ABAP RAP: CDS with Custom Entity + Service Publication   
Check Item number 7 first, so you can have an idea about the App Layout and functionalities

1. CDS Views

|  |
| --- |
| @EndUserText.label: 'Überfällige Forderungen'  @ObjectModel.query.implementedBy: 'ABAP:ZCL\_GET\_DUNNINGS'  @Search.searchable: true  @UI: {  headerInfo: {  typeName: 'Überfällige Forderungen',  typeNamePlural: 'Überfällige Forderungen',  title: {  type: #STANDARD,  label: 'Überfällige Forderungen'  },  description: {  value: 'belnr'  }  }  }  **define** **root** **custom** **entity** ZI\_DUNN\_ABAP\_CALL\_2  **{**  **key** gjahr **:** *gjahr***;**  @UI.facet : [  /\* { // Object Page Header Details  id : 'ftype\_desc',  purpose : #HEADER,  position : 20,  type : #DATAPOINT\_REFERENCE,  targetQualifier: 'ftype\_desc'  },  { // Object Page Header Details  id : 'name1',  purpose : #HEADER,  position : 30,  type : #DATAPOINT\_REFERENCE,  targetQualifier: 'name1'  },  { // Object Page Header Details  id : 'budat',  purpose : #HEADER,  position : 40,  type : #DATAPOINT\_REFERENCE,  targetQualifier: 'budat'  },  { // Object Page Header Details  id : 'faedt',  purpose : #HEADER,  position : 50,  type : #DATAPOINT\_REFERENCE,  targetQualifier: 'faedt'  },  { // Object Page Header Details  id : 'utage',  purpose : #HEADER,  position : 60,  type : #DATAPOINT\_REFERENCE,  targetQualifier: 'utage'  },  { // Object Page Header Details  id : 'wrbtr',  purpose : #HEADER,  position : 70,  type : #DATAPOINT\_REFERENCE,  targetQualifier: 'wrbtr'  },\*/  { // Link the custom entity Header with the Positions- To be able to see it at Preview  id : 'PositionsDetails',  purpose : #STANDARD,  position : 10,  //label : 'PositionDetails',  type : #LINEITEM\_REFERENCE,  targetElement : '\_Positions'  },  {  targetQualifier: 'headerData',  type : #FIELDGROUP\_REFERENCE,  purpose : #HEADER  }  // { // Link the custom entity Header with PDF- To be able to see it at Preview  // id : 'PdfDetails',  // purpose : #STANDARD,  // position : 20,  // label : 'Beleg',  // type : #IDENTIFICATION\_REFERENCE,  // targetQualifier: 'Pdf'  // }  ]  @UI.lineItem : [{position: 10, cssDefault.width: '8rem'}]  // @UI.identification: [{ position: 10, type: #FOR\_ACTION, importance: #HIGH, dataAction: 'inkasso', label: 'Inkasso/Mahnen', emphasized: true }]  @UI.dataPoint : { qualifier: 'belnr'}  @Search.defaultSearchElement: true  **key** belnr **:** *belnr\_d***;**  //@UI.identification: [{ position: 20, type: #FOR\_ACTION, importance: #HIGH, dataAction: 'verlaengern', label: 'Verlängern' }]  **key** bukrs **:** *bukrs***;**  //@UI.identification: [{ position: 30, type: #FOR\_ACTION, importance: #HIGH, dataAction: 'ausbuchen', label: 'Ausbuchen' }]  **key** BUZEI **:** *buzei***;**  @UI.hidden: true  ftype **:** *zfi\_e\_nb\_ftype***;**  @UI.lineItem : [{position: 20, cssDefault.width: '10rem', emphasized: true}] --emphasized not working  @UI.dataPoint : { qualifier: 'ftype\_desc'}  @UI.fieldGroup : [{type: #STANDARD, qualifier: 'headerData', position: 10}]  ftype\_desc **:** *zfi\_e\_nb\_ftype\_desc***;**  @UI.hidden: true  ERFASSER **:** *usnam***;**  @UI.hidden: true  KUNNR **:** *kunnr***;**  @UI.lineItem : [{position: 30, cssDefault.width: '12rem'}]  @UI.dataPoint : { qualifier: 'name1'}  @UI.fieldGroup : [{type: #STANDARD, qualifier: 'headerData', position: 20}]  @Search :{ defaultSearchElement: true, fuzzinessThreshold: 0.7}  NAME1 **:** *z\_fi\_name1***;**  @UI.lineItem : [{position: 40, cssDefault.width: '12rem'}]  @UI.dataPoint : { qualifier: 'budat'}  @UI.fieldGroup : [{type: #STANDARD, qualifier: 'headerData', position: 30}]  BUDAT **:** *z\_fi\_budat***;**  @UI.lineItem : [{position: 50, cssDefault.width: '12rem'}]  @UI.dataPoint : { qualifier: 'faedt'}  @UI.fieldGroup : [{type: #STANDARD, qualifier: 'headerData', position: 40}]  FAEDT **:** *z\_fi\_dats***;**  @UI.lineItem : [{position: 60, cssDefault.width: '10rem'}]  @UI.dataPoint : { qualifier: 'utage'}  @UI.fieldGroup : [{type: #STANDARD, qualifier: 'headerData', position: 50}]  UTAGE **:** *zfi\_e\_nb\_utage2***;**  @Semantics.currencyCode: true  WAERS **:** *waers***;**  @Semantics.amount.currencyCode: 'WAERS'  @UI.lineItem : [{position: 70, cssDefault.width: '19rem'}]  @UI.dataPoint : { qualifier: 'wrbtr'}  @UI.fieldGroup : [{type: #STANDARD, qualifier: 'headerData', position: 60}]  WRBTR **:** *z\_fi\_wrbtr***;**  //@UI.identification: [{ qualifier: 'Pdf' }]  ZFBDT **:** *dzfbdt***;**  @UI.hidden:true  PROJK **:** *abap***.***char***(** 8 **);** //ps\_psp\_pnr;  @UI.hidden: true  PROJK\_NAME **:** *ps\_post1***;**  @UI.hidden: true  BETR\_BOX **:** *char1***;**  @UI.hidden: true  BETREIBUNG\_AKTIV **:** *char1***;**  @UI.hidden: true  MAHN\_BOX **:** *char1***;**  @UI.hidden: true  VERL\_BOX **:** *char1***;**  @UI.hidden: true  VERL\_DATUM **:** *dats***;**  @UI.hidden:true  AUSB\_BOX **:** *char1***;**  //@UI.identification: [{ qualifier: 'Pdf' }]  @UI.hidden: true  pdf **:** *z\_fi\_pdf***;**  BEMERKUNG **:** *char50***;**  @UI.hidden:true  pdf\_url **:** *z\_fi\_pdf\_url***;**  \_Positions **:** **composition** **[**1**..\*]** **of** ZFI\_I\_POSITION\_DETAILS\_2**;**  \_Actions **:** **composition** **[**1**..**1**]** **of** ZFI\_DUNN\_ACTION\_2**;**  **}** |

* 1. Class ZCL\_GET\_DUNNINGS

|  |
| --- |
| CLASS zcl\_get\_dunnings DEFINITION  PUBLIC  FINAL  CREATE PUBLIC .  PUBLIC SECTION.  INTERFACES if\_rap\_query\_provider .  PROTECTED SECTION.  PRIVATE SECTION.  METHODS get\_pdf  IMPORTING  !iv\_belnr TYPE belnr\_d  !iv\_gjahr TYPE gjahr  EXPORTING  !ev\_pdf TYPE z\_fi\_pdf  !ev\_pdf\_url TYPE z\_fi\_pdf\_url.  ENDCLASS.  CLASS ZCL\_GET\_DUNNINGS IMPLEMENTATION.  METHOD if\_rap\_query\_provider~select.  \*----------------------------------------------------------------------\*  \*Methodenname: ZCL\_GET\_DUNNINGS \*  \*Entwickler: CCSAP ETH / NOTTERR \*  \*Beschreibung: Aufruf aus Custom CDS holt sich Forderungen aud SAPGui \*  \*Auftraggeber: D. Brandenberger, CCSAP \*  \*Programmspezifikation: keine \*  \*Verantwortliche Stelle ETHZ: CCSAP, R.Notter \*  \*----------------------------------------------------------------------\*  \*SAP-Release ERP2005 / 700 \*  \*Erstellungsdatum: 17.08.2022 \*  \*----------------------------------------------------------------------\*  \*Änderungs-Log \*  \* >001< 20220916 notterr User-Mapping um Forderungen eines and. \*  \* Benutzer zu holen \*  \*----------------------------------------------------------------------\*  DATA: lv\_param TYPE char8,  lv\_bname TYPE sy-uname,  gt\_dunnings\_tmp TYPE zfi\_s\_nb\_ford\_inbox\_ethis\_2, "zfi\_t\_nb\_ford\_inbox\_ethis,  lt\_result TYPE STANDARD TABLE OF zfi\_s\_nb\_ford\_inbox\_ethis\_2, "zfi\_t\_nb\_ford\_inbox\_ethis,  ls\_result TYPE zfi\_s\_nb\_ford\_inbox\_ethis\_2, "zfi\_t\_nb\_ford\_inbox\_ethis,  lt\_sort\_elements TYPE if\_rap\_query\_request=>tt\_sort\_elements.  TRY.  TRY.  TRY.  io\_request->is\_data\_requested( ).  DATA(lveid) = io\_request->get\_entity\_id( ).  DATA(sexpr) = io\_request->get\_search\_expression( ).  " IF io\_request->is\_data\_requested( ).  io\_request->get\_paging( ).  DATA(lt\_filter\_cond) = io\_request->get\_parameters( ).  DATA(lt\_filter\_cond2) = io\_request->get\_filter( )->get\_as\_ranges( ).  "lv\_belnr = VALUE #( lt\_filter\_cond[ parameter\_name = 'BELNR' ]-value OPTIONAL ). "Fetching the parameter value  "lv\_gjahr = VALUE #( lt\_filter\_cond[ parameter\_name = 'GJAHR' ]-value OPTIONAL ). "Fetching the parameter value  IF lt\_filter\_cond2 IS NOT INITIAL.  DATA(lr\_belnr) = VALUE rstt\_t\_range\_string( lt\_filter\_cond2[ name = 'BELNR' ]-range OPTIONAL ). "Fetching the parameter value  DATA(lr\_gjahr) = VALUE rstt\_t\_range\_string( lt\_filter\_cond2[ name = 'GJAHR' ]-range OPTIONAL ). "Fetching the parameter value  ENDIF.  CALL METHOD io\_request->get\_sort\_elements  RECEIVING  rt\_sort\_elements = lt\_sort\_elements.  lv\_param = 'PBETU'.  ">001< eingefügt 20220916 notterr  SELECT SINGLE \* FROM zfidunnmapusr INTO @DATA(ls\_zfidunnmapusr) WHERE uname = @sy-uname AND aktiv EQ 'X'.  IF sy-subrc EQ 0.  MOVE ls\_zfidunnmapusr-uname\_deputy TO lv\_bname.  ELSE.  ">001< Ende eingefügt 20220916 notterr  MOVE sy-uname TO lv\_bname.  ">001< eingefügt 20220916 notterr  ENDIF.  ">001< Ende eingefügt 20220916 notterr  "MOVE 'WILDIA' TO lv\_bname.  CALL METHOD zficl\_nb=>get\_forderungen  EXPORTING  iv\_param = lv\_param  iv\_bname = lv\_bname  ir\_belnr = lr\_belnr  ir\_gjahr = lr\_gjahr  IMPORTING  et\_inbox\_e = DATA(lt\_dunnings\_tmp). "gt\_dunnings\_tmp. "gt\_inb\_e.  IF lt\_dunnings\_tmp IS NOT INITIAL.  LOOP AT lt\_dunnings\_tmp ASSIGNING FIELD-SYMBOL(<fs\_dunnings>).  MOVE-CORRESPONDING <fs\_dunnings> TO ls\_result.  ls\_result-ftype\_desc = SWITCH #( <fs\_dunnings>-ftype WHEN 'A' THEN 'Inkasso' ELSE 'Mahnsperre' ).  CALL METHOD zficl\_nb=>get\_forderungen\_detail(  EXPORTING  iv\_Belnr = ls\_result-belnr  iv\_gjahr = ls\_result-gjahr  iv\_buzei = ls\_result-buzei  IMPORTING  et\_detail = ls\_result-positions[] ).  CALL METHOD me->get\_pdf(  EXPORTING  iv\_belnr = ls\_result-belnr  iv\_gjahr = ls\_result-gjahr  IMPORTING  ev\_pdf = ls\_result-pdf  ev\_pdf\_url = ls\_result-pdf\_url ).  APPEND ls\_result TO lt\_result. CLEAR ls\_result.  ENDLOOP.  ENDIF.  DATA(lv\_agg) = io\_request->get\_aggregation( ).  io\_response->set\_data( lt\_result ).  DATA(lv\_offset) = io\_request->get\_paging( )->get\_offset( ).  DATA(lv\_page\_size) = io\_request->get\_paging( )->get\_page\_size( ).  " io\_response->set\_total\_number\_of\_records( lines( lt\_result ) ).  "IF\_RAP\_QUERY\_RESPONSE~SET\_DATA->set\_data( lt\_result ).  " io\_response->set\_data( lt\_result ).  io\_response->set\_total\_number\_of\_records( lines( lt\_result ) ).  "io\_response->set\_total\_number\_of\_records( lo\_response->get\_count( ) ).  " ENDIF.  CATCH cx\_rap\_query\_provider INTO DATA(lx\_exc).  ENDTRY.  CATCH cx\_rfc\_dest\_provider\_error INTO DATA(lx\_dest).  ENDTRY.  CATCH cx\_rap\_query\_filter\_no\_range.  ENDTRY.  ENDMETHOD.  METHOD get\_pdf.  DATA: lv\_object\_id TYPE saeobjid,  lv\_arc\_doc\_id TYPE saeardoid,  lv\_arch\_id TYPE saearchivi,  lv\_tabref TYPE REF TO data,  wf\_pdf\_x TYPE xstring,  lt\_URI\_TABLE TYPE TABLE OF toauri.  \* Ermittlung der Archivzuordnung und Verknüpfungstabelle  SELECT SINGLE \* FROM toaom  INTO @DATA(ls\_toaom)  WHERE sap\_object = 'BKPF'  AND ar\_object = @zsdcl\_verrechnungstool=>cv\_zfidociv "'ZFIDOCIV'  AND ar\_status = @abap\_true.  CHECK sy-subrc = 0.  lv\_object\_id = zodbcl\_tools=>mc\_ethz\_bukrs && iv\_belnr && iv\_gjahr.  SELECT SINGLE arc\_doc\_id, archiv\_id FROM (ls\_toaom-connection)  INTO ( @lv\_arc\_doc\_id, @lv\_arch\_id )  WHERE ( sap\_object = 'BKPF' )  AND ( object\_id = @lv\_object\_id )  AND ( archiv\_id = @ls\_toaom-archiv\_id OR archiv\_id = 'F1')  AND ( ar\_object = 'ZFIDOCIV' OR ar\_object = 'SDOINVOICE' ) ."'ZFIDOC01'. "@zsdcl\_verrechnungstool=>cv\_zfidociv.  \*\*Ausnahmen behandeln  IF sy-subrc NE 0.  DATA(lv\_faktura) = |%{ iv\_belnr }%|.  " Faktura lesen  SELECT SINGLE arc\_doc\_id, archiv\_id FROM (ls\_toaom-connection)  INTO ( @lv\_arc\_doc\_id, @lv\_arch\_id )  WHERE ( sap\_object = 'VBRK' )  AND ( object\_id LIKE @lv\_faktura )  AND ( archiv\_id = @ls\_toaom-archiv\_id OR archiv\_id = 'L1' )  AND ( ar\_object = 'SDOINVOICE' ) ."'ZFIDOC01'. "@zsdcl\_verrechnungstool=>cv\_zfidociv.  IF sy-subrc NE 0.  DATA(lv\_error) = abap\_true.  ENDIF.  ENDIF.  CHECK lv\_error IS INITIAL.  TRY .  wf\_pdf\_x = zbcarchcl\_archive=>get\_data\_from\_archive(  EXPORTING  iv\_archiv\_id = lv\_arch\_id  iv\_arc\_doc\_id = lv\_arc\_doc\_id ).  CATCH cx\_root.  lv\_error = abap\_true.  ENDTRY.  CHECK lv\_error IS INITIAL.  MOVE wf\_pdf\_x TO ev\_pdf.  CALL FUNCTION 'ARCHIVOBJECT\_GET\_URI'  EXPORTING  objecttype = 'BKPF'  object\_id = lv\_object\_id  location = space  TABLES  uri\_table = lt\_URI\_TABLE  EXCEPTIONS  error\_archiv = 1  error\_communicationtable = 2  error\_kernel = 3  error\_http = 4  error\_dp = 5  OTHERS = 6.  IF sy-subrc = 0.  READ TABLE lt\_URI\_TABLE INTO DATA(ls\_uri\_table) INDEX 1.  IF sy-subrc = 0.  ev\_pdf\_url = ls\_uri\_table-uri.  REPLACE FIRST OCCURRENCE OF 'https://ot-art5.ethz.ch:8090/' IN ev\_pdf\_url  WITH 'https://ethisflp-d.ethz.ch/'.  \* REPLACE FIRST OCCURRENCE OF 'https://ethisflp-d.ethz.ch' IN ev\_pdf\_url  \* WITH 'space'.  \*  \* SHIFT ev\_pdf\_url LEFT DELETING LEADING space.  ENDIF.  ENDIF.  ENDMETHOD.  ENDCLASS. |

* 1. Class ZFICL\_NB

|  |
| --- |
| class ZFICL\_NB definition  public  final  create public .  public section.  types:  begin of ls\_ty\_text,  text\_type type string,  langu type langu,  text\_tab type string\_table,  end of ls\_ty\_text .  types:  lt\_ty\_text type hashed table of ls\_ty\_text  with unique key text\_type langu .  constants CO\_TEXT\_ABSCHLUSS type STRING value 'ABSCHLUSS' ##NO\_TEXT.  constants CO\_TEXT\_NACHFRAGE type STRING value 'NACHFRAGE' ##NO\_TEXT.  constants CO\_TEXT\_NACHFRAGE\_1 type STRING value 'NACHFRAGE\_1' ##NO\_TEXT.  constants CO\_TEXT\_NACHFRAGE\_2 type STRING value 'NACHFRAGE\_2' ##NO\_TEXT.  constants CO\_TEXT\_FORD\_MAIL type STRING value 'FORD\_MAIL' ##NO\_TEXT.  constants CO\_TEXT\_NATP\_INFO\_RW type STRING value 'NATP\_INFO\_RW' ##NO\_TEXT.  constants CO\_TEXT\_NATP\_INFO\_RW\_P type STRING value 'NATP\_INFO\_RW\_P' ##NO\_TEXT.  constants CO\_TEXT\_ANREDE type STRING value 'ANREDE' ##NO\_TEXT.  constants CO\_TEXT\_ID\_ZUNF type TDID value 'ZUNF' ##NO\_TEXT.  constants CO\_TEXT\_ID\_ZINS type TDID value 'ZINS' ##NO\_TEXT.  constants CO\_TEXT\_ID\_ST type TDID value 'ST' ##NO\_TEXT.  constants CO\_TEXT\_OBJECT\_ZUNF type TDOBJECT value 'TEXT' ##NO\_TEXT.  constants CO\_TEXT\_OBJECT\_ZINS type TDOBJECT value 'TEXT' ##NO\_TEXT.  constants CO\_SENDER\_UNAME type UNAME value 'SAPMAIL\_FFM4' ##NO\_TEXT.  constants CO\_LANGU\_NATP type SPRAS value 'D' ##NO\_TEXT.  constants CO\_BU\_TYPE\_1 type BU\_TYPE value '1' ##NO\_TEXT.  constants CO\_BU\_TYPE\_2 type BU\_TYPE value '2' ##NO\_TEXT.  class-methods UPLOAD\_DOKUMENT\_TMP  importing  !IV\_ZAEHLER type ZFI\_E\_NB\_LAUFNR  exporting  !ET\_RETURN type ZFI\_T\_NB\_RETURN .  class-methods UPLOAD\_DOKUMENT\_ECHT  importing  !IV\_BP type BU\_PARTNER  exporting  !ET\_RETURN type ZFI\_T\_NB\_RETURN .  class-methods GET\_FS\_OPENFILE  importing  !IV\_FILENAME type STRING  exporting  !EV\_FILENAME type STRING  !EV\_XSTRING type XSTRING .  class-methods GET\_POPUP\_HISTORY  importing  !IV\_BUKRS type BUKRS optional  !IV\_BELNR type BELNR\_D  !IV\_GJAHR type GJAHR optional  exporting  !EV\_BUKRS type BUKRS  !EV\_BELNR type BELNR\_D  !EV\_GJAHR type GJAHR  !EV\_SUBRC type SUBRC .  class-methods MAIL\_REPLACE\_PARAMETER  importing  !IT\_PARAMETER\_VALUE type WDR\_SIMPLE\_NAME\_VALUE\_MAP  !IT\_TEXT\_RAW type STRING\_TABLE  returning  value(ET\_TEXT\_FORMAT) type STRING\_TABLE .  