTAT <= '`  lwa\_calc\_group\_fun\_src-bstto *"`' ) `*  
              INTO  lv\_quals\_string.  
              lv\_select\_bs = abap\_true.  
            ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
              CONCATENATE lv\_quals\_string ` AND ( D.BSTAT = '` lwa\_calc\_group\_fun\_src-bsfrm *"`' ) `*  
              INTO lv\_quals\_string.  
              lv\_select\_bs = abap\_true.  
              *"++SOC - DCLM 5826 - Handling BSTAT*  
            ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
              lv\_quals\_string = |{ lv\_quals\_string } AND ( D.BSTAT BETWEEN ' ' AND '{ lwa\_calc\_group\_fun\_src-bstto }'|.  
              lv\_select\_bs = abap\_true.  
            ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
              lv\_quals\_string = |{ lv\_quals\_string } AND ( D.BSTAT = ' '|.  
              lv\_select\_bs = abap\_true.  
            ENDIF.  
            *"++EOC - DCLM 5826 - Handling BSTAT*  
          ELSE.  
  
            IF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
              CONCATENATE lv\_quals\_string ` ) AND ( D.BSTAT >= '` lwa\_calc\_group\_fun\_src-bsfrm  
              `' AND D.BSTAT <= '`  lwa\_calc\_group\_fun\_src-bstto `' ` *"`' ) `*  
              INTO  lv\_quals\_string.  
              lv\_select\_bs = abap\_true.  
            ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS NOT INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
              CONCATENATE lv\_quals\_string ` ) AND ( D.BSTAT = '` lwa\_calc\_group\_fun\_src-bsfrm `' ` *"`' ) `*  
              INTO lv\_quals\_string.  
              lv\_select\_bs = abap\_true.  
              *"++SOC - DCLM 5826 - Handling BSTAT*  
            ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS NOT INITIAL.  
              lv\_quals\_string = |{ lv\_quals\_string } AND ( D.BSTAT BETWEEN ' ' AND '{ lwa\_calc\_group\_fun\_src-bstto }'|.  
              lv\_select\_bs = abap\_true.  
            ELSEIF lwa\_calc\_group\_fun\_src-bsfrm IS INITIAL AND lwa\_calc\_group\_fun\_src-bstto IS INITIAL.  
              lv\_quals\_string = |{ lv\_quals\_string } AND ( D.BSTAT = ' '|.  
              lv\_select\_bs = abap\_true.  
            ENDIF.  
            *"++EOC - DCLM 5826 - Handling BSTAT*  
          ENDIF.  
  
          IF NOT lv\_quals\_string IS INITIAL.  
            CONCATENATE lv\_neg\_string lv\_quals\_string ` ) ` INTO lv\_neg\_string SEPARATED BY space.  
          ENDIF.  
  
        ENDIF.  
  
  
      ENDLOOP.  
      IF lv\_pos\_string IS INITIAL AND lv\_neg\_string IS INITIAL..  
        EXIT.  
      ENDIF.  
*\*\*\*New changes*  
      IF lv\_pos\_string IS NOT INITIAL AND lv\_neg\_string IS NOT INITIAL.  
        CONCATENATE lv\_final\_string ` ` lv\_pos\_string ` ) ) THEN KWERT WHEN ( ` lv\_neg\_string ` ) ) THEN -1 \* KWERT ` INTO lv\_final\_string.  
      ELSEIF lv\_pos\_string IS NOT INITIAL AND lv\_neg\_string IS INITIAL.  
        CONCATENATE lv\_final\_string ` ` lv\_pos\_string ` ) ) THEN KWERT ` INTO lv\_final\_string.  
      ELSE.  
        CONCATENATE lv\_final\_string ` ` lv\_neg\_string ` ) ) THEN -1 \* KWERT ` INTO lv\_final\_string.  
      ENDIF.  
  
      CONCATENATE lv\_final\_string `ELSE 0 END ) AS KWERT ` INTO lv\_final\_string.  
  
  
      IF lv\_final\_string IS NOT INITIAL  
  
        AND lv\_from\_table\_clrq IS NOT INITIAL  
        AND lv\_where\_clrq IS NOT INITIAL.  
  