class-methods MAIL\_UNFO\_NACHFRAGE  importing  !IS\_DATEN type ZFI\_S\_NB\_UNECHTE\_MAHNDRUCK optional  returning  value(ET\_TEXT\_NACHFRAGE) type STRING\_TABLE .  class-methods MAIL\_UNFO\_BEGRUESSUNG  importing  !IS\_DATEN type ZFI\_S\_NB\_UNECHTE\_MAHNDRUCK optional  returning  value(ET\_TEXT\_ANREDE) type STRING\_TABLE .  class-methods MAIL\_UNFO\_ABSCHLUSS  importing  !IS\_DATEN type ZFI\_S\_NB\_UNECHTE\_MAHNDRUCK optional  returning  value(ET\_TEXT\_ABSCHLUSS) type STRING\_TABLE .  class-methods MAIL\_FORD\_INFORMATION  importing  !IS\_DATEN type ZFI\_S\_NB\_FORD\_MAIL optional  returning  value(ET\_TEXT\_INFO) type STRING\_TABLE .  class-methods MAIL\_NATP\_INFORMATION\_RW  importing  !IS\_DATEN type ZFI\_S\_NB\_NATP\_INFO optional  returning  value(ET\_TEXT\_INFO) type STRING\_TABLE .  class-methods MAIL\_INST\_ABSCHLUSS  importing  !IS\_DATEN type ZFI\_S\_NB\_BENACH\_MAHNDRUCK optional  returning  value(ET\_TEXT\_ABSCHLUSS) type STRING\_TABLE .  class-methods MAIL\_INST\_BEGRUESSUNG  importing  !IS\_DATEN type ZFI\_S\_NB\_BENACH\_MAHNDRUCK optional  returning  value(ET\_TEXT\_ANREDE) type STRING\_TABLE .  class-methods MAIL\_INST\_NACHFRAGE  importing  !IS\_DATEN type ZFI\_S\_NB\_BENACH\_MAHNDRUCK optional  returning  value(ET\_TEXT\_NACHFRAGE) type STRING\_TABLE .  class-methods MAIL\_FORD\_GET\_RAW\_TEXT  importing  !IV\_TEXT\_TYPE type CSEQUENCE  !IV\_LANGU type LANGU  returning  value(ET\_RAW\_TEXT) type STRING\_TABLE .  class-methods MAIL\_UNFO\_GET\_RAW\_TEXT  importing  !IV\_TEXT\_TYPE type CSEQUENCE  !IV\_LANGU type LANGU  returning  value(ET\_RAW\_TEXT) type STRING\_TABLE .  class-methods MAIL\_NATP\_GET\_RAW\_TEXT  importing  !IV\_TEXT\_TYPE type CSEQUENCE  !IV\_LANGU type LANGU  returning  value(ET\_RAW\_TEXT) type STRING\_TABLE .  class-methods MAIL\_INST\_GET\_RAW\_TEXT  importing  !IV\_TEXT\_TYPE type CSEQUENCE  !IV\_LANGU type LANGU  returning  value(ET\_RAW\_TEXT) type STRING\_TABLE .  class-methods MAIL\_FORD\_PROCESS\_TEXT  importing  !IV\_TEXT\_TYPE type CSEQUENCE  !IT\_PARAMETER\_VALUE type WDR\_SIMPLE\_NAME\_VALUE\_MAP optional  !IV\_LANGU type LANGU default SY-LANGU  returning  value(ET\_TEXT) type STRING\_TABLE .  class-methods MAIL\_INST\_PROCESS\_TEXT  importing  !IV\_TEXT\_TYPE type CSEQUENCE  !IT\_PARAMETER\_VALUE type WDR\_SIMPLE\_NAME\_VALUE\_MAP optional  !IV\_LANGU type LANGU default SY-LANGU  returning  value(ET\_TEXT) type STRING\_TABLE .  class-methods MAIL\_NATP\_PROCESS\_TEXT  importing  !IV\_TEXT\_TYPE type CSEQUENCE  !IT\_PARAMETER\_VALUE type WDR\_SIMPLE\_NAME\_VALUE\_MAP optional  !IV\_LANGU type LANGU default SY-LANGU  returning  value(ET\_TEXT) type STRING\_TABLE .  class-methods MAIL\_UNFO\_PROCESS\_TEXT  importing  !IV\_TEXT\_TYPE type CSEQUENCE  !IT\_PARAMETER\_VALUE type WDR\_SIMPLE\_NAME\_VALUE\_MAP optional  !IV\_LANGU type LANGU default SY-LANGU  returning  value(ET\_TEXT) type STRING\_TABLE .  class-methods GET\_ATTACHMENTS\_2  importing  !IV\_BUKRS type BUKRS  !IV\_BELNR type BELNR\_D  !IV\_GJAHR type GJAHR  exporting  !ET\_CONNECTIONS type CONNECTION\_TAB .  class-methods GET\_ATTACHMENTS  importing  !IV\_BUKRS type BUKRS  !IV\_BELNR type BELNR\_D  !IV\_GJAHR type GJAHR  exporting  !ET\_DISPDOCS type ALINK\_CONNECTION\_TABLE .  class-methods MAHNSPERRRE\_ENTFERNEN  importing  !IS\_FORDERUNG\_KEY type ZFI\_S\_NB\_FORDERUNG\_KEY  !IV\_AENDERER type USNAM  exporting  !EV\_SUBRC type SUBRC .  class-methods GET\_VBUND\_FROM\_BP  importing  !IV\_BP type BU\_PARTNER  exporting  !EV\_VBUND type RASSC .  class-methods SPERREN\_ENTSPERREN\_10K  importing  !IV\_BUKRS type BUKRS  !IV\_GJAHR type GJAHR  !IV\_BELNR type BELNR\_D  !IV\_BUZEI type BUZEI  !IV\_ACTION type CHAR1 optional  exporting  !EV\_LOCK\_UNAME type SYUNAME  returning  value(RV\_GESPERRT) type ABAP\_BOOL .  class-methods GET\_LIFNR\_FROM\_BP  importing  !IV\_BP type BU\_PARTNER  exporting  !EV\_LIFNR type LIFNR  !EV\_NAME1 type NAME1\_GP  exceptions  NO\_BP  NO\_VENDOR .  class-methods GET\_KUNNR\_FROM\_BP  importing  !IV\_BP type BU\_PARTNER  exporting  !EV\_KUNNR type KUNNR  !EV\_NAME1 type NAME1\_GP  exceptions  NO\_BP  NO\_CUSTOMER .  class-methods SPERREN\_ENTSPERREN\_RU  importing  !IV\_BUKRS type BUKRS  !IV\_GJAHR type GJAHR  !IV\_BELNR type BELNR\_D  !IV\_BUZEI type BUZEI  !IV\_ACTION type CHAR1 optional  exporting  !EV\_LOCK\_UNAME type SYUNAME  returning  value(RV\_GESPERRT) type ABAP\_BOOL .  class-methods GET\_FORDERUNG\_AUSBUCH\_PDF  importing  !IS\_AUSBUCHUNG type ZFI\_S\_NB\_FORD\_AUSBUCHUNG .  class-methods GET\_FORDERUNGSTYP  importing  !IV\_MANSP type MANSP  !IV\_BLART type BLART  !IV\_VKBUR type VKBUR  exporting  !EV\_TYP type CHAR1 .  class-methods CHECK\_DOKUMENTATION  importing  !IV\_ID type DOKU\_ID  !IV\_OBJECT type DOKU\_OBJ  !IV\_LANGU type DOKU\_LANGU  exporting  !EV\_RESULT type ABAP\_BOOL .  class-methods CHECK\_SANKTIONEN\_RUSSLAND  importing  !IV\_LIFNR type LIFNR  exporting  !EV\_SANKTION type ABAP\_BOOL .  class-methods CHECK\_SV\_NUMMER  importing  !IV\_SV\_NUMMER type CHAR16  exporting  !EV\_ERROR type CHAR1 .  class-methods CHECK\_USER  importing  !IV\_UNAME type UNAME  exporting  !EV\_INAKTIV type ABAP\_BOOL .  class-methods CREATE\_FORDERUNG\_AKTION  importing  !IS\_FORDERUNG\_KEY type ZFI\_S\_NB\_FORDERUNG\_KEY  !IV\_AENDERER type USNAM  !IV\_AKTION type ZFI\_E\_NB\_ATYP  !IV\_FLAG type CHAR1 optional  !IV\_DATUM type DATS optional  !IV\_BEMERKUNG type CHAR50 optional  !IV\_ERFASSER type USNAM optional  !IV\_ENAME type AD\_NAMTEXT optional  !IV\_MAIL type AD\_SMTPADR optional  !IV\_DATUM\_ALT type DATS optional  !IV\_BEMERKUNG\_ALT type CHAR50 optional  !IV\_ERFASSER\_ALT type USNAM optional  !IV\_ENAME\_ALT type AD\_NAMTEXT optional  !IV\_MAIL\_ALT type AD\_SMTPADR optional  !IV\_BEGRUENDUNG type TXT1024 optional  !IS\_RAP\_BEHAVIOR type ABAP\_BOOL optional  exporting  !EV\_SUBRC type SUBRC .  class-methods CREATE\_FORDERUNG\_AUSBUCHUNG  importing  !IS\_FORDERUNG\_KEY type ZFI\_S\_NB\_FORDERUNG\_KEY  !IV\_ERFASSER type USNAM optional  !IV\_DATUM type DATS optional  !IV\_BEGRUENDUNG type TXT1024 optional  !IS\_RAP\_BEHAVIOR type ABAP\_BOOL optional  exporting  !EV\_SUBRC type SUBRC .  class-methods CREATE\_FORDERUNG\_HISTORY  importing  !IS\_FORDERUNG\_KEY type ZFI\_S\_NB\_FORDERUNG\_KEY  !IV\_AENDERER type USNAM  !IV\_AKTION type ZFI\_E\_NB\_ATYP  !IV\_FLAG type CHAR1 optional  !IV\_DATUM type DATS optional  !IV\_BEMERKUNG type TEXT50 optional  !IV\_ERFASSER type USNAM optional  !IV\_ENAME type AD\_NAMTEXT optional  !IV\_MAIL type AD\_SMTPADR optional  !IV\_DATUM\_ALT type DATS optional  !IV\_BEMERKUNG\_ALT type TEXT50 optional  !IV\_ERFASSER\_ALT type USNAM optional  !IV\_ENAME\_ALT type AD\_NAMTEXT optional  !IV\_MAIL\_ALT type AD\_SMTPADR optional  !IS\_RAP\_BEHAVIOR type ABAP\_BOOL optional  exporting  !EV\_SUBRC type SUBRC .  class-methods CREATE\_FORDERUNG\_ZUSATZ  importing  !IS\_FORDERUNG\_KEY type ZFI\_S\_NB\_FORDERUNG\_KEY  !IV\_AKTION type CHAR1  !IV\_AENDERER type USNAM optional  !IV\_DATUM type DATS optional  !IV\_FLAG type CHAR1 optional  !IV\_ERFASSER type USNAM optional  !IV\_ENAME type AD\_NAMTEXT optional  !IV\_MAIL type AD\_SMTPADR optional  !IV\_BEMERKUNG type TEXT50 optional  !IS\_RAP\_BEHAVIOR type ABAP\_BOOL optional  exporting  !EV\_SUBRC type SUBRC .  class-methods DO\_SHOW\_PDF  importing  !IV\_PDF type XSTRING .  class-methods GET\_ADRESSDATEN  importing  !IV\_BNAME type UNAME  exporting  !ES\_ADRESSDATEN type ZFI\_S\_NB\_ADRESSDATEN .  class-methods GET\_ANREDE  importing  !IV\_TITLE type AD\_TITLE  !IV\_LANGU type SPRAS  exporting  !EV\_ANREDE type AD\_TITLETX .  class-methods GET\_BELEGAENDERUNGEN  importing  !IS\_KEY type ZFI\_S\_NB\_FORDERUNG\_KEY  returning  value(ET\_HISTORY) type ZFI\_T\_NB\_FORD\_HIST .  class-methods GET\_BP\_CUSTOMER\_DATA  importing  !IV\_BP type BU\_PARTNER  exporting  !ES\_BUT000 type BUT000  !ES\_KNA1 type KNA1  !ES\_ADRC type ADRC  !EV\_KUNNR type KUNNR .  class-methods GET\_BP\_FROM\_KUNNR  importing  !IV\_KUNNR type KUNNR  exporting  !EV\_BP type BU\_PARTNER .  class-methods GET\_BP\_FROM\_LIFNR  importing  !IV\_LIFNR type LIFNR  exporting  !EV\_BP type BU\_PARTNER .  class-methods GET\_BP\_VENDOR\_DATA  importing  !IV\_BP type BU\_PARTNER  exporting  !ES\_BUT000 type BUT000  !ES\_LFA1 type LFA1  !ES\_ADRC type ADRC  !EV\_LIFNR type LIFNR .  class-methods GET\_ERFASSER\_DR  importing  !IV\_BUKRS type BUKRS  !IV\_BELNR type BELNR\_D  !IV\_GJAHR type GJAHR  exporting  !ES\_ADRESSDATEN type ZFI\_S\_NB\_ADRESSDATEN .  class-methods GET\_ERFASSER\_FORDERUNG\_CDHDR  importing  !IV\_VBELV type VBELN\_VON  !IV\_POSNV type POSNR\_VON  exporting  !EV\_BNAME type USNAM .  class-methods GET\_ERFASSER\_FORDERUNG\_FI  importing  !IV\_BUKRS type BUKRS  !IV\_BELNR type BELNR\_D  !IV\_GJAHR type GJAHR  exporting  !ES\_ADRESSDATEN type ZFI\_S\_NB\_ADRESSDATEN .  class-methods GET\_ERFASSER\_FORDERUNG\_SD  importing  !IV\_BUKRS type BUKRS  !IV\_BELNR type BELNR\_D  !IV\_GJAHR type GJAHR  exporting  !ES\_ADRESSDATEN type ZFI\_S\_NB\_ADRESSDATEN .  class-methods GET\_ERFASSER\_FORDERUNG\_VT  importing  !IV\_BUKRS type BUKRS  !IV\_BELNR type BELNR\_D  !IV\_GJAHR type GJAHR  exporting  !ES\_ADRESSDATEN type ZFI\_S\_NB\_ADRESSDATEN .  class-methods GET\_ERFASSER\_RD  importing  !IV\_BUKRS type BUKRS  !IV\_BELNR type BELNR\_D  !IV\_GJAHR type GJAHR  !IV\_PROJK type PS\_PSP\_PNR  exporting  !ES\_ADRESSDATEN type ZFI\_S\_NB\_ADRESSDATEN .  class-methods GET\_ERFASSER\_RD\_SD  importing  !IV\_VBELN type VBELN\_VA  !IV\_PROJK type PS\_PSP\_PNR  exporting  !ES\_ADRESSDATEN type ZFI\_S\_NB\_ADRESSDATEN .  class-methods GET\_ERFASSER\_VD  importing  !IV\_BUKRS type BUKRS  !IV\_BELNR type BELNR\_D  !IV\_GJAHR type GJAHR  exporting  !ES\_ADRESSDATEN type ZFI\_S\_NB\_ADRESSDATEN .  class-methods GET\_FORDERUNGEN  importing  !IV\_PARAM type CHAR8  !IV\_BNAME type UNAME optional  !IR\_BELNR type RSTT\_T\_RANGE\_STRING optional  !IR\_GJAHR type RSTT\_T\_RANGE\_STRING optional  exporting  !ET\_INBOX\_E type ZFI\_T\_NB\_FORD\_INBOX\_ETHIS  !ET\_INBOX\_S type ZFI\_T\_NB\_FORD\_INBOX\_SAPGUI .  class-methods GET\_FORDERUNGEN\_DETAIL  importing  !IV\_BELNR type BELNR\_D  !IV\_GJAHR type GJAHR  !IV\_BUZEI type BUZEI optional  exporting  !ET\_DETAIL type ZFI\_T\_NB\_FORDERUNGEN\_DETAILS .  class-methods GET\_FORDERUNGEN\_VERDICHTUNG  importing  !IS\_KEY type ZFI\_S\_NB\_FORDERUNG\_KEY  exporting  !ES\_BSEG type ZFI\_S\_NB\_FORD\_BSEG .  class-methods GET\_ARC\_DOC\_ID  importing  !IV\_BUKRS type BUKRS  !IV\_BELNR type BELNR\_D  !IV\_GJAHR type GJAHR  !IV\_SAP\_OBJECT type SAEANWDID  exporting  !EV\_ARCHIV\_ID type SAEARCHIVI  !EV\_ARC\_DOC\_ID type SAEARDOID  !EV\_AR\_OBJECT type SAEOBJART .  class-methods GET\_DATA\_FROM\_ARCHIV  importing  !IV\_ARCHIV\_ID type CLIKE  !IV\_ARC\_DOC\_ID type CLIKE  !IV\_DOCUMENT\_TYPE type CLIKE optional  returning  value(EV\_DATA\_XSTRING) type XSTRING .  class-methods MAIL\_INST\_SEND  importing  !IS\_DATEN type ZFI\_S\_NB\_BENACH\_MAHNDRUCK  !IT\_ATTACH type ZFI\_T\_NB\_BENACH\_ATTACHMENT  !IV\_TEST type CHAR1 .  class-methods MAIL\_FORD\_SEND  importing  !IS\_FORD\_MAIL type ZFI\_S\_NB\_FORD\_MAIL .  class-methods MAIL\_NATP\_SEND  importing  !IS\_NATP\_INFO type ZFI\_S\_NB\_NATP\_INFO .  class-methods MAIL\_UNFO\_SEND  importing  !IS\_DATEN type ZFI\_S\_NB\_UNECHTE\_MAHNDRUCK  !IT\_ATTACH type ZFI\_T\_NB\_UNECHTE\_ATTACHMENT  !IV\_TEST type CHAR1 .  class-methods SEND\_BCS\_MAIL  importing  !IV\_EMAIL\_SENDER type CLIKE optional  !IV\_USER\_SENDER type CLIKE optional  !IT\_BODY type BCSY\_TEXT  !IT\_ATTACH type ZFI\_T\_NB\_UNECHTE\_ATTACHMENT optional  !IV\_SUBJECT type CLIKE  !IV\_IMPORTANCE type CLIKE optional  !IV\_SEND\_IMMEDIATELY type ABAP\_BOOL default ABAP\_FALSE  !IT\_RECIPIENT\_TO type STRING\_TABLE optional  !IT\_RECIPIENT\_CC type STRING\_TABLE optional  !IT\_RECIPIENT\_BCC type STRING\_TABLE optional  !IV\_CONTENT\_TYPE type CLIKE  !IV\_WITH\_COMMIT\_WORK type ABAP\_BOOL optional  !IV\_SPRAS type LANGU optional .  protected section.  private section.  class-data GT\_TEXT type ZFICL\_NB=>LT\_TY\_TEXT .  ENDCLASS.  CLASS ZFICL\_NB IMPLEMENTATION.  method check\_dokumentation.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Es wird geprüft, ob eine Programmdoku vorhanden ist  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_id type doku\_id,  lv\_object type doku\_obj,  lv\_langu type doku\_langu,  lv\_result type abap\_bool.  data: lt\_dokhl type table of dokhl,  ls\_dokhl type dokhl,  lt\_doktl type table of doktl,  ls\_doktl type doktl.  constants: co\_langu type doku\_langu value 'D'.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_id = iv\_id.  lv\_object = iv\_object.  lv\_langu = iv\_langu.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  clear lv\_result.  \* lesen der Dokumenationstabellen  \* Kopftabelle  select \* from dokhl into corresponding fields of table lt\_dokhl  where id = lv\_id  and object = lv\_object  and langu = co\_langu.  \* Zeilentabelle  select \* from doktl into corresponding fields of table lt\_doktl  where id = lv\_id  and object = lv\_object  and langu = co\_langu  and ( dokformat ne 'U1' and dokformat ne 'U2')  and doktext ne space.  \* Einträge vorhanden ?  \* if lt\_dokhl[] is initial and lt\_doktl[] is initial.  if lt\_doktl[] is initial.  lv\_result = abap\_false.  else .  lv\_result = abap\_true.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  ev\_result = lv\_result.  endmethod.  method check\_sanktionen\_russland.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* es wird geprüft, ob der BP relevant für Sanktionen ist  \*  \* Abfrage:  \* Holen der Prüfkriterien ab zfi\_kred\_c\_ru  \* Flag 'CHECK\_LAND' gesetzt  \* - prüfen BP Länderkennzeichen Russland  \* Flag 'CHECK\_LAND' gesetzt  \* - prüfen BP Bankverbindungen RU  \* Flag 'CHECK\_SNB' gesetzt  \* - prüfen BP Bankverbindungen SWIFT auf Sanktionsliste  \*  \* wenn ein Flag aus Prüfung gesetzt --> EV-Sanktion = 'X'.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* 21.02.2022 NIW: Belarus ebenfalls berücksichtigen  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_lifnr type lifnr,  lv\_partner type bu\_partner,  ls\_adrc type adrc,  ls\_c\_ru type zfi\_kred\_c\_ru,  ls\_c\_bk type zfi\_kred\_c\_ru\_bk.  data: lv\_rel\_land type flag,  lv\_rel\_bank type flag,  lv\_rel\_snb type flag.  types: begin of bank,  banks type banks,  bankl type bankl,  swift type swift.  types: end of bank.  data: lt\_bank type table of bank,  ls\_bank like line of lt\_bank.  field-symbols: <ls\_bank> like ls\_bank.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  clear: lv\_rel\_land, lv\_rel\_bank, lv\_rel\_snb.  clear: ls\_c\_ru.  \* Customizingeinträge holen  select \* from zfi\_kred\_c\_ru into corresponding fields of ls\_c\_ru.  endselect.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  \* Muss Prüfung durchlaufen werden ?  if ls\_c\_ru-check\_land is initial  and ls\_c\_ru-check\_bank is initial  and ls\_c\_ru-check\_snb is initial.  \* keine Aktion  else.  \* Holen BP aufgrund LIFNR  call method zficl\_nb=>get\_bp\_from\_lifnr  exporting  iv\_lifnr = iv\_lifnr  importing  ev\_bp = lv\_partner.  \* Holen Land des BP's  call method zficl\_nb=>get\_bp\_vendor\_data  exporting  iv\_bp = lv\_partner  importing  es\_adrc = ls\_adrc.  if ls\_c\_ru-check\_land = abap\_true.  if ls\_adrc-country = 'RU' or  ls\_adrc-country = 'BY'. "do. Belarus  lv\_rel\_land = abap\_true.  endif.  endif.  \* check Bankverbindungen RU  select bnka~banks, bnka~bankl, bnka~swift from bnka left outer join but0bk  on but0bk~banks = bnka~banks  and but0bk~bankl = bnka~bankl  into corresponding fields of table @lt\_bank  where but0bk~partner = @lv\_partner  and ( but0bk~banks = 'RU' or but0bk~banks = 'BY' ) .  if ls\_c\_ru-check\_bank = abap\_true.  if not lt\_bank[] is initial.  lv\_rel\_bank = abap\_true.  endif.  endif.  \* Check SWIFT gemäss Bankverbindung auf Sanktionsliste  if ls\_c\_ru-check\_snb = abap\_true.  loop at lt\_bank assigning <ls\_bank>.  select \* from zfi\_kred\_c\_ru\_bk into ls\_c\_bk  where banks = 'RU'  or banks = 'BY'  and bankl = <ls\_bank>-bankl  and sanktion = 'X'.  endselect.  \* gefunden  if sy-subrc = 0. "relevanter SWIFT mit 'X' gefunden  lv\_rel\_snb = abap\_true.  exit.  endif.  endloop.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  if lv\_rel\_land = abap\_true  or lv\_rel\_bank = abap\_true  or lv\_rel\_snb = abap\_true.  ev\_sanktion = abap\_true.  endif.  endif.  endmethod.  method check\_sv\_nummer.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* hier wird die SV-Nummer geprüft auf formale Gültigkeit  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_sv\_nummer type char16,  lv\_error type char1.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_sv\_nummer = iv\_sv\_nummer.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  clear lv\_error.  if lv\_sv\_nummer(3) cn '756'.  lv\_error = abap\_true.  else.  if lv\_sv\_nummer+3(1) cn '.'.  lv\_error = abap\_true.  else.  if lv\_sv\_nummer+4(4) cn '0123456789'.  lv\_error = abap\_true.  else.  if lv\_sv\_nummer+8(1) cn '.'.  lv\_error = abap\_true.  else.  if lv\_sv\_nummer+9(4) cn '0123456789'.  lv\_error = abap\_true.  else.  if lv\_sv\_nummer+13(1) cn '.'.  lv\_error = abap\_true.  else.  if lv\_sv\_nummer+14(2) cn '0123456789'.  lv\_error = abap\_true.  endif.  endif.  endif.  endif.  endif.  endif.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  ev\_error = lv\_error.  endmethod.  method check\_user.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Es wird geprüft, ob der User gemäss ZUID noch aktiv ist  \*  \* Abfrage:  \* - Username = SAP-User aber klein geschrieben  \* - is\_mainuser = 'X' (Achtung im S4D nicht immer gepflegt)  \* - is\_employe = 'X'  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_uname type uname,  lv\_userid type xubname.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_uname = iv\_uname.  \* Umstellen auf Kleinbuchstaben  translate lv\_uname to lower case.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  ev\_inaktiv = abap\_true.  \* Lesen der Tabelle ZUID  select sap\_user from zuid into lv\_userid  where username = lv\_uname  and is\_mainuser = abap\_true  and is\_employee = abap\_true.  endselect.  if sy-subrc = 0. "User ist inaktiv  ev\_inaktiv = abap\_false.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  endmethod.  method create\_forderung\_aktion.  \*----------------------------------------------------------------------\*  \*Methodenname: ZFICL\_NB-create\_forderung\_aktion() \*  \*Entwickler: CCSAP ETH / \*  \*Beschreibung: Aktionen aus SAPUI5 verarbeiten \*  \*Programmspezifikation: keine \*  \*Verantwortliche Stelle ETHZ: CCSAP, R.Notter \*  \*----------------------------------------------------------------------\*  \*SAP-Release ERP2005 / 700 \*  \*Erstellungsdatum: \*  \*----------------------------------------------------------------------\*  \*Änderungs-Log \*  \* Datendefinitionen-----------------------------------------------------  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Aufrufen Aktion  \*  \* folgende Ereignisse werden abgehandelt  \* - N --> Neuer Eintrag  \* - F --> Fällige Forderung  \* - B --> Betreiben anfordern  \* - M --> Mahnen (Mahnsperre entfernen)  \* - A --> Ausbuchen  \* - V --> Verlängern  \* - E --> Erfasser ändern  \* - K --> Kommentar erfassen  \* - I --> Infomail versenden  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: ls\_key type zfi\_s\_nb\_forderung\_key,  lv\_bname type usnam,  lv\_aktion type zfi\_e\_nb\_atyp,  lv\_datum type dats,  lv\_bemerkung type char50,  lv\_erfasser type usnam,  lv\_ename type ad\_namtext,  lv\_mail type ad\_smtpadr,  lt\_messages type tab\_bdcmsgcoll,  ls\_messages type bdcmsgcoll.  data: lv\_subrc type subrc.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  clear lv\_subrc.  \* Verarbeitung Aktion und History  case iv\_aktion.  when 'N'.  call method zficl\_nb=>create\_forderung\_history  exporting  is\_forderung\_key = is\_forderung\_key  iv\_aenderer = iv\_aenderer  iv\_aktion = iv\_aktion  is\_rap\_behavior = is\_rap\_behavior  importing  ev\_subrc = lv\_subrc.  when 'F'.  call method zficl\_nb=>create\_forderung\_history  exporting  is\_forderung\_key = is\_forderung\_key  iv\_aenderer = iv\_aenderer  iv\_aktion = iv\_aktion  is\_rap\_behavior = is\_rap\_behavior  importing  ev\_subrc = lv\_subrc.  when 'B'. "Betreiben  call method zficl\_nb=>create\_forderung\_history  exporting  is\_forderung\_key = is\_forderung\_key  iv\_aenderer = iv\_aenderer  iv\_aktion = iv\_aktion  iv\_flag = iv\_flag  iv\_bemerkung = iv\_bemerkung  iv\_bemerkung\_alt = iv\_bemerkung\_alt  is\_rap\_behavior = is\_rap\_behavior  importing  ev\_subrc = lv\_subrc.  when 'M'. "Mahnen  call method zficl\_nb=>mahnsperrre\_entfernen  exporting  is\_forderung\_key = is\_forderung\_key  iv\_aenderer = iv\_aenderer  importing  ev\_subrc = lv\_subrc.  \*  when 'A'. "ausbuchen  call method zficl\_nb=>create\_forderung\_history  exporting  is\_forderung\_key = is\_forderung\_key  iv\_aenderer = iv\_aenderer  iv\_aktion = iv\_aktion  iv\_flag = iv\_flag  iv\_bemerkung = iv\_bemerkung  iv\_bemerkung\_alt = iv\_bemerkung\_alt  is\_rap\_behavior = is\_rap\_behavior  importing  ev\_subrc = lv\_subrc.  \*\* \* Ausbuchungsbeleg erstellen  if iv\_flag = abap\_true.  call method zficl\_nb=>create\_forderung\_ausbuchung  exporting  is\_forderung\_key = is\_forderung\_key  iv\_erfasser = iv\_aenderer  iv\_datum = sy-datum  iv\_begruendung = iv\_begruendung  is\_rap\_behavior = is\_rap\_behavior  importing  ev\_subrc = lv\_subrc.  endif.  when 'V'. "Verlängern  if iv\_datum = sy-datum. " aktion aus ETHIS + 30 Tag  lv\_datum = sy-datum + 30.  else.  " Datum kommt aus RW-Inbox, so belassen  lv\_datum = iv\_datum.  endif.  call method zficl\_nb=>create\_forderung\_history  exporting  is\_forderung\_key = is\_forderung\_key  iv\_aenderer = iv\_aenderer  iv\_aktion = iv\_aktion  iv\_datum = lv\_datum  iv\_datum\_alt = iv\_datum\_alt  is\_rap\_behavior = is\_rap\_behavior  importing  ev\_subrc = lv\_subrc.  when 'E'. "erfasser  call method zficl\_nb=>create\_forderung\_history  exporting  is\_forderung\_key = is\_forderung\_key  iv\_aenderer = iv\_aenderer  iv\_aktion = iv\_aktion  iv\_erfasser = iv\_erfasser  iv\_ename = iv\_ename  iv\_mail = iv\_mail  iv\_erfasser\_alt = iv\_erfasser\_alt  iv\_ename\_alt = iv\_ename\_alt  iv\_mail\_alt = iv\_mail\_alt  is\_rap\_behavior = is\_rap\_behavior  importing  ev\_subrc = lv\_subrc.  when 'K'. "Kommentar (Bemerkung)  call method zficl\_nb=>create\_forderung\_history  exporting  is\_forderung\_key = is\_forderung\_key  iv\_aenderer = iv\_aenderer  iv\_aktion = iv\_aktion  iv\_bemerkung = iv\_bemerkung  iv\_bemerkung\_alt = iv\_bemerkung\_alt  is\_rap\_behavior = is\_rap\_behavior  importing  ev\_subrc = lv\_subrc.  when 'I'. "Infomail  call method zficl\_nb=>create\_forderung\_history  exporting  is\_forderung\_key = is\_forderung\_key  iv\_aenderer = iv\_aenderer  iv\_aktion = iv\_aktion  iv\_datum = iv\_datum  is\_rap\_behavior = is\_rap\_behavior  importing  ev\_subrc = lv\_subrc.  endcase.  if lv\_subrc = 0.  \* Verarbeitung Zusatztabelle  case iv\_aktion.  when 'N'. "Neu Zusatztabelle Initial eintrag  call method zficl\_nb=>create\_forderung\_zusatz  exporting  is\_forderung\_key = is\_forderung\_key  iv\_aenderer = iv\_aenderer  iv\_aktion = 'N'  is\_rap\_behavior = is\_rap\_behavior  importing  ev\_subrc = lv\_subrc.  when 'M'. "Mahnen (Mahnsperre entfernen)  \* --> keine Anpassung Zusatztabelle  when others.  call method zficl\_nb=>create\_forderung\_zusatz  exporting  is\_forderung\_key = is\_forderung\_key  iv\_aenderer = iv\_aenderer  iv\_aktion = iv\_aktion  iv\_datum = iv\_datum  iv\_flag = iv\_flag  iv\_bemerkung = iv\_bemerkung  iv\_erfasser = iv\_erfasser  iv\_ename = iv\_ename  iv\_mail = iv\_mail  is\_rap\_behavior = is\_rap\_behavior  importing  ev\_subrc = lv\_subrc.  endcase.  else.  ev\_subrc = lv\_subrc.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  ev\_subrc = lv\_subrc.  endmethod.  method create\_forderung\_ausbuchung.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Erstellen Eintrag in Ausbuchungstabelle  \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_bukrs type bukrs,  lv\_belnr type belnr\_d,  lv\_gjahr type gjahr,  lv\_buzei type buzei,  lv\_aenderer type usnam,  lv\_datum type dats,  lv\_begruendung type txt1024.  data: ls\_ausbuch type zfi\_nb\_ford\_aus,  lv\_subrc type subrc.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  clear ls\_ausbuch.  \* Holen Eintrag, wenn vorhanden  select \* from zfi\_nb\_ford\_aus into corresponding fields of ls\_ausbuch  where bukrs = is\_forderung\_key-bukrs  and gjahr = is\_forderung\_key-gjahr  and belnr = is\_forderung\_key-belnr  and buzei = is\_forderung\_key-buzei.  endselect.  if sy-subrc = 0. "modify Eintrag  ls\_ausbuch-erfasser = iv\_erfasser.  ls\_ausbuch-datum = iv\_datum.  ls\_ausbuch-begruendung = iv\_begruendung.  \*  modify zfi\_nb\_ford\_aus from ls\_ausbuch.  commit work.  if sy-subrc <> 0.  lv\_subrc = 1. "Update nicht erfolgreich  endif.  else. "neuer Eintrag  ls\_ausbuch-mandt = '0330'.  ls\_ausbuch-bukrs = is\_forderung\_key-bukrs.  ls\_ausbuch-gjahr = is\_forderung\_key-gjahr.  ls\_ausbuch-belnr = is\_forderung\_key-belnr.  ls\_ausbuch-buzei = is\_forderung\_key-buzei.  ls\_ausbuch-erfasser = iv\_erfasser.  ls\_ausbuch-datum = iv\_datum.  ls\_ausbuch-begruendung = iv\_begruendung.  \*  insert zfi\_nb\_ford\_aus from ls\_ausbuch.  if is\_rap\_behavior IS INITIAL.  commit work.  endif.  if sy-subrc <> 0.  lv\_subrc = 1. "nicht erfolgreich  endif.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  ev\_subrc = lv\_subrc.  endmethod.  method create\_forderung\_history.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Erstellen Eintrag in History-Tabelle  \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_bukrs type bukrs,  lv\_belnr type belnr\_d,  lv\_gjahr type gjahr,  lv\_buzei type buzei,  lv\_bname type usnam.  data: ls\_hist type zfi\_nb\_ford\_hist,  lv\_subrc type subrc.  data: tzeit type timestampl.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  clear ls\_hist.  \* Aufbau Schlüssel.  ls\_hist-mandt = '0330'.  ls\_hist-bukrs = is\_forderung\_key-bukrs.  ls\_hist-gjahr = is\_forderung\_key-gjahr.  ls\_hist-belnr = is\_forderung\_key-belnr.  ls\_hist-buzei = is\_forderung\_key-buzei.  \* Timestamp Sekunden reichen nicht  get time stamp field ls\_hist-timestamp.  \*  ls\_hist-bname = iv\_aenderer.  ls\_hist-aktion = iv\_aktion.  case iv\_aktion.  when 'N'. "neu anlage  \* keine weiteren Angaben  when 'F'. "fällig  ls\_hist-akt\_datum = iv\_datum.  \*  when 'B'. "betreiben  if iv\_flag = abap\_true. "Betreiben gesetzt  ls\_hist-feld1 = iv\_flag.  ls\_hist-feld1\_alt = space.  else. "Betreibung aufgehoben  ls\_hist-feld1 = space.  ls\_hist-feld1\_alt = abap\_true.  endif.  when 'M'. "mahnen  ls\_hist-feld1\_alt = 'G'. "Mahnsperre 'G' entfernt  when 'A'. "ausbuchen  if iv\_flag = abap\_true. " ausbuchen verlangt  ls\_hist-feld1 = iv\_flag.  ls\_hist-feld1\_alt = space.  else. "Ausbuchen aufgehoben  ls\_hist-feld1 = space.  ls\_hist-feld1\_alt = abap\_true.  endif.  when 'V'. "verlängern  ls\_hist-akt\_datum = iv\_datum.  \* ls\_hist-datum\_alt = iv\_datum\_alt.  when 'E'. "erfasser  ls\_hist-feld1 = iv\_erfasser.  ls\_hist-feld2 = iv\_ename.  ls\_hist-feld3 = iv\_mail.  ls\_hist-feld1\_alt = iv\_erfasser\_alt.  ls\_hist-feld2\_alt = iv\_ename\_alt.  ls\_hist-feld3\_alt = iv\_mail\_alt.  when 'K'. "Kommentar  ls\_hist-feld2 = iv\_bemerkung.  ls\_hist-feld2\_alt = iv\_bemerkung\_alt.  when 'I'. "Infomail  ls\_hist-akt\_datum = iv\_datum.  \* ls\_hist-datum\_alt = iv\_datum\_alt.  endcase.  insert zfi\_nb\_ford\_hist from ls\_hist.  if is\_rap\_behavior IS INITIAL.  commit work.  endif.  if sy-subrc <> 0.  lv\_subrc = 1. " konnte nicht angelegt werden.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  ev\_subrc = lv\_subrc.  endmethod.  method create\_forderung\_zusatz.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Erstellen Eintrag in Forderungs-Zusatz-Tabelle  \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_bukrs type bukrs,  lv\_belnr type belnr\_d,  lv\_gjahr type gjahr,  lv\_buzei type buzei,  lv\_bname type usnam.  data: ls\_zusatz type zfi\_nb\_ford\_zus,  lv\_subrc type subrc.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  clear ls\_zusatz.  case iv\_aktion.  when 'N'. "anlegen intial-Eintrag  ls\_zusatz-mandt = '0330'.  ls\_zusatz-bukrs = is\_forderung\_key-bukrs.  ls\_zusatz-gjahr = is\_forderung\_key-gjahr.  ls\_zusatz-belnr = is\_forderung\_key-belnr.  ls\_zusatz-buzei = is\_forderung\_key-buzei.  insert zfi\_nb\_ford\_zus from ls\_zusatz.  if is\_rap\_behavior IS INITIAL.  commit work and wait.  endif.  if sy-subrc <> 0.  lv\_subrc = 1. "nicht erfolgreich  endif.  when others. "Mutieren Eintrag  select \* from zfi\_nb\_ford\_zus into corresponding fields of ls\_zusatz  where bukrs = is\_forderung\_key-bukrs  and gjahr = is\_forderung\_key-gjahr  and belnr = is\_forderung\_key-belnr  and buzei = is\_forderung\_key-buzei.  endselect.  if sy-subrc = 0.  \* Felder gemäss Aktionstyp versorgen  case iv\_aktion.  when 'E'. "Erfasser  ls\_zusatz-erfasser = iv\_erfasser.  ls\_zusatz-ename = iv\_ename.  ls\_zusatz-mail = iv\_mail.  when 'K'. "Kommentar  ls\_zusatz-bemerkung = iv\_bemerkung.  when 'V'. "Verlängerungsdatum  ls\_zusatz-flg\_faellig = space.  ls\_zusatz-flg\_infomail = space.  ls\_zusatz-mail\_datum = space.  ls\_zusatz-vldat = iv\_datum.  if iv\_datum is initial. "Verlängerungdatum gelöscht (RW)  ls\_zusatz-vl\_zaehler = 0.  else.  ls\_zusatz-vl\_zaehler = ls\_zusatz-vl\_zaehler + 1.  endif.  when 'F'. "Fällig  ls\_zusatz-flg\_faellig = abap\_true.  when 'I'. "Infomail.  ls\_zusatz-mail\_datum = iv\_datum.  