  
  
        IF NOT cl\_abap\_dbfeatures=>use\_features(  
                  EXPORTING  
                    requested\_features =  
                      VALUE #( ( cl\_abap\_dbfeatures=>call\_amdp\_method )  
                               ( cl\_abap\_dbfeatures=>amdp\_table\_function ) ) ).  
          *"Error handling..*  
        ELSE.  
          TRY.  
              NEW /cgdc/cl\_clrq\_process\_amdp( )->get\_clrqsrc\_cond\_dist(  
                EXPORTING  
                  iv\_client          = sy-mandt  
                  iv\_select          = lv\_final\_string  
                  iv\_from\_table      = lv\_from\_table\_clrq  
                  iv\_where\_condition = lv\_where\_clrq  
                IMPORTING  
                  et\_result          = et\_result  
              ).  
              lt\_komv = CORRESPONDING #( et\_result ).  
            CATCH cx\_amdp\_error INTO DATA(amdp\_error).  
              *"error handling*  
          ENDTRY.  
  
        ENDIF.  
  
        READ TABLE lt\_komv INTO lwa\_komv INDEX 1.  
        c\_value = lwa\_komv-kwert.  
      ENDIF.  
  
  
      IF lt\_source\_cond\_as\_trg IS NOT INITIAL.  
        LOOP AT lt\_source\_cond\_as\_trg INTO lwa\_calc\_group\_fun\_src .  
          CALL FUNCTION '/CGDC/CLRQ\_DATA\_GET'  
            IMPORTING  
              et\_clgrp\_itm\_buffer = lt\_clgrp\_itm\_buffer  
              et\_clgrp\_cnd\_buffer = lt\_clgrp\_cnd\_buffer.  
          lv\_where\_for\_src\_as\_tgt = cl\_abap\_dyn\_prg=>escape\_quotes\_str( lv\_where\_for\_src\_as\_tgt ). *"ATC Correction*  
          LOOP AT lt\_clgrp\_itm\_buffer INTO lwa\_clgrp\_itm\_buffer WHERE (lv\_where\_for\_src\_as\_tgt).  
            LOOP AT lt\_clgrp\_cnd\_buffer INTO lwa\_clgrp\_cnd\_buffer WHERE clrit = lwa\_clgrp\_itm\_buffer-clrit AND kschl = lwa\_calc\_group\_fun\_src-kschl\_src.  
              IF lwa\_calc\_group\_fun\_src-contr = '-'.  
                c\_value = c\_value - lwa\_clgrp\_cnd\_buffer-kwert.  
              ELSE.  
                c\_value = c\_value + lwa\_clgrp\_cnd\_buffer-kwert.  
              ENDIF.  
            ENDLOOP.  
          ENDLOOP.  
  
        ENDLOOP.  
  
      ENDIF.  
  
      LOOP AT et\_result INTO DATA(ls\_result).  
        c\_value = c\_value + ls\_result-kwert.  
      ENDLOOP.  
  
      CLEAR:lwa\_komv,  
            lt\_komv.  
    ENDLOOP.  
  
*\*    ENDIF.*  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>UPDATE\_MLTIPM\_CLRQ\_DATA*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IS\_MLPAY\_DOC                   TYPE        /CGDC/S\_MLPAY\_DOC(optional)*  
*\* | [--->] IS\_MLPAYSA\_DOC                 TYPE        /CGDC/S\_MLPAYSA\_DOC(optional)*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD update\_mltipm\_clrq\_data.  
    DATA: lwa\_mlpay1 TYPE /cgdc/s\_mlpay.  
    DATA : lt\_pspnr\_r  TYPE RANGE OF ps\_psp\_pnr,  
           lwa\_pspnr\_r LIKE LINE OF lt\_pspnr\_r.  
    DATA : lt\_vbeln\_r  TYPE RANGE OF vbeln\_va,  
           lwa\_vbeln\_r LIKE LINE OF lt\_vbeln\_r.  
    DATA : ls\_clrq\_doc TYPE /cgdc/s\_clrq\_doc,  
           lt\_clrq\_doc TYPE TABLE OF /cgdc/s\_clrq\_doc.  
    DATA: lt\_clrqpit TYPE TABLE OF /cgdc/s\_clrqpit.  
    DATA lt\_crhdr TYPE TABLE OF /cgdc/\_clrqhd.  
    DATA lwa\_crhdr TYPE  /cgdc/\_clrqhd.  
    CONSTANTS : lc\_updkz\_update TYPE updkz VALUE 'U'.  
    CONSTANTS : lc\_updkz\_insert TYPE updkz VALUE 'I'.  
    CONSTANTS : lc\_wbs\_removed\_error TYPE c VALUE 'W'.  
  
    lwa\_pspnr\_r-sign = 'I'.  
    lwa\_pspnr\_r-option = 'EQ'.  
    lwa\_vbeln\_r-sign = 'I'.  
    lwa\_vbeln\_r-option = 'EQ'.  
    LOOP AT is\_mlpay\_doc-x-mlpay INTO lwa\_mlpay1 WHERE updkz IS NOT INITIAL.  
      READ TABLE is\_mlpay\_doc-y-mlpay INTO DATA(lwa\_mlpay2) WITH KEY mlpayid = lwa\_mlpay1-mlpayid.  
      IF sy-subrc = 0.  
        IF lwa\_mlpay1-srcwbs\_del <> lwa\_mlpay2-srcwbs\_del OR lwa\_mlpay1-tarwbs\_del <> lwa\_mlpay2-tarwbs\_del.  
          lwa\_vbeln\_r-low = lwa\_mlpay1-ocont.  
          APPEND lwa\_vbeln\_r TO lt\_vbeln\_r.  
        ELSEIF lwa\_mlpay1-totfund <> lwa\_mlpay2-totfund OR lwa\_mlpay1-tagfund <> lwa\_mlpay2-tagfund.  
          lwa\_pspnr\_r-low = lwa\_mlpay1-tarwbs.  
          APPEND lwa\_pspnr\_r TO lt\_pspnr\_r.  
        ENDIF.  
      ELSE.  
        lwa\_vbeln\_r-low = lwa\_mlpay1-ocont.  
        APPEND lwa\_vbeln\_r TO lt\_vbeln\_r.  
      ENDIF.  
    ENDLOOP.  
  
    LOOP AT is\_mlpaysa\_doc-x-mlpay INTO lwa\_mlpay1 WHERE updkz IS NOT INITIAL.  
      READ TABLE is\_mlpaysa\_doc-y-mlpay INTO lwa\_mlpay2 WITH KEY mlpayid = lwa\_mlpay1-mlpayid.  
      IF sy-subrc = 0.  
        IF lwa\_mlpay1-srcwbs\_del <> lwa\_mlpay2-srcwbs\_del OR lwa\_mlpay1-tarwbs\_del <> lwa\_mlpay2-tarwbs\_del.  
          lwa\_vbeln\_r-low = lwa\_mlpay1-ocont.  
          APPEND lwa\_vbeln\_r TO lt\_vbeln\_r.  
        ENDIF.  
        IF lwa\_mlpay1-totfund <> lwa\_mlpay2-totfund OR lwa\_mlpay1-tagfund <> lwa\_mlpay2-tagfund OR lwa\_mlpay1-allocation <> lwa\_mlpay2-allocation.  
          lwa\_pspnr\_r-low = lwa\_mlpay1-tarwbs.  
          APPEND lwa\_pspnr\_r TO lt\_pspnr\_r.  
        ENDIF.  
      ELSE.  
        lwa\_vbeln\_r-low = lwa\_mlpay1-ocont.  
        APPEND lwa\_vbeln\_r TO lt\_vbeln\_r.  
      ENDIF.  
    ENDLOOP.  
  