ls\_zusatz-flg\_infomail = abap\_true.  when 'A'. "ausbuchen  ls\_zusatz-flg\_ausbuchen = iv\_flag.  when 'B'. "betreiben  ls\_zusatz-flg\_betreiben = iv\_flag.  endcase.  \*  modify zfi\_nb\_ford\_zus from ls\_zusatz.  if is\_rap\_behavior IS INITIAL.  commit work.  endif.  if sy-subrc <> 0.  lv\_subrc = 1.  endif.  else.  if sy-subrc <> 0.  lv\_subrc = 1.  endif.  endif.  endcase.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  ev\_subrc = lv\_subrc.  endmethod.  method do\_show\_pdf .  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Anzeigen PDF-Dokument  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: ls\_formoutput type fpformoutput,  lv\_sapworkdir type string,  lv\_tmp type string,  lv\_length type i,  lt\_tab type table of tbl1024,  lt\_pdf type solix\_tab.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  "Formular online Anzeigen  cl\_gui\_frontend\_services=>get\_sapgui\_workdir( changing sapworkdir = lv\_sapworkdir ).  cl\_gui\_cfw=>flush( ).  concatenate lv\_sapworkdir '\' 'temp' sy-datum sy-uzeit '.pdf' into lv\_tmp.  "  \* CALL FUNCTION 'SCMS\_XSTRING\_TO\_BINARY'  \* EXPORTING  \* buffer = iv\_pdf  \* IMPORTING  \* output\_length = lv\_length  \* TABLES  \* binary\_tab = lt\_tab.  lt\_pdf = cl\_document\_bcs=>xstring\_to\_solix( ip\_xstring = iv\_pdf ).  cl\_gui\_frontend\_services=>gui\_download(  exporting  filename = lv\_tmp  filetype = 'BIN'  \* IMPORTING  \* filelength = lv\_length  changing  data\_tab = lt\_pdf  exceptions  file\_write\_error = 1  no\_batch = 2  gui\_refuse\_filetransfer = 3  invalid\_type = 4  no\_authority = 5  unknown\_error = 6  header\_not\_allowed = 7  separator\_not\_allowed = 8  filesize\_not\_allowed = 9  header\_too\_long = 10  dp\_error\_create = 11  dp\_error\_send = 12  dp\_error\_write = 13  unknown\_dp\_error = 14  access\_denied = 15  dp\_out\_of\_memory = 16  disk\_full = 17  dp\_timeout = 18  file\_not\_found = 19  dataprovider\_exception = 20  control\_flush\_error = 21  not\_supported\_by\_gui = 22  error\_no\_gui = 23  others = 24 ).  if sy-subrc <> 0.  message id sy-msgid type 'S' number sy-msgno with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4 display like sy-msgty.  exit.  endif.  cl\_gui\_frontend\_services=>execute(  exporting  document = lv\_tmp  exceptions  cntl\_error = 1  error\_no\_gui = 2  bad\_parameter = 3  file\_not\_found = 4  path\_not\_found = 5  file\_extension\_unknown = 6  error\_execute\_failed = 7  synchronous\_failed = 8  not\_supported\_by\_gui = 9  others = 10  ).  if sy-subrc <> 0.  message id sy-msgid type 'S' number sy-msgno  with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4 display like sy-msgty.  endif.  endmethod.  method get\_adressdaten.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* es werden Adressdaten geholt aufgrund BNAME  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* 09.12.2021 NIW: wenn keine Adressdaten gefunden-Struktur löschen  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_bname type uname,  ls\_adressdaten type zfi\_s\_nb\_adressdaten,  ls\_adrc type adrc,  lv\_anrede type ad\_titletx.  data: l\_adr type ad\_addrnum,  l\_per type ad\_persnum,  lv\_street type ad\_street,  lv\_house\_num1 type ad\_hsnm1,  lv\_post\_code1 type ad\_pstcd1,  lv\_city1 type ad\_city1.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_bname = iv\_bname.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  \* Holen der Adressnummer, Personennummer des Benutzers  ls\_adressdaten-bname = lv\_bname.  select single addrnumber persnumber from usr21 into (l\_adr, l\_per)  where bname eq lv\_bname.  \*  ls\_adressdaten-adrnr = l\_adr.  ls\_adressdaten-pernr = l\_per.  \* Holen des Names der Person  if sy-subrc = 0.  \* Namen holen aufgrund der PERSNUMBER  select name\_text title langu from adrp into  ( ls\_adressdaten-ename, ls\_adressdaten-title, ls\_adressdaten-langu )  where persnumber = l\_per  and date\_from < sy-datum.  endselect.  call method zficl\_nb=>get\_anrede  exporting  iv\_title = ls\_adressdaten-title  iv\_langu = ls\_adressdaten-langu  importing  ev\_anrede = ls\_adressdaten-anrede.  \*\* wieder aktiviert am 10.03.2020 wg. Fehler im Zfiunfo, zfiinst  \*\*\*\*\* wird ersetzt durch Adresse von HR  \*Holen der Emailadresse des Erfassers der Rechnung  select single smtp\_addr from adr6 into ls\_adressdaten-email  where addrnumber eq l\_adr  and persnumber eq l\_per.  if sy-subrc = 0.  \* Firmendaten holen aufgrund der ADRNR des Users  select street house\_num1 post\_code1 city1 from adrc  into ( lv\_street, lv\_house\_num1, lv\_post\_code1, lv\_city1 )  where addrnumber = l\_adr  and date\_from < sy-datum.  endselect.  concatenate lv\_street lv\_house\_num1 into  ls\_adressdaten-strasse separated by space.  concatenate 'CH' '-' lv\_post\_code1 lv\_city1  into ls\_adressdaten-plz\_ort separated by space.  \* Department + Telefonnummer holen aus ADCP  select tel\_number department from adcp  into ( ls\_adressdaten-tel\_number, ls\_adressdaten-department )  where addrnumber = l\_adr  and persnumber = l\_per  and date\_from < sy-datum.  endselect.  else.  message i040(zfi\_nb) with lv\_bname. "EMail-Adresse nicht gefunden  endif.  else.  message i036(zfi\_nb) with lv\_bname.  clear ls\_adressdaten.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  es\_adressdaten = ls\_adressdaten.  endmethod.  method get\_anrede.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Zuteilen der Anrede aufgrund Title  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_title type ad\_title,  lv\_langu type spras,  lv\_anrede type ad\_titletx,  lt\_tsad3t type table of tsad3t,  ls\_tsad3t like line of lt\_tsad3t.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_title = iv\_title.  lv\_langu = iv\_langu.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  \* Holen der Enträge für Personen  select title\_medi from tsad3t into lv\_anrede  where langu = iv\_langu  and title = lv\_title.  endselect.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  ev\_anrede = lv\_anrede.  endmethod.  method get\_arc\_doc\_id.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Holen Archiv\_Id für ein Objekt \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinitionen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lv\_bukrs type bukrs,  lv\_belnr type belnr\_d,  lv\_gjahr type gjahr,  lv\_sap\_object type saeanwdid,  lv\_object\_id type saeobjid,  lv\_archiv\_id type saearchivi,  lv\_arc\_doc\_id type saeardoid,  lv\_ar\_object type saeobjart,  ls\_toa01 type toa01.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* -----------------------------------------------------  \* Import-Parameter übernehmen  \* -----------------------------------------------------  lv\_bukrs = iv\_bukrs.  lv\_belnr = iv\_belnr.  lv\_gjahr = iv\_gjahr.  lv\_sap\_object = iv\_sap\_object.  \* Initialisierung  clear: lv\_object\_id,  lv\_archiv\_id,  lv\_arc\_doc\_id,  lv\_ar\_object.  \* Aufbau des Zugriffschlüssels  case lv\_sap\_object.  when 'BKPF'.  concatenate lv\_bukrs lv\_belnr lv\_gjahr into lv\_object\_id.  when 'VBRK'.  concatenate lv\_belnr '\*' into lv\_object\_id.  endcase.  \* Holen der Archiv\_id  select single \* from toa01 into corresponding fields of ls\_toa01  where sap\_object = lv\_sap\_object  and object\_id = lv\_object\_id.  if sy-subrc = 0.  lv\_archiv\_id = ls\_toa01-archiv\_id.  lv\_arc\_doc\_id = ls\_toa01-arc\_doc\_id.  lv\_ar\_object = ls\_toa01-ar\_object.  endif.  \*------------------------------------------------------  \* Export Parameter  \*------------------------------------------------------  ev\_archiv\_id = lv\_archiv\_id.  ev\_arc\_doc\_id = lv\_arc\_doc\_id.  ev\_ar\_object = lv\_ar\_object.  endmethod.  method get\_attachments.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* 27.02.2019 NIW: Korrektur Zugriff  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: ls\_bkpf type bkpf,  lt\_toaom type table of toaom,  ls\_toaom type toaom,  lt\_toa01 type table of toa01,  ls\_toa01 type toa01,  lt\_dispdocs type alink\_connection\_table,  ls\_dispdocs like line of lt\_dispdocs.  data: lv\_bukrs type bukrs,  lv\_belnr type belnr\_d,  lv\_gjahr type gjahr.  data: lv\_connection type saeverkn,  lv\_sap\_object type saeanwdid,  lv\_object\_id type saeobjid.  constants: co\_bkpf type saeanwdid value 'BKPF',  co\_bus2081 type saeanwdid value 'BUS2081',  co\_vbrk type saeanwdid value 'VBRK'.  field-symbols: <ls\_toa01> like ls\_toa01,  <ls\_toaom> like ls\_toaom.  \* Import-Parameter übernehmen  lv\_bukrs = iv\_bukrs.  lv\_belnr = iv\_belnr.  lv\_gjahr = iv\_gjahr.  \* holen awtyp und awkey aus bkpf für zugriff Verknüpfungstabelle  select single \* from bkpf into ls\_bkpf  where bukrs = lv\_bukrs  and belnr = lv\_belnr  and gjahr = lv\_gjahr.  if sy-subrc = 0.  case ls\_bkpf-awtyp.  when 'BKPF'. "normale FI-Belege  lv\_sap\_object = co\_bkpf.  lv\_object\_id = lv\_bukrs && lv\_belnr && lv\_gjahr.  when 'ZBKPF'. "Belege aus VT  lv\_sap\_object = co\_bkpf.  lv\_object\_id = lv\_bukrs && lv\_belnr && lv\_gjahr.  when 'BKPFF'.  lv\_sap\_object = co\_bkpf.  lv\_object\_id = lv\_bukrs && lv\_belnr && lv\_gjahr.  when 'RMRP'. "log.Rechnungsprüfung  lv\_sap\_object = co\_bus2081.  lv\_object\_id = ls\_bkpf-awkey.  when 'VBRK'. "Vetriebsbelege  lv\_sap\_object = co\_vbrk.  lv\_object\_id = ls\_bkpf-awkey && '\*'.  when others.  lv\_sap\_object = co\_bkpf.  lv\_object\_id = lv\_bukrs && lv\_belnr && lv\_gjahr.  endcase.  \* Holen Verknüpfungstabelle für den Beleg  select single \* from toaom into ls\_toaom  where sap\_object = lv\_sap\_object.  lv\_connection = ls\_toaom-connection.  \* Holen Verknüpfungsinfo für den FI-Beleg  select \* from (lv\_connection) into corresponding fields of table lt\_toa01  where sap\_object = lv\_sap\_object  and object\_id = lv\_object\_id.  if lt\_toa01[] is initial.  message i007(zfi\_alg).  else.  loop at lt\_toa01 assigning <ls\_toa01>.  clear ls\_dispdocs.  ls\_dispdocs-adid = <ls\_toa01>-arc\_doc\_id.  ls\_dispdocs-aid = <ls\_toa01>-archiv\_id.  ls\_dispdocs-oid = <ls\_toa01>-object\_id.  ls\_dispdocs-oti = <ls\_toa01>-sap\_object.  append ls\_dispdocs to lt\_dispdocs.  endloop.  endif.  endif.  \* Export-Parameter  et\_dispdocs[] = lt\_dispdocs[].  endmethod.  method get\_attachments\_2.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Dokumente holen  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: ls\_bkpf type bkpf,  lt\_connections type connection\_tab,  ls\_connections like line of lt\_connections.  data: lv\_bukrs type bukrs,  lv\_belnr type belnr\_d,  lv\_gjahr type gjahr.  data: lv\_sap\_object type saeanwdid,  lv\_object\_id type saeobjid.  constants: co\_bkpf type saeanwdid value 'BKPF',  co\_bus2081 type saeanwdid value 'BUS2081',  co\_vbrk type saeanwdid value 'VBRK'.  \* Import-Parameter übernehmen  lv\_bukrs = iv\_bukrs.  lv\_belnr = iv\_belnr.  lv\_gjahr = iv\_gjahr.  \* holen awtyp und awkey aus bkpf für zugriff Verknüpfungstabelle  select single \* from bkpf into ls\_bkpf  where bukrs = lv\_bukrs  and belnr = lv\_belnr  and gjahr = lv\_gjahr.  if sy-subrc = 0.  case ls\_bkpf-awtyp.  when 'BKPF'. "normale FI-Belege  lv\_sap\_object = co\_bkpf.  lv\_object\_id = lv\_bukrs && lv\_belnr && lv\_gjahr.  when 'ZBKPF'. "Belege aus VT  lv\_sap\_object = co\_bkpf.  lv\_object\_id = lv\_bukrs && lv\_belnr && lv\_gjahr.  when 'BKPFF'.  lv\_sap\_object = co\_bkpf.  lv\_object\_id = lv\_bukrs && lv\_belnr && lv\_gjahr.  when 'RMRP'. "log.Rechnungsprüfung  lv\_sap\_object = co\_bus2081.  lv\_object\_id = ls\_bkpf-awkey.  when 'VBRK'. "Vetriebsbelege  lv\_sap\_object = co\_vbrk.  lv\_object\_id = ls\_bkpf-awkey.  when others.  lv\_sap\_object = co\_bkpf.  lv\_object\_id = lv\_bukrs && lv\_belnr && lv\_gjahr.  endcase.  call function 'ARCHIV\_CONNECTINFO\_GET\_META'  exporting  object\_id = lv\_object\_id  sap\_object = lv\_sap\_object  tables  connect\_info = lt\_connections  exceptions  others = 1.  endif.  \* Export-Parameter  et\_connections[] = lt\_connections[].  endmethod.  method get\_belegaenderungen.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Holen Belegänderungen für Offene Forderungen:  \* Mahnstufe / Mahnsperre  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lt\_key type zfi\_t\_nb\_forderung\_key,  ls\_key type zfi\_s\_nb\_forderung\_key,  lt\_hist type zfi\_t\_nb\_ford\_hist,  ls\_hist type zfi\_s\_nb\_ford\_hist.  data: lv\_cdtabkey type cdtabkey,  lv\_cpudt type cpudt,  lv\_cputm type cputm,  lv\_usnam type usnam.  types: begin of bhist,  objectid type cdobjectcl,  username type cdusername,  udat type cddatum,  utime type cduzeit,  fname type fieldname,  value\_new type cdfldvaln,  value\_old type cdfldvalo,  end of bhist.  data: lt\_bhist type table of bhist,  ls\_bhist like line of lt\_bhist.  field-symbols: <ls\_bhist> like line of lt\_bhist.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_cdtabkey = '330' && is\_key-bukrs && is\_key-belnr && is\_key-gjahr && is\_key-buzei.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  \* Join über CDHDR und CDPOS  select h~objectid, h~username, h~udate, h~utime, p~fname, p~value\_new, p~value\_old  from ( cdpos as p  inner join cdhdr as h on p~objectclas = h~objectclas  and p~objectid = h~objectid  and p~changenr = h~changenr )  where p~tabname = 'BSEG'  and p~tabkey = @lv\_cdtabkey  and ( p~fname = 'MANST' or p~fname = 'MANSP' )  into table @lt\_bhist.  \* Tabelleneinträge übernehmen für History-Tabelle  loop at lt\_bhist assigning <ls\_bhist>.  clear ls\_hist.  \*  ls\_hist-bukrs = is\_key-bukrs.  ls\_hist-gjahr = is\_key-gjahr.  ls\_hist-belnr = is\_key-belnr.  ls\_hist-buzei = is\_key-buzei.  ls\_hist-datum = <ls\_bhist>-udat.  ls\_hist-uzeit = <ls\_bhist>-utime(2) && ':' && <ls\_bhist>-utime+2(2) &&':' && <ls\_bhist>-utime+4(2).  ls\_hist-bname = <ls\_bhist>-username.  case <ls\_bhist>-fname.  when 'MANST'.  ls\_hist-aktion = 'S'.  ls\_hist-bezeichnung = 'Mahnstufe'.  when 'MANSP'.  ls\_hist-aktion = 'M'.  ls\_hist-bezeichnung = 'Mahnsperre'.  endcase.  ls\_hist-feld1 = <ls\_bhist>-value\_new.  ls\_hist-feld1\_alt = <ls\_bhist>-value\_old.  \*  append ls\_hist to lt\_hist.  endloop.  \* Ausgleichsinfo holen  select k~cpudt, k~cputm, k~usnam  from ( bseg as p  inner join bkpf as k on k~bukrs = p~bukrs  and k~belnr = p~augbl  and k~gjahr = p~auggj )  where p~bukrs = @is\_key-bukrs  and p~belnr = @is\_key-belnr  and p~gjahr = @is\_key-gjahr  and p~buzei = @is\_key-buzei  into ( @lv\_cpudt, @lv\_cputm, @lv\_usnam ).  endselect.  \* Wenn Ausgleich vorhanden --> Eintrag in lt\_hist erstellen  if not lv\_cpudt is initial.  clear ls\_hist.  \*  ls\_hist-bukrs = is\_key-bukrs.  ls\_hist-gjahr = is\_key-gjahr.  ls\_hist-belnr = is\_key-belnr.  ls\_hist-buzei = is\_key-buzei.  ls\_hist-datum = lv\_cpudt.  ls\_hist-uzeit = lv\_cputm(2) && ':' && lv\_cputm+2(2) &&':' && lv\_cputm+4(2).  ls\_hist-bname = lv\_usnam.  ls\_hist-aktion = 'Z'.  ls\_hist-bezeichnung = 'Ausgleich'.  append ls\_hist to lt\_hist.