    IF lt\_pspnr\_r IS NOT INITIAL.  
      SELECT clrno clrit clrsit  
        FROM /cgdc/\_clrqic  
        INTO CORRESPONDING FIELDS OF TABLE lt\_clrqpit  
        WHERE ps\_psp\_pnr IN lt\_pspnr\_r.  
      IF sy-subrc = 0.  
        SORT lt\_clrqpit BY clrno clrit clrsit.  
        DELETE lt\_clrqpit WHERE clrsit = '000000'.  
        DELETE ADJACENT DUPLICATES FROM lt\_clrqpit COMPARING clrno clrit clrsit.  
  
        LOOP AT lt\_clrqpit ASSIGNING FIELD-SYMBOL(<lwa\_clrqpit>).  
          <lwa\_clrqpit>-wbs\_alloc\_chng = 'X'.  
          <lwa\_clrqpit>-updkz = lc\_updkz\_insert .  
        ENDLOOP.  
  
        CALL FUNCTION '/CGDC/CLRQ\_CLRQPRS\_UPDATE\_DB' IN UPDATE TASK  
          EXPORTING  
            it\_clrqpit = lt\_clrqpit  
*\*           IT\_CLRQPCN =*  
          .  
      ENDIF.  
    ENDIF.  
    IF lt\_vbeln\_r IS NOT INITIAL.  
      SELECT \*  
        FROM /cgdc/\_clrqhd  
        INTO CORRESPONDING FIELDS OF TABLE lt\_crhdr  
        WHERE vgbel IN lt\_vbeln\_r.  
      IF sy-subrc = 0.  
        LOOP AT lt\_crhdr INTO lwa\_crhdr.  
          MOVE-CORRESPONDING lwa\_crhdr TO ls\_clrq\_doc-x-crhdr.  
          ls\_clrq\_doc-x-crhdr-msgcode = lc\_wbs\_removed\_error.  
          ls\_clrq\_doc-x-crhdr-updkz = lc\_updkz\_update.  
          APPEND ls\_clrq\_doc TO lt\_clrq\_doc.  
  
        ENDLOOP.  
        CALL FUNCTION '/CGDC/CLRQ\_SAVE'  
          EXPORTING  
            i\_set\_update\_task = 'X'  
*\*           I\_COMMIT\_WORK     = 'X'*  
          CHANGING  
            ct\_clrq\_doc       = lt\_clrq\_doc  
          EXCEPTIONS  
            no\_change         = 1  
            OTHERS            = 2.  
        IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
        ENDIF.  
  
      ENDIF.  
    ENDIF.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>COPY\_CLGRP\_COND*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IS\_KOMP                        TYPE        KOMP*  
*\* | [--->] IV\_KOMV\_KSCHL                  TYPE        KSCHL*  
*\* | [--->] IS\_KOMK                        TYPE        KOMK*  
*\* | [<-->] CV\_KWERT                       TYPE        KWERT*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD copy\_clgrp\_cond.  
    DATA : lt\_komv  TYPE komv\_tab,  
           lwa\_komv TYPE komv.  
    *"FM Call*  
    CALL FUNCTION '/CGDC/CLRQ\_DATA\_GET'  
      IMPORTING  
*\*       ES\_CLRHD      =*  
*\*       ES\_CLRIT      =*  
*\*       ES\_CRGRR      =*  
*\*       ET\_CRGRRF     =*  
*\*       ET\_CLGRP\_ITM\_BUFFER       =*  
*\*       ET\_CLGRP\_CND\_BUFFER       =*  
*\*       ET\_PBD\_CONDS  =*  
*\*       ET\_CLRQIT\_GRP\_CONDS       =*  
        et\_crgrr\_komv = lt\_komv.  
  
    READ TABLE lt\_komv INTO lwa\_komv WITH KEY  kposn = is\_komp-kposn kschl = iv\_komv\_kschl. *"KNUMV is logic missing*  
    IF sy-subrc = 0.  
      cv\_kwert = lwa\_komv-kwert.  
    ENDIF.  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>GET\_FIELDINFO*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] I\_TABNAME                      TYPE        TABNAME*  
*\* | [<---] ET\_DFIES                       TYPE        DFIES\_TAB*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD get\_fieldinfo.  
  
    CLEAR: et\_dfies[].  
  
    CALL FUNCTION 'DDIF\_FIELDINFO\_GET'  
      EXPORTING  
        tabname        = i\_tabname  
        langu          = sy-langu  
        all\_types      = abap\_true  
      TABLES  
        dfies\_tab      = et\_dfies  
      EXCEPTIONS  
        not\_found      = 1  
        internal\_error = 2  
        OTHERS         = 3.  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>GET\_CR\_DOCTYPE*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IS\_VBFA                        TYPE        DOCUMENT\_FLOW\_ALV\_STRUC*  
*\* | [<-()] RV\_DOCTYPE                     TYPE        FMRE\_BATXT*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD get\_cr\_doctype.  
  
    IF gt\_cr\_doctype IS INITIAL.  
      *"get all descr. in user language*  
      SELECT \* FROM /cgdc/t\_clrtypet  
               WHERE spras = @sy-langu  
               INTO TABLE @gt\_cr\_doctype.  
      IF sy-subrc <> 0.  
        CLEAR: gt\_cr\_doctype.  
      ENDIF.  
    ENDIF.  
  
    *"return requested descr.*  
    rv\_doctype = VALUE #( gt\_cr\_doctype[ clrqt = is\_vbfa-vbtyp\_n ]-descr OPTIONAL ).  
  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>IS\_RTBP\_ON\_CONTR*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IV\_CLRNO                       TYPE        /CGDC/CLRQNO*  
*\* | [<-()] RV\_RESULT                      TYPE        ABAP\_BOOL*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD is\_rtbp\_on\_contr.  
    SELECT SINGLE b~/cgdc/rtbpn FROM /cgdc/\_clrqhd AS a  
                           INNER JOIN vbak         AS b  
                           ON ( a~vgbel = b~vbeln )  
                           WHERE a~clrno = @iv\_clrno  
                           INTO @DATA(lv\_rtbpn).  
    IF sy-subrc = 0 AND lv\_rtbpn IS NOT INITIAL.  
      rv\_result = abap\_true.  
    ENDIF.  
  ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>READ\_CF\_POPSTART\_DATE*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IV\_VBELN                       TYPE        VBELN\_VA*  
*\* | [--->] IV\_POSNR                       TYPE        POSNR\_VA*  
*\* | [<-()] RV\_FENDAT                      TYPE        /CGDC/FENDAT*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  method READ\_CF\_POPSTART\_DATE.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: Read CF popstart date for Contract item*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
  
    DATA(ls\_pop\_stdt) = VALUE #( gt\_cf\_pop\_endat[ vbeln = iv\_vbeln posnr = iv\_posnr ] OPTIONAL ).  
  