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  et\_history[] = lt\_hist[].  endmethod.  method get\_bp\_customer\_data.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* es wird der BP geholt aufgrund der Kundennummer  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: ls\_but000 type but000.  data: lv\_bp type bu\_partner.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_bp = iv\_bp.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  select single but000~\*, cvi\_cust\_link~customer, kna1~\*, adrc~\*  from but000  left outer join cvi\_cust\_link  on cvi\_cust\_link~partner\_guid = but000~partner\_guid  left outer join kna1  on kna1~kunnr = cvi\_cust\_link~customer  left outer join adrc  on adrc~addrnumber = kna1~adrnr  where but000~partner = @lv\_bp  into @data(ls\_bp\_data).  if sy-subrc eq 0.  es\_but000 = ls\_bp\_data-but000.  es\_kna1 = ls\_bp\_data-kna1.  es\_adrc = ls\_bp\_data-adrc.  ev\_kunnr = ls\_bp\_data-customer.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  endmethod.  method get\_bp\_from\_kunnr.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* es wird der BP geholt aufgrund der Kundennummer  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: ls\_but000 type but000.  data: lv\_kunnr type kunnr,  lv\_partner\_guid type bu\_partner\_guid,  lv\_partner type bu\_partner.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_kunnr = iv\_kunnr.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  \* Holen Partner\_guid aufgrund Customer  select partner\_guid from cvi\_cust\_link into lv\_partner\_guid  where customer = lv\_kunnr.  endselect.  \* Holen Partner aufgrund der Partner\_guid  select partner from but000 into lv\_partner  where partner\_guid = lv\_partner\_guid.  endselect.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  ev\_bp = lv\_partner.  endmethod.  method get\_bp\_from\_lifnr.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* es wird der BP geholt aufgrund der Lieferantennummer  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: ls\_but000 type but000.  data: lv\_lifnr type lifnr,  lv\_partner\_guid type bu\_partner\_guid,  lv\_partner type bu\_partner.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_lifnr = iv\_lifnr.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  \* Holen Partner\_guid aufgrund Customer  select partner\_guid from cvi\_vend\_link into lv\_partner\_guid  where vendor = lv\_lifnr.  endselect.  \* Holen Partner aufgrund der Partner\_guid  select partner from but000 into lv\_partner  where partner\_guid = lv\_partner\_guid.  endselect.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  ev\_bp = lv\_partner.  endmethod.  method get\_bp\_vendor\_data.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* es wird der BP geholt aufgrund der Lieferantennummer  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: ls\_but000 type but000.  data: lv\_bp type bu\_partner.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_bp = iv\_bp.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  select single but000~\*, cvi\_vend\_link~vendor, lfa1~\*, adrc~\*  from but000  left outer join cvi\_vend\_link  on cvi\_vend\_link~partner\_guid = but000~partner\_guid  left outer join lfa1  on lfa1~lifnr = cvi\_vend\_link~vendor  left outer join adrc  on adrc~addrnumber = lfa1~adrnr  where but000~partner = @lv\_bp  into @data(ls\_bp\_data).  if sy-subrc eq 0.  es\_but000 = ls\_bp\_data-but000.  es\_lfa1 = ls\_bp\_data-lfa1.  es\_adrc = ls\_bp\_data-adrc.  ev\_lifnr = ls\_bp\_data-vendor.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  endmethod.  method GET\_DATA\_FROM\_ARCHIV.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Attachment Daten aus dem Archiv lesen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  DATA: lv\_archiv\_id TYPE saearchivi,  lv\_arc\_doc\_id TYPE saeardoid,  lv\_document\_type TYPE saedoktyp,  lv\_bin\_len TYPE num12,  lv\_len TYPE i.  DATA: lt\_bin TYPE TABLE OF tbl1024.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Export \*  \*==========================================\*  CLEAR: ev\_data\_xstring.  \*==========================================\*  \* Import \*  \*==========================================\*  lv\_archiv\_id = iv\_archiv\_id.  lv\_arc\_doc\_id = iv\_arc\_doc\_id.  lv\_document\_type = iv\_document\_type.  \*==========================================\*  \* Binäre Daten aus dem Archiv \*  \*==========================================\*  CLEAR: lv\_bin\_len.  REFRESH: lt\_bin.  CALL FUNCTION 'ARCHIVOBJECT\_GET\_TABLE'  EXPORTING  archiv\_id = lv\_archiv\_id  document\_type = lv\_document\_type  archiv\_doc\_id = lv\_arc\_doc\_id  IMPORTING  binlength = lv\_bin\_len  TABLES  binarchivobject = lt\_bin  EXCEPTIONS  error\_archiv = 1  error\_communicationtable = 2  error\_kernel = 3  OTHERS = 4.  IF sy-subrc NE 0.  CLEAR: ev\_data\_xstring.  RETURN.  ENDIF.  \*==========================================\*  \* Binäre Daten in XSTRING umwandeln \*  \*==========================================\*  lv\_len = lv\_bin\_len.  CALL FUNCTION 'SCMS\_BINARY\_TO\_XSTRING'  EXPORTING  input\_length = lv\_len  IMPORTING  buffer = ev\_data\_xstring  TABLES  binary\_tab = lt\_bin.  endmethod.  method get\_erfasser\_dr.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* es  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_bukrs type bukrs,  lv\_belnr type belnr\_d,  lv\_gjahr type gjahr,  lv\_bname type usnam,  ls\_adressdaten type zfi\_s\_nb\_adressdaten.  data: ls\_head type ztsd\_iv\_head,  lv\_usnam type uname,  lv\_ppnam type ppnam.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_bukrs = iv\_bukrs.  lv\_belnr = iv\_belnr.  lv\_gjahr = iv\_gjahr.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  clear lv\_bname.  \*\*\* Holen des Vorerfassers und Erfasser aus der Tabelle BKPF  select usnam ppnam from bkpf into (lv\_usnam, lv\_ppnam)  where bukrs = iv\_bukrs  and belnr = iv\_belnr  and gjahr = iv\_gjahr.  endselect.  if not lv\_ppnam is initial.  lv\_bname = lv\_ppnam.  else.  lv\_bname = lv\_usnam.  endif.  \* restliche Informationen des Erfassers zusammenstellen  call method zficl\_nb=>get\_adressdaten  exporting  iv\_bname = lv\_bname  importing  es\_adressdaten = ls\_adressdaten.  \*  ls\_adressdaten-vorerfasser = lv\_ppnam.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  es\_adressdaten = ls\_adressdaten.  endmethod.  method get\_erfasser\_forderung\_cdhdr.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Holen Erfasser aus Aenderungsbelegen gem RVSCD100  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_vbelv type vbeln\_von,  lv\_posnr type posnr\_von,  lv\_bname type usnam.  data: lt\_cdhdr type table of cdhdr,  lt\_cdpos type table of cdpos,  lv\_changenr type cdchangenr,  lv\_tabname type tabname,  lv\_tabkey type cdtabkey,  lv\_fname type fieldname,  lv\_username type cdusername.  constants: co\_objectclas type cdobjectcl value 'VERKBELEG',  co\_tabname type tabname value 'VBAP',  co\_fname type fieldname value 'FAKSP'.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_vbelv = iv\_vbelv.  lv\_posnr = iv\_posnv.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  \* Aufbau Tabellenkey  concatenate sy-mandt lv\_vbelv lv\_posnr into lv\_tabkey.  \* holen des entsprechenden Eintrages aus CDPOS für Changenr  select changenr from cdpos into lv\_changenr  where objectclas = co\_objectclas  and objectid = lv\_vbelv  and tabname = co\_tabname  and tabkey = lv\_tabkey  and fname = co\_fname.  endselect.  if not lv\_changenr is initial.  \* Holen des Kopfeintrages mit Username aufgrund changenr  select username from cdhdr into lv\_username  where objectclas = co\_objectclas  and objectid = lv\_vbelv  and changenr = lv\_changenr.  endselect.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  ev\_bname = lv\_username.  endmethod.  method get\_erfasser\_forderung\_fi.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Holen der Erfasser und Bucherdaten für Forderungen aus FI  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_bukrs type bukrs,  lv\_belnr type belnr\_d,  lv\_gjahr type gjahr,  lv\_bname type usnam,  ls\_adressdaten type zfi\_s\_nb\_adressdaten.  data: lv\_usnam type uname,  lv\_ppnam type ppnam.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_bukrs = iv\_bukrs.  lv\_belnr = iv\_belnr.  lv\_gjahr = iv\_gjahr.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  clear lv\_bname.  \*\*\* Holen des Vorerfassers und Erfasser aus der Tabelle BKPF  select usnam ppnam from bkpf into (lv\_usnam, lv\_ppnam)  where bukrs = iv\_bukrs  and belnr = iv\_belnr  and gjahr = iv\_gjahr.  endselect.  if not lv\_ppnam is initial.  lv\_bname = lv\_ppnam.  else.  lv\_bname = lv\_usnam.  endif.  \* restliche Informationen des Erfassers zusammenstellen  call method zficl\_nb=>get\_adressdaten  exporting  iv\_bname = lv\_bname  importing  es\_adressdaten = ls\_adressdaten.  \*  ls\_adressdaten-vorerfasser = lv\_ppnam.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  es\_adressdaten = ls\_adressdaten.  endmethod.  method get\_erfasser\_forderung\_sd.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Holen der Erfasser und Bucherdaten für Forderungen aus SD  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_bukrs type bukrs,  lv\_belnr type belnr\_d,  lv\_gjahr type gjahr,  lv\_bname type usnam,  ls\_adressdaten type zfi\_s\_nb\_adressdaten.  types: begin of sd,  ernam type ernam,  vkorg type vkorg,  vkbur type vkbur,  end of sd.  data: ls\_sd type sd.  data: lv\_vbeln type vbeln\_va,  lv\_vbelv type vbeln\_von,  lv\_posnv type posnr\_von.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_bukrs = iv\_bukrs.  lv\_belnr = iv\_belnr.  lv\_gjahr = iv\_gjahr.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  clear ls\_sd.  \*\*\* Holen VBELV aus VBFA für Zugriff auf VBAK  select vbelv posnv from vbfa into ( lv\_vbelv, lv\_posnv )  where vbeln = lv\_belnr.  endselect.  \*\*\* Zugriff auf VBAK  if not lv\_vbelv is initial.  select \* from vbak into corresponding fields of ls\_sd  where vbeln = lv\_vbelv.  endselect.  if not ls\_sd is initial.  \* Verarbeitung gemäss VKORG  case ls\_sd-vkorg.  when 'LW00' or 'FW00' or 'GW00'. "Liegenschaften, Fernwärme, Gastwohnungen  call method zficl\_nb=>get\_adressdaten  exporting  iv\_bname = ls\_sd-ernam  importing  es\_adressdaten = ls\_adressdaten.  \*  ls\_adressdaten-vkbur = ls\_sd-vkbur.  ls\_adressdaten-vorerfasser = ls\_sd-ernam.  when 'FM00'. "Forderungsmanagement  call method zficl\_nb=>get\_erfasser\_forderung\_cdhdr  exporting  iv\_vbelv = lv\_vbelv  iv\_posnv = lv\_posnv  importing  ev\_bname = lv\_bname.  call method zficl\_nb=>get\_adressdaten  exporting  iv\_bname = lv\_bname  importing  es\_adressdaten = ls\_adressdaten.  \*  ls\_adressdaten-vkbur = ls\_sd-vkbur.  ls\_adressdaten-vorerfasser = lv\_bname.  when others.  clear ls\_adressdaten.  endcase.  endif.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  es\_adressdaten = ls\_adressdaten.  endmethod.  method get\_erfasser\_forderung\_vt.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Holen Adressdaten Erfasser und Buchung VT, FT  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_bukrs type bukrs,  lv\_belnr type belnr\_d,  lv\_gjahr type gjahr,  lv\_bname type usnam,  ls\_adressdaten type zfi\_s\_nb\_adressdaten.  data: ls\_head type ztsd\_iv\_head,  lv\_nfirst type ad\_namefir,  lv\_nlast type ad\_namelas,  lv\_mail type ad\_smtpadr.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_bukrs = iv\_bukrs.  lv\_belnr = iv\_belnr.  lv\_gjahr = iv\_gjahr.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  clear lv\_bname.  \* Holen des Erfassers aut ZTSD\_IV\_HEAD  select \* from ztsd\_iv\_head into corresponding fields of ls\_head  where bukrs = lv\_bukrs  and belnr = lv\_belnr  and gjahr = lv\_gjahr.  if sy-subrc = 0. "Daten übernehmen  case ls\_head-vkbur.  when 'AD01' or 'AD02'. "Studenten aus LISETH  ls\_adressdaten-bname = ls\_head-bucher.  concatenate ls\_head-name\_first ls\_head-name\_last into ls\_adressdaten-ename  separated by space.  ls\_adressdaten-email = ls\_head-smtp\_addr.  ls\_adressdaten-student\_nr = ls\_head-nebis\_nr.  \* Vorerfasser holen aus ZVR\_SS\_LIS\_ADR mit NAME\_FIRST; NAME\_LAST  select bname from zvr\_ss\_lis\_adr into ls\_adressdaten-vorerfasser  where name\_first = ls\_head-name\_first  and name\_last = ls\_head-name\_last.  endselect.  when 'AD03'. "Exkursionen  select bname name\_first name\_last smtp\_addr from zvr\_ss\_lis\_adr  into ( lv\_bname, lv\_nfirst, lv\_nlast, lv\_mail )  where kontakt = 'AREX'.  endselect.  ls\_adressdaten-vorerfasser = lv\_bname.  concatenate lv\_nfirst lv\_nlast into ls\_adressdaten-ename separated by space.  ls\_adressdaten-email = lv\_mail.  when 'AD04'. "D-CHAB  select bname name\_first name\_last smtp\_addr from zvr\_ss\_lis\_adr  into ( lv\_bname, lv\_nfirst, lv\_nlast, lv\_mail )  where kontakt = 'CHAB'.  endselect.  ls\_adressdaten-vorerfasser = lv\_bname.  concatenate lv\_nfirst lv\_nlast into ls\_adressdaten-ename separated by space.  ls\_adressdaten-email = lv\_mail.  when 'AD05'. "D-ARCH Benutzung Plotter, etc.  when 'AD06'. "SCE  select bname name\_first name\_last smtp\_addr from zvr\_ss\_lis\_adr  into ( lv\_bname, lv\_nfirst, lv\_nlast, lv\_mail )  where kontakt = 'SCE '.  endselect.  ls\_adressdaten-vorerfasser = lv\_bname.  concatenate lv\_nfirst lv\_nlast into ls\_adressdaten-ename separated by space.  ls\_adressdaten-email = lv\_mail.  when 'RW02'. "Daylight  \* Vorerfasser holen aus ZVR\_SS\_KW\_C\_VER mit VeranstalterID  select bname name\_first name\_last smtp\_addr from zvr\_ss\_kw\_c\_ver  into ( lv\_bname, lv\_nfirst, lv\_nlast, lv\_mail )  where veranstalterid = ls\_head-veranstalterid.  endselect.  ls\_adressdaten-vorerfasser = lv\_bname.  concatenate lv\_nfirst lv\_nlast into ls\_adressdaten-ename separated by space.  ls\_adressdaten-email = lv\_mail.  when 'RW03'. "Excel\_upload  select contact from zvr\_book\_excel into lv\_bname  where book\_belnr = lv\_belnr  and book\_gjahr = lv\_gjahr.  endselect.  translate lv\_bname to upper case.  call method zficl\_nb=>get\_adressdaten  exporting  iv\_bname = lv\_bname  importing  es\_adressdaten = ls\_adressdaten.  ls\_adressdaten-vorerfasser = ls\_adressdaten-bname.  ls\_adressdaten-email = ls\_adressdaten-email.  when 'RK01' or 'RK02' or 'RK03'.  lv\_bname = ls\_head-bucher.  call method zficl\_nb=>get\_adressdaten  exporting  iv\_bname = lv\_bname  importing  es\_adressdaten = ls\_adressdaten.  ls\_adressdaten-vorerfasser = ls\_head-bucher.  ls\_adressdaten-email = ls\_adressdaten-email.  when 'IS10'. "IT-Shop  \*  when others.  if not ls\_head-vorerfasser is initial.  lv\_bname = ls\_head-vorerfasser.  \*  call method zficl\_nb=>get\_adressdaten  exporting  iv\_bname = lv\_bname  importing  es\_adressdaten = ls\_adressdaten.  ls\_adressdaten-vorerfasser = ls\_head-vorerfasser.  ls\_adressdaten-email = ls\_adressdaten-email.  \* Bucher richtigstellen  ls\_adressdaten-bname = ls\_head-bucher.  else.  lv\_bname = ls\_head-bucher.  \*  call method zficl\_nb=>get\_adressdaten  exporting  iv\_bname = lv\_bname  importing  es\_adressdaten = ls\_adressdaten.  endif.  endcase.  endif.  endselect.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  es\_adressdaten = ls\_adressdaten.  endmethod.  method get\_erfasser\_rd.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* es  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_bukrs type bukrs,  lv\_belnr type belnr\_d,  lv\_gjahr type gjahr,  lv\_bname type usnam,  lv\_projk type ps\_psp\_pnr,  ls\_adressdaten type zfi\_s\_nb\_adressdaten.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_bukrs = iv\_bukrs.  lv\_belnr = iv\_belnr.  lv\_gjahr = iv\_gjahr.  lv\_projk = iv\_projk.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  call method zficl\_nb=>get\_erfasser\_rd\_sd  exporting  iv\_vbeln = lv\_belnr  iv\_projk = lv\_projk  importing  es\_adressdaten = ls\_adressdaten.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  es\_adressdaten = ls\_adressdaten.  endmethod.  method get\_erfasser\_rd\_sd.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* es  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  types: begin of sd,  ernam type ernam,  vkorg type vkorg,  zzsender\_name type zfmt\_e\_sender\_name,  zzemail type zfmt\_e\_email,  end of sd.  data: ls\_sd type sd.  data: lv\_vbeln type vbeln\_va,  lv\_projk type ps\_psp\_pnr,  lv\_bname type usnam,  lv\_vbelv type vbeln\_von.  data: ls\_adressdaten type zfi\_s\_nb\_adressdaten,  ls\_adressdaten\_fm type zfmt\_s\_sender.  data: ls\_head type ztsd\_iv\_head,  lv\_usnam type uname,  lv\_ppnam type ppnam.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_vbeln = iv\_vbeln.  lv\_projk = iv\_projk.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  clear ls\_sd.  \*\*\* Holen VBELV aus VBFA für Zugriff auf VBAK  select vbelv from vbfa into lv\_vbelv  where vbeln = lv\_vbeln.  endselect.  \*\*\* Zugriff auf VBAK  if not lv\_vbelv is initial.  select \* from vbak into corresponding fields of ls\_sd  where vbeln = lv\_vbelv.  endselect.  if not ls\_sd is initial.  \* Verarbeitung gemäss VKORG  case ls\_sd-vkorg.  when 'LW00' or 'FW00' or 'GW00'. "Liegenschaften, Fernwärme, Gastwohnungen  call method zficl\_nb=>get\_adressdaten  exporting  iv\_bname = ls\_sd-ernam  importing  es\_adressdaten = ls\_adressdaten.  \*  ls\_adressdaten-vkbur = ls\_sd-vkorg.  when 'FM00'. "Forderungsmanagement  \* Holen Adressdaten gem FM  call method zfmtcl\_pay\_plan=>get\_sender\_by\_pspnr  exporting  iv\_pspnr = lv\_projk  importing  es\_sender = ls\_adressdaten\_fm.  \*  ls\_adressdaten-bname = ls\_sd-ernam.  ls\_adressdaten-ename = ls\_adressdaten\_fm-zzsender\_name.  ls\_adressdaten-email = ls\_adressdaten\_fm-zzemail.  ls\_adressdaten-vkbur = ls\_sd-vkorg.  when others.  endcase.  endif.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  es\_adressdaten = ls\_adressdaten.  endmethod.  method get\_erfasser\_vd.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* es  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_bukrs type bukrs,  lv\_belnr type belnr\_d,  lv\_gjahr type gjahr,  lv\_bname type usnam,  ls\_adressdaten type zfi\_s\_nb\_adressdaten.  data: ls\_head type ztsd\_iv\_head.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_bukrs = iv\_bukrs.  lv\_belnr = iv\_belnr.  lv\_gjahr = iv\_gjahr.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  clear lv\_bname.  \* Holen des Erfassers aut ZTSD\_IV\_HEAD  select \* from ztsd\_iv\_head into corresponding fields of ls\_head  where bukrs = lv\_bukrs  and belnr = lv\_belnr  and gjahr = lv\_gjahr.  if sy-subrc = 0. "Daten übernehmen  case ls\_head-vkbur.  when 'AD01' or 'AD02' or 'AD06'.  ls\_adressdaten-bname = ls\_head-bucher.  concatenate ls\_head-name\_first ls\_head-name\_last into ls\_adressdaten-ename  separated by space.  ls\_adressdaten-email = ls\_head-smtp\_addr.  ls\_adressdaten-student\_nr = ls\_head-nebis\_nr.  when others.  if not ls\_head-vorerfasser is initial.  lv\_bname = ls\_head-vorerfasser.  \*  call method zficl\_nb=>get\_adressdaten  exporting  iv\_bname = lv\_bname  importing  es\_adressdaten = ls\_adressdaten.  ls\_adressdaten-vorerfasser = ls\_head-vorerfasser.  \* Bucher richtigstellen  ls\_adressdaten-bname = ls\_head-bucher.  else.  lv\_bname = ls\_head-bucher.  \*  call method zficl\_nb=>get\_adressdaten  exporting  iv\_bname = lv\_bname  importing  es\_adressdaten = ls\_adressdaten.  endif.  endcase.  endif.  endselect.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  es\_adressdaten = ls\_adressdaten.  endmethod.  METHOD get\_forderungen.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Es werden die Debitoren Mahndaten geholt  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  DATA: lt\_ford TYPE zfi\_t\_nb\_forderungen,  lt\_ford\_save TYPE zfi\_t\_nb\_forderungen,  ls\_ford TYPE zfi\_s\_nb\_forderungen,  ls\_ford\_save TYPE zfi\_s\_nb\_forderungen,  lt\_bsid TYPE zfi\_t\_nb\_bsid,  ls\_bsid TYPE zfi\_s\_nb\_bsid,  lt\_t052 TYPE TABLE OF t052.  DATA: lt\_zusatz TYPE TABLE OF zfi\_nb\_ford\_zus,  ls\_zusatz TYPE zfi\_nb\_ford\_zus,  lt\_hist TYPE TABLE OF zfi\_nb\_ford\_hist,  ls\_hist TYPE zfi\_nb\_ford\_hist,  lt\_inb\_e TYPE zfi\_t\_nb\_ford\_inbox\_ethis,  ls\_inb\_e TYPE zfi\_s\_nb\_ford\_inbox\_ethis,  lt\_inb\_s TYPE zfi\_t\_nb\_ford\_inbox\_sapgui,  ls\_inb\_s TYPE zfi\_s\_nb\_ford\_inbox\_sapgui,  ls\_key TYPE zfi\_s\_nb\_forderung\_key.  TYPES: BEGIN OF bseg,  bukrs TYPE bukrs,  belnr TYPE belnr\_d,  gjahr TYPE gjahr,  hkont TYPE hkont,  projk TYPE ps\_psp\_pnr,  kostl TYPE kostl,  sgtxt TYPE sgtxt,  END OF bseg.  DATA: lt\_bseg TYPE TABLE OF bseg,  ls\_bseg LIKE LINE OF lt\_bseg.  DATA: lt\_url TYPE table\_of\_strings.  DATA: lv\_bname TYPE uname,  ls\_adressdaten TYPE zfi\_s\_nb\_adressdaten,  ls\_adrc TYPE adrc,  lv\_anrede TYPE ad\_titletx.  DATA: l\_adr TYPE ad\_addrnum,  l\_per TYPE ad\_persnum,  lv\_street TYPE ad\_street,  lv\_house\_num1 TYPE ad\_hsnm1,  lv\_post\_code1 TYPE ad\_pstcd1,  lv\_city1 TYPE ad\_city1.  DATA: lv\_subrc TYPE subrc.  CONSTANTS: co\_bukrs\_0330 TYPE bukrs VALUE '0330',  co\_mandt\_330 TYPE mandt VALUE '330',  co\_usnam\_jobadm TYPE uname VALUE 'JOBADM'.  FIELD-SYMBOLS: <ls\_bsid> TYPE zfi\_s\_nb\_bsid,  <ls\_ford> TYPE zfi\_s\_nb\_forderungen,  <ls\_t052> TYPE t052,  <ls\_bseg> TYPE bseg,  <ls\_zus> TYPE zfi\_nb\_ford\_zus.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  \*-----------------------------------------------------------------------  \* Tabellen übernehmen --------------------------------------------------  \*-----------------------------------------------------------------------  SELECT \* FROM t052 INTO TABLE lt\_t052.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  \* 1. Daten aufgrund OP-Debitoren holen  \* 1. A: ZFIINST Debitoren OP mit Mahnsperre I übernehmen  SELECT bukrs kunnr gjahr belnr buzei budat bldat cpudt waers  xblnr blart shkzg dmbtr wrbtr mansp zterm zfbdt mwskz  FROM bsid INTO CORRESPONDING FIELDS OF TABLE lt\_bsid  WHERE bukrs = co\_bukrs\_0330  AND belnr IN ir\_belnr  AND gjahr IN ir\_gjahr  AND bschl = '01'  AND mansp = 'I'.  \* 1. B: ZFIUNFO: Debitoren OP mit Mahnsperre 'G' übernehmen  SELECT bukrs kunnr gjahr belnr buzei budat bldat cpudt waers  xblnr blart shkzg dmbtr wrbtr mansp zterm zfbdt mwskz  FROM bsid APPENDING CORRESPONDING FIELDS OF TABLE lt\_bsid  WHERE belnr IN ir\_belnr  AND gjahr IN ir\_gjahr  AND bukrs = co\_bukrs\_0330  AND ( blart = 'RD' OR blart = 'VD' )  AND bschl = '01'  AND mansp = 'G'.  \* 1. C: OP's in Betreibung übernehmem (mit Mahnsperre 'B' )  SELECT bukrs kunnr gjahr belnr buzei budat bldat cpudt waers  xblnr blart shkzg dmbtr wrbtr mansp zterm zfbdt mwskz  FROM bsid APPENDING CORRESPONDING FIELDS OF TABLE lt\_bsid  WHERE belnr IN ir\_belnr  AND gjahr IN ir\_gjahr  AND bukrs = co\_bukrs\_0330  \* and ( blart = 'RD' or blart = 'VD' )  AND bschl = '01'  AND mansp = 'B'.  \* 1. C: OP's in Inkasso übernehmem (mit Mahnsperre 'K' )  SELECT bukrs kunnr gjahr belnr buzei budat bldat cpudt waers  xblnr blart shkzg dmbtr wrbtr mansp zterm zfbdt mwskz  FROM bsid APPENDING CORRESPONDING FIELDS OF TABLE lt\_bsid  WHERE belnr IN ir\_belnr  AND gjahr IN ir\_gjahr  AND bukrs = co\_bukrs\_0330  \* and ( blart = 'RD' or blart = 'VD' )  AND bschl = '01'  AND mansp = 'K'.  \* 2. Sortieren und löschen von doppelten Einträgen  SORT lt\_bsid.  DELETE ADJACENT DUPLICATES FROM lt\_bsid.  \*\*\* AB HIER kann ersetzt werden mit Methode  \* Start mit loop at lt\_bsid  \* Gegenpositionen aus BSEG holen und ablegen und komprimieren  SELECT \* FROM bseg INTO CORRESPONDING FIELDS OF TABLE lt\_bseg  FOR ALL ENTRIES IN lt\_bsid  WHERE bukrs = lt\_bsid-bukrs  AND belnr = lt\_bsid-belnr  AND gjahr = lt\_bsid-gjahr  AND koart = 'S'  AND hkont NE '0020001001'.  \* sortieren  SORT lt\_bseg BY bukrs belnr gjahr hkont projk kostl sgtxt.  \* 1. Komprimierung auf Einträge ohne Anpassungen  DELETE ADJACENT DUPLICATES FROM lt\_bseg  COMPARING bukrs belnr gjahr hkont projk kostl sgtxt.  \* 2. Komprimierung  \* Komprimierung z.b. Ad01.,,, noch offen  \*\*\*\*  \* Ergänzen der Daten mit Belegdatei (HKONT/SHKZG/SGTXT)  LOOP AT lt\_bsid ASSIGNING <ls\_bsid>.  \*  MOVE-CORRESPONDING <ls\_bsid> TO ls\_ford.  READ TABLE lt\_bseg INTO ls\_bseg  WITH KEY bukrs = <ls\_bsid>-bukrs  belnr = <ls\_bsid>-belnr  gjahr = <ls\_bsid>-gjahr.  IF sy-subrc = 0.  MOVE ls\_bseg-hkont TO ls\_ford-hkont.  MOVE ls\_bseg-projk TO ls\_ford-projk.  MOVE ls\_bseg-sgtxt TO ls\_ford-sgtxt.  ENDIF.  \* Zuteilen Name PSP-Element  IF NOT ls\_ford-projk IS INITIAL.  SELECT post1 FROM prps INTO ls\_ford-projk\_name  WHERE pspnr = ls\_ford-projk.  ENDSELECT.  ENDIF.  \*  APPEND ls\_ford TO lt\_ford.  ENDLOOP.  \*  \* sortieren der internen Tabelle vor delete adjacent  SORT lt\_ford BY bukrs gjahr belnr.  \* Doppelte Zeilen löschen  DELETE ADJACENT DUPLICATES FROM lt\_ford COMPARING gjahr belnr.  \*\*\* Bis hier Umstellung  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Selektionstabelle leer, Fehlermeldung  IF lt\_ford[] IS INITIAL.  MESSAGE e171(zfi\_nb). "keine offenen Forderungen gefunden  EXIT.  ENDIF.  \* Sortieren der Tabelle  SORT lt\_ford BY mansp gjahr belnr.  \* ergänzen der Forderungen mit weiteren Daten  \* Wertfelder richtigstellen:  LOOP AT lt\_ford ASSIGNING <ls\_ford>.  IF <ls\_ford>-shkzg = 'H'.  <ls\_ford>-wrbtr = <ls\_ford>-wrbtr \* -1.  ENDIF.  \* Name des Kunden holen  IF <ls\_ford>-kunnr NE space.  SELECT name1 xcpdk spras FROM kna1  INTO (<ls\_ford>-name1, <ls\_ford>-xcpdk, <ls\_ford>-spras)  WHERE kunnr = <ls\_ford>-kunnr.  ENDSELECT.  IF <ls\_ford>-xcpdk = 'X'.  SELECT name1 FROM bsec INTO  (<ls\_ford>-name1)  WHERE bukrs = <ls\_ford>-bukrs  AND belnr = <ls\_ford>-belnr  AND gjahr = <ls\_ford>-gjahr  AND buzei = <ls\_ford>-buzei.  ENDSELECT.  IF sy-subrc <> 0.  CLEAR: <ls\_ford>-name1, <ls\_ford>-spras.  ENDIF.  ENDIF.  ENDIF.  \* BP holen aufgrund Customer  CALL METHOD zficl\_nb=>get\_bp\_from\_kunnr  EXPORTING  iv\_kunnr = <ls\_ford>-kunnr  IMPORTING  ev\_bp = <ls\_ford>-bp.  \*  \* weitere Daten holen  CLEAR ls\_adressdaten.  \*  CASE <ls\_ford>-blart.  WHEN 'RD'. "Forderung SD (VBAK)  CALL METHOD zficl\_nb=>get\_erfasser\_forderung\_sd  EXPORTING  iv\_bukrs = <ls\_ford>-bukrs  iv\_belnr = <ls\_ford>-belnr  iv\_gjahr = <ls\_ford>-gjahr  IMPORTING  es\_adressdaten = ls\_adressdaten.  WHEN 'VD'. "Forderung VT (ztsd\_iv\_head)  CALL METHOD zficl\_nb=>get\_erfasser\_forderung\_vt  EXPORTING  iv\_bukrs = <ls\_ford>-bukrs  iv\_belnr = <ls\_ford>-belnr  iv\_gjahr = <ls\_ford>-gjahr  IMPORTING  es\_adressdaten = ls\_adressdaten.  WHEN OTHERS. "Forderung FI (BKPF)  CALL METHOD zficl\_nb=>get\_erfasser\_forderung\_fi  EXPORTING  iv\_bukrs = <ls\_ford>-bukrs  iv\_belnr = <ls\_ford>-belnr  iv\_gjahr = <ls\_ford>-gjahr  IMPORTING  es\_adressdaten = ls\_adressdaten.  ENDCASE.  \* weitere Adressdaten zuteilen  <ls\_ford>-bucher = ls\_adressdaten-bname.  <ls\_ford>-erfasser = ls\_adressdaten-vorerfasser.  <ls\_ford>-ename = ls\_adressdaten-ename.  <ls\_ford>-mail = ls\_adressdaten-email.  \* Zuteilen VBKUR/VKORG  CASE <ls\_ford>-blart.  WHEN 'VD'.  <ls\_ford>-vkbur = <ls\_ford>-xblnr.  WHEN 'RD'.  <ls\_ford>-vkbur = ls\_adressdaten-vkbur.  ENDCASE.  \* Zuteilen Forderungstyp  CALL METHOD zficl\_nb=>get\_forderungstyp  EXPORTING  iv\_mansp = <ls\_ford>-mansp  iv\_blart = <ls\_ford>-blart  iv\_vkbur = <ls\_ford>-vkbur  IMPORTING  ev\_typ = <ls\_ford>-ftype.  \* Fälligkeit berechnen und überfällige Tage  READ TABLE lt\_t052 ASSIGNING <ls\_t052>  WITH KEY zterm = <ls\_ford>-zterm.  IF sy-subrc = 0.  CASE <ls\_ford>-ftype.  WHEN 'A' OR 'B'.  <ls\_ford>-faedt = <ls\_ford>-bldat + <ls\_t052>-ztag1.  <ls\_ford>-akfdt = <ls\_ford>-faedt.  <ls\_ford>-utage = sy-datum - <ls\_ford>-faedt.  WHEN 'C' OR 'D'.  <ls\_ford>-faedt = <ls\_ford>-bldat + <ls\_t052>-ztag1.  <ls\_ford>-akfdt = <ls\_ford>-faedt + 10.  <ls\_ford>-utage = sy-datum - <ls\_ford>-faedt.  ENDCASE.  ELSE.  CASE <ls\_ford>-ftype.  WHEN 'A' OR 'B'.  <ls\_ford>-faedt = <ls\_ford>-bldat.  <ls\_ford>-akfdt = <ls\_ford>-faedt.  <ls\_ford>-utage = sy-datum - <ls\_ford>-faedt.  WHEN 'C' OR 'D'.  <ls\_ford>-faedt = <ls\_ford>-bldat.  <ls\_ford>-akfdt = <ls\_ford>-faedt + 10.  <ls\_ford>-utage = sy-datum - <ls\_ford>-faedt.  ENDCASE.  ENDIF.  \* Holen der Daten aus der Zusatztabelle  SELECT \* FROM zfi\_nb\_ford\_zus INTO CORRESPONDING FIELDS OF ls\_zusatz  WHERE bukrs = <ls\_ford>-bukrs  AND gjahr = <ls\_ford>-gjahr  AND belnr = <ls\_ford>-belnr  AND buzei = <ls\_ford>-buzei.  ENDSELECT.  IF sy-subrc = 0.  IF NOT ls\_zusatz-bemerkung IS INITIAL.  <ls\_ford>-bemerkung = ls\_zusatz-bemerkung.  ENDIF.  IF NOT ls\_zusatz-erfasser IS INITIAL.  <ls\_ford>-erfasser = ls\_zusatz-erfasser.  <ls\_ford>-ename = ls\_zusatz-ename.  <ls\_ford>-mail = ls\_zusatz-mail.  ENDIF.  \* Prüfen, ob Erfasser noch aktiv  CALL METHOD zficl\_nb=>check\_user  EXPORTING  iv\_uname = <ls\_ford>-erfasser  IMPORTING  ev\_inaktiv = <ls\_ford>-inaktiv.  IF NOT ls\_zusatz-flg\_faellig IS INITIAL.  <ls\_ford>-flg\_faellig = ls\_zusatz-flg\_faellig.  ENDIF.  IF NOT ls\_zusatz-flg\_infomail IS INITIAL.  <ls\_ford>-flg\_infomail = ls\_zusatz-flg\_infomail.  ENDIF.  IF NOT ls\_zusatz-mail\_datum IS INITIAL.  <ls\_ford>-mail\_datum = ls\_zusatz-mail\_datum.  ENDIF.  IF NOT ls\_zusatz-flg\_ausbuchen IS INITIAL.  <ls\_ford>-flg\_ausbuchen = ls\_zusatz-flg\_ausbuchen.  ENDIF.  IF NOT ls\_zusatz-flg\_betreiben IS INITIAL  AND <ls\_ford>-mansp = 'I'.  <ls\_ford>-flg\_betreiben = ls\_zusatz-flg\_betreiben.  ENDIF.  IF NOT ls\_zusatz-vldat IS INITIAL.  <ls\_ford>-vldat = ls\_zusatz-vldat.  ENDIF.  IF NOT ls\_zusatz-vl\_zaehler IS INITIAL.  <ls\_ford>-vl\_zaehler = ls\_zusatz-vl\_zaehler.  ENDIF.  \* aktuelles Fälligkeitsdatum setzen  IF NOT <ls\_ford>-vldat IS INITIAL.  <ls\_ford>-akfdt = <ls\_ford>-vldat.  <ls\_ford>-utage = sy-datum - <ls\_ford>-faedt.  ELSE.  <ls\_ford>-akfdt = <ls\_ford>-akfdt.  ENDIF.  \*  ELSE.  \* Zusatztabelle: neuen Eintrag initial ablegen  \* Aufbau Schlüssel  CLEAR ls\_key.  ls\_key-bukrs = <ls\_ford>-bukrs.  ls\_key-belnr = <ls\_ford>-belnr.  ls\_key-gjahr = <ls\_ford>-gjahr.  ls\_key-buzei = <ls\_ford>-buzei.  CALL METHOD zficl\_nb=>create\_forderung\_aktion  EXPORTING  is\_forderung\_key = ls\_key  iv\_aenderer = co\_usnam\_jobadm  iv\_aktion = 'N'  IMPORTING  ev\_subrc = lv\_subrc.  ENDIF.  ENDLOOP.  \*\* Aufbereiten der Tabellenausgabe aufgrund Parameter  lt\_ford\_save[] = lt\_ford[].  \* nur relevanten Benutzer oder alle nehmen  IF NOT iv\_bname IS INITIAL.  DELETE lt\_ford  WHERE erfasser NE iv\_bname.  ENDIF.  \* Datenselektion aufgrund geliefertem Parameter  CLEAR: ls\_inb\_e, ls\_inb\_s.  CASE iv\_param.  \*\* für ETHIS - INBOX Aufbau der Tabelle LT\_INB\_E  WHEN 'PBETU'. "ETHIS Inbox mit NAME + Inaktiv = ' '  LOOP AT lt\_ford ASSIGNING <ls\_ford>  WHERE erfasser = iv\_bname  \* afr Sept. 2022 Zeile gestrichen, es ist egal ob jemand aktiv oder nicht ist  \* leider werden Spezialuser wie Berater, Testuser, externe, etc. auch als inaktive geflaged  \* daher darf das inaktiv nicht geprüft werden  \* es ist auch nicht relevant, wenn der user tatsächlich inaktiv ist, kommt er auch nicht ins ETHIS/SAP  \* dann sieht er auch nichts, wenn er aber reinkommt, ist es egal ob als inaktiv geflaged, dann muss er es sehen  \* and inaktiv = abap\_false  AND ( ftype = 'A' OR ftype = 'C' )  AND ( mansp = 'I' OR mansp = 'G' )  AND akfdt < sy-datum  AND flg\_betreiben = abap\_false  AND flg\_ausbuchen = abap\_false.  CLEAR: ls\_inb\_e.  MOVE-CORRESPONDING <ls\_ford> TO ls\_inb\_e.  \* Aktionen ergaenzen  CASE <ls\_ford>-ftype.  WHEN 'A'. "Mahnsperre I  CASE <ls\_ford>-mansp.  WHEN 'I'.  ls\_inb\_e-betr\_box = 'X'.  ls\_inb\_e-verl\_box = 'X'.  ls\_inb\_e-ausb\_box = 'X'.  ENDCASE.  WHEN 'C'. "Mahnsperre G  ls\_inb\_e-mahn\_box = 'X'.  ls\_inb\_e-verl\_box = 'X'.  ls\_inb\_e-ausb\_box = 'X'.  ENDCASE.  \* URL für Dokument holen  TRY.  CALL METHOD zbcarchcl\_archive=>get\_archive\_url\_fi\_document  EXPORTING  iv\_bukrs = <ls\_ford>-bukrs  iv\_belnr = <ls\_ford>-belnr  iv\_gjahr = <ls\_ford>-gjahr  RECEIVING  rt\_url = ls\_inb\_e-url\_dokument.  CATCH zcx\_bcgen.  CLEAR ls\_inb\_e-url\_dokument.  ENDTRY.  APPEND ls\_inb\_e TO lt\_inb\_e.  ENDLOOP.  \*\* Für RW-INBOX "RW-Tabelle LT\_IN\_S  WHEN 'PBETA'. "Bearbeitung aus ETHIS-->Aktion RW  LOOP AT lt\_ford ASSIGNING <ls\_ford>  WHERE ( ftype = 'A' OR ftype = 'C' )  AND ( mansp = 'I' OR mansp = 'G' )  AND ( flg\_ausbuchen = abap\_true OR  flg\_betreiben = abap\_true )  AND akfdt < sy-datum.  MOVE-CORRESPONDING <ls\_ford> TO ls\_inb\_s.  APPEND ls\_inb\_s TO lt\_inb\_s.  ENDLOOP.  \*  WHEN 'PBETF'. "Fällige ETHIS  LOOP AT lt\_ford ASSIGNING <ls\_ford>  WHERE ( ftype = 'A' OR ftype = 'C' )  AND ( mansp = 'I' OR mansp = 'G' )  AND ( flg\_ausbuchen = abap\_false AND  flg\_betreiben = abap\_false ).  \* auch nicht fälligen sollen kommen, damit bearbeitet werden können  \* and akfdt < sy-datum.  \* auch nicht fälligen sollen kommen, damit bearbeitet werden können  MOVE-CORRESPONDING <ls\_ford> TO ls\_inb\_s.  APPEND ls\_inb\_s TO lt\_inb\_s.  ENDLOOP.  \*  WHEN 'PBRW'. "Bearbeitung RW  LOOP AT lt\_ford ASSIGNING <ls\_ford>  WHERE ( ftype = 'B' OR ftype = 'D' )  AND ( mansp = 'I' OR mansp = 'G' )  AND ( blart = 'VD' OR blart = 'RD').  \* auch nicht fälligen sollen kommen, damit bearbeitet werden können  \* and akfdt < sy-datum.  \* auch nicht fälligen sollen kommen, damit bearbeitet werden können  MOVE-CORRESPONDING <ls\_ford> TO ls\_inb\_s.  APPEND ls\_inb\_s TO lt\_inb\_s.  ENDLOOP.  \*  WHEN 'PBRWD'. "Bearbeitung RW andere  LOOP AT lt\_ford ASSIGNING <ls\_ford>  WHERE ( ftype = 'B' OR ftype = 'D' )  AND ( mansp = 'I' OR mansp = 'G' )  AND ( blart NE 'VD' AND blart NE 'RD').  \* auch nicht fälligen sollen kommen, damit bearbeitet werden können  \* and akfdt < sy-datum.  \* auch nicht fälligen sollen kommen, damit bearbeitet werden können  MOVE-CORRESPONDING <ls\_ford> TO ls\_inb\_s.  APPEND ls\_inb\_s TO lt\_inb\_s.  ENDLOOP.  \*  WHEN 'PAET'. "Ansicht ETHIS alle  LOOP AT lt\_ford ASSIGNING <ls\_ford>  WHERE ( ftype = 'A' OR ftype = 'C' )  AND ( mansp = 'I' OR mansp = 'G' ).  MOVE-CORRESPONDING <ls\_ford> TO ls\_inb\_s.  APPEND ls\_inb\_s TO lt\_inb\_s.  ENDLOOP.  \*  WHEN 'PARW'. "Ansicht RW alle  LOOP AT lt\_ford ASSIGNING <ls\_ford>  WHERE ( ftype = 'B' OR ftype = 'D' )  AND ( mansp = 'I' OR mansp = 'G' ).  MOVE-CORRESPONDING <ls\_ford> TO ls\_inb\_s.  APPEND ls\_inb\_s TO lt\_inb\_s.  ENDLOOP.  \*  WHEN OTHERS.  IF iv\_param(2) = 'PI'. "Ansicht/Inkasso  IF iv\_param+2(1) = 'X'. "Ansicht Betreibung alle  LOOP AT lt\_ford ASSIGNING <ls\_ford>  WHERE mansp = 'B'.  MOVE-CORRESPONDING <ls\_ford> TO ls\_inb\_s.  APPEND ls\_inb\_s TO lt\_inb\_s.  ENDLOOP.  ENDIF.  IF iv\_param+3(1) = 'X'. "Ansicht Inkasso alle  LOOP AT lt\_ford ASSIGNING <ls\_ford>  WHERE mansp = 'K'.  MOVE-CORRESPONDING <ls\_ford> TO ls\_inb\_s.  APPEND ls\_inb\_s TO lt\_inb\_s.  ENDLOOP.  ENDIF.  ENDIF.  IF iv\_param(4) = 'PATO'. "--> alle Typen PATOA,B,C,D  IF iv\_param+4(1) = 'X'. "Ansicht FTYPE A  LOOP AT lt\_ford ASSIGNING <ls\_ford>  WHERE ftype = 'A'  AND ( mansp = 'I' OR mansp = 'G' ).  CLEAR ls\_inb\_s.  MOVE-CORRESPONDING <ls\_ford> TO ls\_inb\_s.  APPEND ls\_inb\_s TO lt\_inb\_s.  ENDLOOP.  ENDIF.  IF iv\_param+5(1) = 'X'. " Ansicht FTYPE B  LOOP AT lt\_ford ASSIGNING <ls\_ford>  WHERE ftype = 'B'  AND ( mansp = 'I' OR mansp = 'G' ).  CLEAR ls\_inb\_s.  MOVE-CORRESPONDING <ls\_ford> TO ls\_inb\_s.  APPEND ls\_inb\_s TO lt\_inb\_s.  ENDLOOP.  ENDIF.  IF iv\_param+6(1) = 'X'. " Ansicht FTYPE C  LOOP AT lt\_ford ASSIGNING <ls\_ford>  WHERE ftype = 'C'  AND ( mansp = 'I' OR mansp = 'G' ).  CLEAR ls\_inb\_s.  MOVE-CORRESPONDING <ls\_ford> TO ls\_inb\_s.  APPEND ls\_inb\_s TO lt\_inb\_s.  ENDLOOP.  ENDIF.  IF iv\_param+7(1) = 'X'. " Ansicht FTYPE D  LOOP AT lt\_ford ASSIGNING <ls\_ford>  WHERE ftype = 'D'  AND ( mansp = 'I' OR mansp = 'G' ).  CLEAR ls\_inb\_s.  MOVE-CORRESPONDING <ls\_ford> TO ls\_inb\_s.  APPEND ls\_inb\_s TO lt\_inb\_s.  ENDLOOP.  ENDIF.  ELSE. "irgendwas anderes ?  \* keine Aktion, Tabelle lt\_inb\_s = leer  ENDIF.  ENDCASE.  \* Sortieren der Tabellen  SORT lt\_inb\_e BY ftype belnr.  SORT lt\_inb\_s BY ftype belnr.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  et\_inbox\_e = lt\_inb\_e.  et\_inbox\_s = lt\_inb\_s.  ENDMETHOD.  method get\_forderungen\_verdichtung.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Es werden die Einträge der Forderungen verdichtet  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lt\_verd type zfi\_t\_nb\_ford\_verd,  ls\_verd type zfi\_s\_nb\_ford\_verd,  ls\_verd\_alt type zfi\_s\_nb\_ford\_verd,  ls\_key type zfi\_s\_nb\_forderung\_key,  lt\_bseg type zfi\_t\_nb\_ford\_verd,  ls\_bseg type zfi\_s\_nb\_ford\_verd.  data: lv\_subrc type subrc,  lv\_lines type int4,  lv\_space type flag.  field-symbols: <ls\_verd> type zfi\_s\_nb\_ford\_verd,  <ls\_bseg> type zfi\_s\_nb\_ford\_verd.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  \*-----------------------------------------------------------------------  \* Tabellen übernehmen --------------------------------------------------  \*-----------------------------------------------------------------------  select \* from bseg into corresponding fields of table lt\_verd  where bukrs = is\_key-bukrs  and belnr = is\_key-belnr  and gjahr = is\_key-gjahr  and koart = 'S'  and hkont ne '0020001001'.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  \* sortieren  sort lt\_verd.  \* 1. Komprimierung ohne Anpassung  delete adjacent duplicates from lt\_verd  comparing hkont projk kostl sgtxt.  describe table lt\_verd lines lv\_lines.  if lv\_lines = 1.  loop at lt\_verd assigning <ls\_verd>.  move-corresponding <ls\_verd> to es\_bseg.  endloop.  else.  \* 2. Komprimierung für HKONT  clear: ls\_verd\_alt, lv\_space.  loop at lt\_verd assigning <ls\_verd>.  if ls\_verd\_alt is initial.  ls\_verd\_alt = <ls\_verd>.  \* do noting.  else.  if <ls\_verd>-hkont ne ls\_verd\_alt-hkont.  lv\_space = 'X'.  exit.  else.  ls\_verd\_alt = <ls\_verd>.  endif.  endif.  endloop.  if lv\_space = 'X'.  \* betreffendes Feld auf Space setzen.  loop at lt\_verd assigning <ls\_verd>.  <ls\_verd>-hkont = space.  endloop.  endif.  delete adjacent duplicates from lt\_verd  comparing hkont projk kostl sgtxt.  \* prüfen ob nur noch eine Zeile  describe table lt\_verd lines lv\_lines.  if lv\_lines = 1.  loop at lt\_verd assigning <ls\_verd>.  move-corresponding <ls\_verd> to es\_bseg.  endloop.  else.  \* 3. Komprimierung für PROJK  clear: ls\_verd\_alt, lv\_space.  loop at lt\_verd assigning <ls\_verd>.  if ls\_verd\_alt is initial.  ls\_verd\_alt = <ls\_verd>.  \* do noting.  else.  if <ls\_verd>-projk ne ls\_verd\_alt-projk.  lv\_space = 'X'.  exit.  else.  ls\_verd\_alt = <ls\_verd>.  endif.  endif.  endloop.  if lv\_space = 'X'.  \* betreffendes Feld auf Space setzen.  loop at lt\_verd assigning <ls\_verd>.  <ls\_verd>-projk = space.  endloop.  endif.  delete adjacent duplicates from lt\_verd  comparing hkont projk kostl sgtxt.  \* prüfen ob nur noch eine Zeile  describe table lt\_verd lines lv\_lines.  if lv\_lines = 1.  loop at lt\_verd assigning <ls\_verd>.  move-corresponding <ls\_verd> to es\_bseg.  endloop.  else.  \* 4. Komprimierung für SGTXT  clear: ls\_verd\_alt, lv\_space.  loop at lt\_verd assigning <ls\_verd>.  if ls\_verd\_alt is initial.  ls\_verd\_alt = <ls\_verd>.  \* do noting.  else.  if <ls\_verd>-sgtxt ne ls\_verd\_alt-sgtxt.  lv\_space = 'X'.  exit.  else.  ls\_verd\_alt = <ls\_verd>.  endif.  endif.  endloop.  if lv\_space = 'X'.  \* betreffendes Feld auf Space setzen.  loop at lt\_verd assigning <ls\_verd>.  <ls\_verd>-sgtxt = space.  endloop.  endif.  delete adjacent duplicates from lt\_verd  comparing hkont projk kostl sgtxt.  \* prüfen ob nur noch eine Zeile  describe table lt\_verd lines lv\_lines.  if lv\_lines = 1.  loop at lt\_verd assigning <ls\_verd>.  move-corresponding <ls\_verd> to es\_bseg.  endloop.  else.  \* Fehler aufgetaucht ?!?  endif.  endif.  endif.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  endmethod.  method get\_forderungstyp .  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Zuteilen Forderungstyp  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  \* Zuteilen Forderungstyp  case iv\_mansp.  when 'I'.  case iv\_blart.  when 'VD'.  case iv\_vkbur.  when 'AD01'. ev\_typ = 'B'.  when 'AD02'. ev\_typ = 'B'.  when 'AD03'. ev\_typ = 'B'.  when 'AD04'. ev\_typ = 'B'.  when 'AD05'. ev\_typ = 'B'.  when 'AD06'. ev\_typ = 'B'.  when 'EB01'. ev\_typ = 'B'.  when 'RW02'. ev\_typ = 'B'.  when 'IS10'. ev\_typ = 'B'.  when others. ev\_typ = 'A'.  endcase.  when 'RD'.  case iv\_vkbur.  when 'FM01'. ev\_typ = 'A'.  when others. ev\_typ = 'B'.  endcase.  when 'RS'. ev\_typ = 'B'.  when others. ev\_typ = 'B'.  endcase.  when 'G'.  case iv\_blart.  when 'VD'.  case iv\_vkbur.  when 'AD01'. ev\_typ = 'D'.  when 'AD02'. ev\_typ = 'D'.  when 'AD03'. ev\_typ = 'D'.  when 'AD04'. ev\_typ = 'D'.  when 'AD05'. ev\_typ = 'D'.  when 'AD06'. ev\_typ = 'D'.  when 'EB01'. ev\_typ = 'D'.  when 'RW02'. ev\_typ = 'D'.  when 'IS10'. ev\_typ = 'D'.  when others. ev\_typ = 'C'.  endcase.  when 'RD'.  case iv\_vkbur.  when 'FM01'. ev\_typ = 'C'.  when others. ev\_typ = 'D'.  endcase.  when 'RS'. ev\_typ = 'D'.  when others. ev\_typ = 'D'.  endcase.  when 'B' or 'K'.  case iv\_blart.  when 'VD'.  case iv\_vkbur.  when 'AD01'. ev\_typ = 'B'.  when 'AD02'. ev\_typ = 'B'.  when 'AD03'. ev\_typ = 'B'.  when 'AD04'. ev\_typ = 'B'.  when 'AD05'. ev\_typ = 'B'.  when 'AD06'. ev\_typ = 'B'.  when 'EB01'. ev\_typ = 'B'.  when 'RW02'. ev\_typ = 'B'.  when 'IS10'. ev\_typ = 'B'.  when others. ev\_typ = 'A'.  endcase.  when 'RD'.  case iv\_vkbur.  when 'FM01'. ev\_typ = 'A'.  when others. ev\_typ = 'B'.  endcase.  when 'RS'. ev\_typ = 'B'.  when others. ev\_typ = 'B'.  endcase.  endcase.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  endmethod.  method get\_forderung\_ausbuch\_pdf .  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Erstellen Ausbuchungsdokument für RW  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: ls\_ausb type zfi\_s\_nb\_ford\_ausbuchung.  data: ls\_outputpar type sfpoutputparams,  lv\_fmname type rs38l\_fnam,  ls\_docparams type sfpdocparams,  ls\_formoutput type fpformoutput.  \*-------------------------------------------------------------  \* Uebernahme Input Parameter  \*-------------------------------------------------------------  ls\_ausb = is\_ausbuchung.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  try.  call function 'FP\_FUNCTION\_MODULE\_NAME'  exporting  i\_name = 'ZFI\_NB\_FORD\_AUSBUCHEN'  importing  e\_funcname = lv\_fmname.  catch cx\_root.  exit.  endtry.  \* gv\_fmname = /1BCDWB/SM00000139  "getpdf  ls\_outputpar-nodialog = abap\_true.  ls\_outputpar-getpdf = abap\_true.  call function 'FP\_JOB\_OPEN'  changing  ie\_outputparams = ls\_outputpar  exceptions  cancel = 1  usage\_error = 2  system\_error = 3  internal\_error = 4  others = 5.  \*\*\*  set country 'CH'.  ls\_docparams-langu = sy-langu.  ls\_docparams-country = 'CH'.  \*\*\*  call function lv\_fmname "'/1BCDWB/SM00000139'  exporting  /1bcdwb/docparams = ls\_docparams  is\_data = ls\_ausb  importing  /1bcdwb/formoutput = ls\_formoutput  exceptions  usage\_error = 1  system\_error = 2  internal\_error = 3  others = 4.  if sy-subrc <> 0.  \* Implement suitable error handling here  endif.  \*\*\*  \*\*\* CALL FUNCTION 'FP\_JOB\_CLOSE'  \*\*\* EXCEPTIONS  \*\*\* usage\_error = 1  \*\*\* system\_error = 2  \*\*\* internal\_error = 3  \*\*\* OTHERS = 4.  \*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*direkt anzeigen  \*\*\* IF iv\_show EQ abap\_true.  do\_show\_pdf( iv\_pdf = ls\_formoutput-pdf ).  \*\*\* ENDIF.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  endmethod.  method GET\_FS\_OPENFILE.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Datei mit Dialog übernehmen  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  DATA: lt\_filetable TYPE filetable,  lv\_status TYPE i.  DATA: lt\_1024\_tab TYPE STANDARD TABLE OF tbl1024,  ls\_1024\_tab LIKE LINE OF lt\_1024\_tab.  DATA: lv\_archiv\_id TYPE saearchivi,  lv\_arc\_doc\_id TYPE saeardoid,  lv\_doc\_type TYPE saedoktyp.  DATA: lv\_filelength TYPE i,  lv\_filelength\_fm TYPE num12.  DATA: ev\_url TYPE string,  es\_ixos\_webviewer TYPE zbc\_ixos\_webview.  IF iv\_filename IS INITIAL.  CALL METHOD cl\_gui\_frontend\_services=>file\_open\_dialog  EXPORTING  window\_title = 'Dokument übernehmen '  default\_filename = space  \* file\_filter = lv\_file\_filter  file\_filter = 'PDF'  \* with\_encoding =  \* initial\_directory =  \* multiselection = SPACE  CHANGING  file\_table = lt\_filetable  rc = lv\_status  \* user\_action = lv\_action  \* file\_encoding =  EXCEPTIONS  file\_open\_dialog\_failed = 1  cntl\_error = 2  error\_no\_gui = 3  not\_supported\_by\_gui = 4  OTHERS = 5.  