    IF ls\_pop\_stdt IS INITIAL.  
      SELECT SINGLE vbeln, posnr,/cgdc/fendat,/cgdc/fstdat  
          FROM /cgdc/cf\_sd\_it  
          INTO @ls\_pop\_stdt  
          WHERE vbeln = @iv\_vbeln  
            AND posnr = @iv\_posnr.  
      IF sy-subrc = 0.  
        rv\_fendat = ls\_pop\_stdt-FSTDAT.  
        APPEND ls\_pop\_stdt TO gt\_cf\_pop\_endat.  
      ENDIF.  
    ELSE.  
      rv\_fendat = ls\_pop\_stdt-FSTDAT.  
    ENDIF.  
  endmethod.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method /CGDC/CL\_CLRQ\_PROCESS=>VALIDATE\_CLRQ\_TRGR*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
  METHOD validate\_clrq\_trgr.  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* Cognitus Technologies Confidential*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\**  
*\*  [2019] - [2025] Cognitus Technologies LLC*  
*\*  All Rights Reserved.*  
*\**  
*\* NOTICE:  All information contained herein is, and remains*  
*\* the property of Cognitus Technologies LLC and its suppliers,*  
*\* if any. The intellectual and technical concepts contained*  
*\* herein are proprietary to Cognitus Technologies LLC and its*  
*\* suppliers and may be covered by U.S. and Foreign Patents,*  
*\* patents in process, and are protected by trade secret or*  
*\* copyright law. Dissemination of this information or*  
*\* reproduction of this material is strictly forbidden unless*  
*\* prior written permission is obtained from Cognitus*  
*\* Consulting LLC.*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\* VERSION CONTROL (Most recent on top):*  
*\**  
*\* RELEASE: SAPK-231COINCGDC (CIS-AD S/4 Hana 2023)*  
*\**  
*\* DESCRIPTION: VALIDATE\_CLRQ\_TRGR*  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*    DATA : lt\_table\_data  TYPE TABLE OF /cgdc/s\_clrq\_buffer\_tables,*  
*\*           lwa\_table\_data TYPE /cgdc/s\_clrq\_buffer\_tables,*  
*\*           lo\_data        TYPE REF TO data.*  
*\*    FIELD-SYMBOLS : <lt\_table> TYPE STANDARD TABLE.*  
*\**  
*\*    IF /cgdc/cl\_clrq\_set\_buffer=>gt\_acdoca IS NOT INITIAL.*  
*\*      lwa\_table\_data-tabname = 'ACDOCA'.*  
*\*      CREATE DATA lo\_data TYPE TABLE OF acdoca.*  
*\*      ASSIGN lo\_data->\* TO <lt\_table>.*  
*\*      <lt\_table> = /cgdc/cl\_clrq\_set\_buffer=>gt\_acdoca[].*  
*\*    ENDIF.*  
*\**  
*\*    IF /cgdc/cl\_clrq\_set\_buffer=>gt\_coep IS NOT INITIAL.*  
*\*       lwa\_table\_data-tabname = 'COEP'.*  
*\*      CREATE DATA lo\_data TYPE TABLE OF coep.*  
*\*      ASSIGN lo\_data->\* TO <lt\_table>.*  
*\*      <lt\_table> = /cgdc/cl\_clrq\_set\_buffer=>gt\_coep[].*  
*\*    ENDIF.*  
*\**  
*\*    IF /cgdc/cl\_clrq\_set\_buffer=>gt\_cobk IS NOT INITIAL.*  
*\*       lwa\_table\_data-tabname = 'COBK'.*  
*\*      CREATE DATA lo\_data TYPE TABLE OF cobk.*  
*\*      ASSIGN lo\_data->\* TO <lt\_table>.*  
*\*      <lt\_table> = /cgdc/cl\_clrq\_set\_buffer=>gt\_cobk[].*  
*\*    ENDIF.*  
*\**  
*\*    IF /cgdc/cl\_clrq\_set\_buffer=>gt\_coepr IS NOT INITIAL.*  
*\*       lwa\_table\_data-tabname = 'COEPR'.*  
*\*      CREATE DATA lo\_data TYPE TABLE OF coepr.*  
*\*      ASSIGN lo\_data->\* TO <lt\_table>.*  
*\*      <lt\_table> = /cgdc/cl\_clrq\_set\_buffer=>gt\_coepr[].*  
*\*    ENDIF.*  
  
*\*    CALL FUNCTION '/CGDC/CLRQ\_TRGR\_REAL\_TIME\_PRCS' IN UPDATE TASK*  
*\*      EXPORTING*  
*\*        it\_data        = lt\_table\_data*  
*\*        iv\_bukrs       = '9880'*  
  
  .  
      CALL FUNCTION '/CGDC/CLRQ\_TRGR\_REAL\_TIME\_PRCS' IN UPDATE TASK  
        EXPORTING  
          iv\_bukrs  = '9080'  
          it\_acdoca = /cgdc/cl\_clrq\_set\_buffer=>gt\_acdoca  
          it\_coep   = /cgdc/cl\_clrq\_set\_buffer=>gt\_coep  
          it\_coepr  = /cgdc/cl\_clrq\_set\_buffer=>gt\_coepr  
          it\_cobk   = /cgdc/cl\_clrq\_set\_buffer=>gt\_cobk  
          it\_mseg   = /cgdc/cl\_clrq\_set\_buffer=>gt\_mseg.  
  
  ENDMETHOD.  
ENDCLASS.