IF sy-subrc IS INITIAL AND lv\_status > 0.  READ TABLE lt\_filetable INTO ev\_filename INDEX 1.  ENDIF.  ELSE.  ev\_filename = iv\_filename.  ENDIF.  \*  "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  "xstring  DATA: l\_rawtab TYPE STANDARD TABLE OF raw255,  l\_len TYPE i.  CALL METHOD cl\_gui\_frontend\_services=>gui\_upload  EXPORTING  filename = ev\_filename  filetype = 'BIN'  IMPORTING  filelength = l\_len  CHANGING  data\_tab = l\_rawtab  EXCEPTIONS  file\_open\_error = 1  file\_read\_error = 2  no\_batch = 3  gui\_refuse\_filetransfer = 4  invalid\_type = 5  no\_authority = 6  unknown\_error = 7  bad\_data\_format = 8  header\_not\_allowed = 9  separator\_not\_allowed = 10  header\_too\_long = 11  unknown\_dp\_error = 12  access\_denied = 13  dp\_out\_of\_memory = 14  disk\_full = 15  dp\_timeout = 16  not\_supported\_by\_gui = 17  error\_no\_gui = 18  OTHERS = 19.  IF sy-subrc IS NOT INITIAL.  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  ENDIF.  CALL FUNCTION 'SCMS\_BINARY\_TO\_XSTRING'  EXPORTING  input\_length = l\_len  IMPORTING  buffer = ev\_xstring  TABLES  binary\_tab = l\_rawtab  EXCEPTIONS  failed = 1  OTHERS = 2.  IF sy-subrc IS NOT INITIAL.  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  ENDIF.  endmethod.  method get\_kunnr\_from\_bp.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* es wird der Debitor geholt aufgrund des BP  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: ls\_but000 type but000.  data: lv\_kunnr type kunnr,  lv\_partner\_guid type bu\_partner\_guid,  lv\_bp type bu\_partner,  lv\_name\_last type bu\_namep\_l,  lv\_name\_first type bu\_namep\_f.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_bp = iv\_bp.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  \* Holen Customer guid  \* Holen Partner aufgrund der Partner\_guid  select partner\_guid name\_last name\_first from but000 into  ( lv\_partner\_guid, lv\_name\_last, lv\_name\_first )  where partner = lv\_bp  and ( type = co\_bu\_type\_1 or type = co\_bu\_type\_2 ).  endselect.  if sy-subrc = 0.  \* Holen customer aufgrund Partner\_Guid  select customer from cvi\_cust\_link into lv\_kunnr  where partner\_guid = lv\_partner\_guid.  endselect.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  ev\_kunnr = lv\_kunnr.  ev\_name1 = lv\_name\_last && ' ' && lv\_name\_first.  endmethod.  method get\_lifnr\_from\_bp.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* es wird der Kreditor geholt aufgrund des BP  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: ls\_but000 type but000.  data: lv\_lifnr type lifnr,  lv\_partner\_guid type bu\_partner\_guid,  lv\_bp type bu\_partner,  lv\_name\_last type bu\_namep\_l,  lv\_name\_first type bu\_namep\_f.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_bp = iv\_bp.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  \* Holen Vendor guid  \* Holen Partner aufgrund der Partner\_guid  select partner\_guid name\_last name\_first from but000 into  ( lv\_partner\_guid, lv\_name\_last, lv\_name\_first )  where partner = lv\_bp  and ( type = co\_bu\_type\_1 or type = co\_bu\_type\_2 ).  endselect.  if sy-subrc = 0.  \* Holen Lieferant aufgrund Partner\_Guid  select vendor from cvi\_vend\_link into lv\_lifnr  where partner\_guid = lv\_partner\_guid.  endselect.  endif.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  ev\_lifnr = lv\_lifnr.  ev\_name1 = lv\_name\_last && ' ' && lv\_name\_first.  endmethod.  method get\_popup\_history.  types: begin of fields,  sval type sval.  types: end of fields.  data: it\_fields type table of fields,  wa\_fields like line of it\_fields.  data: returncode(1) type c,  wf\_popup\_title type char50.  \* POPUP aufbauen mit den Feldern  \*\* clear wa\_fields.  \*\* wa\_fields-sval-tabname = 'ZFI\_NB\_FORD\_HIST'.  \*\* wa\_fields-sval-fieldname = 'BUKRS'.  \*\* wa\_fields-sval-fieldtext = 'Buchungskreis'.  \*\* wa\_fields-sval-value = iv\_bukrs.  \*\* append wa\_fields to it\_fields.  clear wa\_fields.  wa\_fields-sval-tabname = 'ZFI\_NB\_FORD\_HIST'.  wa\_fields-sval-fieldname = 'BELNR'.  wa\_fields-sval-fieldtext = 'Belegnummer'.  wa\_fields-sval-value = iv\_belnr.  append wa\_fields to it\_fields.  \*\* clear wa\_fields.  \*\* wa\_fields-sval-tabname = 'ZFI\_NB\_FORD\_HIST'.  \*\* wa\_fields-sval-fieldname = 'GJAHR'.  \*\* wa\_fields-sval-fieldtext = 'GJAHR'.  \*\* wa\_fields-sval-value = iv\_gjahr.  \*\* append wa\_fields to it\_fields.  wf\_popup\_title = 'Bitte Angaben eintragen '.  call function 'POPUP\_GET\_VALUES'  exporting  no\_value\_check = ' '  popup\_title = wf\_popup\_title  \* START\_COLUMN = '5'  \* START\_ROW = '5'  importing  returncode = returncode  tables  fields = it\_fields.  \* EXCEPTIONS  \* ERROR\_IN\_FIELDS = 1  \* OTHERS = 2  if returncode = 'A'. "Anwender hat abgebrochen  ev\_subrc = 1.  \*  else.  \* Daten übernehmen  loop at it\_fields into wa\_fields.  case wa\_fields-sval-fieldname.  when 'BUKRS'.  ev\_bukrs = wa\_fields-sval-value.  when 'BELNR'.  ev\_belnr = wa\_fields-sval-value.  when 'GJAHR'.  ev\_gjahr = wa\_fields-sval-value.  endcase.  endloop.  endif.  endmethod.  method mahnsperrre\_entfernen.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* mit dieser Methode kann die Mahnsperre entfernt werden  \*  \* die ETHIS-INBOX triggert die Aktion Mahnen  \* aufgrund dieser Meldung soll die Mahnsperre entfernt werden,  \* zudem wird in der ZFI\_NB\_FORD\_HIST ein Eintrag erstellt.  \*  \* Damit ein Aenderungseintrag für den Beleg erstellt wird,  \* wird dies mit FB02 (BTCI) durchgeführt.  \*  \* Die erstellte BTCI-Mappe muss in zwei Versionen aufgerufen werden:  \* - Normaler Beleg  \* - Beleg mit CPD-Kunde, dieser springt noch auf die BSEC  \* - damit festgestellt wird, ob ein CPD dabei ist, muss dies  \* via BSID geholt werden, da diese Info beim Aufruf der ME fehlt  \*  \* Prüfen, mit welchem Benutzer dies gemacht werden kann/soll  \* --> der dezentrale Benutzer wird diese Berechtigung selber  \* nicht haben  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: ls\_key type zfi\_s\_nb\_forderung\_key,  lv\_aenderer type usnam.  data: ls\_bapreturn type bapireturn.  data: lt\_bdcdata type bdcdata\_tab,  lt\_messages type tab\_bdcmsgcoll,  ls\_bdcdata type bdcdata,  ls\_messages type bdcmsgcoll.  data: lv\_xcpdd type xcpdd,  lv\_subrc type subrc.  constants: co\_tcode\_fb02 type tcode value 'FB02'.  field-symbols: <ls\_messages> type bdcmsgcoll.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  ls\_key = is\_forderung\_key.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  sy-uname = 'FIADMIN'.  \*\*\*-----------------------------------------------------------  \* Holen CPD-Flag aus BSID  \*\*\*-----------------------------------------------------------  clear lv\_xcpdd.  select xcpdd from bsid into lv\_xcpdd  where bukrs = ls\_key-bukrs  and gjahr = ls\_key-gjahr  and belnr = ls\_key-belnr  and buzei = ls\_key-buzei.  endselect.  \*\*\*-----------------------------------------------------------  \*\*\* Aufbau BTCI  \*\*\*-----------------------------------------------------------  \*\*  \* Zahlsperre via BTCI anpassen  refresh lt\_bdcdata.  clear ls\_bdcdata.  \*\*\* Einstiegsbild "Normal und CPD  clear ls\_bdcdata.  ls\_bdcdata-program = 'SAPMF05L'.  ls\_bdcdata-dynpro = '0100'.  ls\_bdcdata-dynbegin = 'X'.  append ls\_bdcdata to lt\_bdcdata.  clear ls\_bdcdata.  ls\_bdcdata-fnam = 'BDC\_CURSOR'.  ls\_bdcdata-fval = 'RF05L-BELNR'.  append ls\_bdcdata to lt\_bdcdata.  clear ls\_bdcdata.  ls\_bdcdata-fnam = 'BDC\_OKCODE'.  ls\_bdcdata-fval = '=WEITE'.  append ls\_bdcdata to lt\_bdcdata.  \* Aufruf Beleg  clear ls\_bdcdata.  ls\_bdcdata-fnam = 'RF05L-BELNR'.  ls\_bdcdata-fval = ls\_key-belnr.  append ls\_bdcdata to lt\_bdcdata.  clear ls\_bdcdata.  ls\_bdcdata-fnam = 'RF05L-BUKRS'.  ls\_bdcdata-fval = ls\_key-bukrs.  append ls\_bdcdata to lt\_bdcdata.  clear ls\_bdcdata.  ls\_bdcdata-fnam = 'RF05L-GJAHR'.  ls\_bdcdata-fval = ls\_key-gjahr.  append ls\_bdcdata to lt\_bdcdata.  \* Sprung auf Zeile 1  clear ls\_bdcdata.  ls\_bdcdata-program = 'SAPMF05L'.  ls\_bdcdata-dynpro = '0700'.  ls\_bdcdata-dynbegin = 'X'.  append ls\_bdcdata to lt\_bdcdata.  clear ls\_bdcdata.  ls\_bdcdata-fnam = 'BDC\_CURSOR'.  ls\_bdcdata-fval = 'RF05L-ANZDT(01)'.  append ls\_bdcdata to lt\_bdcdata.  clear ls\_bdcdata.  ls\_bdcdata-fnam = 'BDC\_OKCODE'.  ls\_bdcdata-fval = '=PK'.  append ls\_bdcdata to lt\_bdcdata.  \* Sprung auf CPD  if lv\_xcpdd = abap\_true.  clear ls\_bdcdata.  ls\_bdcdata-program = 'SAPLFCPD'.  ls\_bdcdata-dynpro = '0100'.  ls\_bdcdata-dynbegin = 'X'.  append ls\_bdcdata to lt\_bdcdata.  clear ls\_bdcdata.  ls\_bdcdata-fnam = 'BDC\_CURSOR'.  ls\_bdcdata-fval = 'BSEC-SPRAS'.  append ls\_bdcdata to lt\_bdcdata.  clear ls\_bdcdata.  ls\_bdcdata-fnam = 'BDC\_OKCODE'.  ls\_bdcdata-fval = '/00'.  append ls\_bdcdata to lt\_bdcdata.  endif.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* 2. Bildschirm: Belegzeile \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Mahnsperre löschen  clear ls\_bdcdata.  ls\_bdcdata-program = 'SAPMF05L'.  ls\_bdcdata-dynpro = '0301'.  ls\_bdcdata-dynbegin = 'X'.  append ls\_bdcdata to lt\_bdcdata.  clear ls\_bdcdata.  ls\_bdcdata-fnam = 'BDC\_CURSOR'.  ls\_bdcdata-fval = 'BSEG-MANSP'.  append ls\_bdcdata to lt\_bdcdata.  clear ls\_bdcdata.  ls\_bdcdata-fnam = 'BDC\_OKCODE'.  ls\_bdcdata-fval = '=AE'.  append ls\_bdcdata to lt\_bdcdata.  clear ls\_bdcdata.  ls\_bdcdata-fnam = 'BSEG-MANSP'.  ls\_bdcdata-fval = ' '.  append ls\_bdcdata to lt\_bdcdata.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Transaktion aufrufen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  call transaction co\_tcode\_fb02  using lt\_bdcdata "BDCDATA  mode 'N' "Dunkel abspielen  update 'S' "Synchron verbuchen  messages into lt\_messages. "Msg's in Tab (Standard-format)  \*\*  \* prüfen auf Fehler  clear lv\_subrc.  loop at lt\_messages assigning <ls\_messages>.  if <ls\_messages>-msgid = 'F5'  and <ls\_messages>-msgnr ne '300'.  lv\_subrc = 1.  endif.  endloop.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Export-Parameter aufbauen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ev\_subrc = lv\_subrc.  endmethod.  method mail\_ford\_get\_raw\_text.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Roh Text (also noch mit den Variablen) ermitteln \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lv\_langu type langu,  lv\_text\_type type string,  lv\_id type tdobname.  data: ls\_text like line of zficl\_nb=>gt\_text.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_raw\_text.  \*==========================================\*  \* Import \*  \*==========================================\*  clear: lv\_text\_type.  lv\_text\_type = iv\_text\_type.  lv\_langu = iv\_langu.  \* \*\*\* Sprache: Nur Deutsch und Englisch  clear: lv\_langu.  case iv\_langu.  when 'D' or 'E'.  lv\_langu = iv\_langu.  when others.  lv\_langu = 'D'.  endcase.  \*==========================================\*  \* Buffer lesen \*  \*==========================================\*  clear: ls\_text.  read table zficl\_nb=>gt\_text  into ls\_text  with table key text\_type = lv\_text\_type  langu = lv\_langu.  if sy-subrc eq 0.  et\_raw\_text = ls\_text-text\_tab.  return.  endif.  \*==========================================\*  \* Von der DB lesen \*  \*==========================================\*  clear: ls\_text.  ls\_text-text\_type = lv\_text\_type.  ls\_text-langu = lv\_langu.  clear: lv\_id.  case lv\_text\_type.  when 'FORD\_MAIL'.  lv\_id = 'ZFI\_FORD\_MAIL'.  when others.  clear: lv\_id.  endcase.  refresh: ls\_text-text\_tab.  if not lv\_id is initial.  ls\_text-text\_tab = zbccl\_text=>read\_text\_to\_string\_table(  i\_name = lv\_id  i\_id = zficl\_nb=>co\_text\_id\_st "ST  i\_object = zficl\_nb=>co\_text\_object\_zins "Text  i\_spras = lv\_langu ).  endif.  \*===========================================\*  \* Export \*  \*===========================================\*  et\_raw\_text = ls\_text-text\_tab.  \*\*\*  \*===========================================\*  \* Buffer \*  \*===========================================\*  insert ls\_text into table zficl\_nb=>gt\_text.  endmethod.  method mail\_ford\_information.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Mail: Nachfrage aufbauen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lt\_parameter\_value type wdr\_simple\_name\_value\_map,  ls\_parameter\_value like line of lt\_parameter\_value,  ls\_daten type zfi\_s\_nb\_ford\_mail,  lt\_text type string\_table.  data: ls\_info type string,  lv\_langu type langu,  lv\_link type char30.  constants: co\_sysid\_s4d type sysid value 'S4D',  co\_sysid\_s4q type sysid value 'S4Q',  co\_sysid\_s4p type sysid value 'S4P'.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Import Daten \*  \*==========================================\*  ls\_daten = is\_daten.  lv\_langu = 'D'.  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_text\_info.  \*\*  \*\*==========================================\*  \* Initialisierung \*  \*===========================================\*  clear ls\_info.  case sy-sysid.  when co\_sysid\_s4d. lv\_link = 'https://ethis-d.ethz.ch'.  when co\_sysid\_s4q. lv\_link = 'https://ethis-q.ethz.ch'.  when co\_sysid\_s4p. lv\_link = 'https://ethis.ethz.ch'.  when others. lv\_link = 'https://ethis-d.ethz.ch'.  endcase.  \*==========================================\*  \* Parametertabelle füllen \*  \*==========================================\*  refresh: lt\_parameter\_value.  \*\*  \* System (SYSID)  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `SYSTEM`.  ls\_parameter\_value-value = sy-sysid.  insert ls\_parameter\_value into table lt\_parameter\_value.  \* ENAME (Nachname, Vorname)  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `ENAME`.  ls\_parameter\_value-value = ls\_daten-ename.  insert ls\_parameter\_value into table lt\_parameter\_value.  \* LINK (auf System)  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `LINK`.  ls\_parameter\_value-value = lv\_link.  insert ls\_parameter\_value into table lt\_parameter\_value.  \*\*\*==========================================\*  \*\*\* Mailtext ermitteln \*  \*\*\*==========================================\*  refresh: lt\_text.  lt\_text = zficl\_nb=>mail\_ford\_process\_text(  iv\_text\_type = zficl\_nb=>co\_text\_ford\_mail  it\_parameter\_value = lt\_parameter\_value  iv\_langu = lv\_langu ).  if lt\_text is initial.  return.  endif.  \*\*  \*\*\*==========================================\*  \*\*\* Export \*  \*\*\*==========================================\*  et\_text\_info = lt\_text.  endmethod.  method mail\_ford\_process\_text.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Prozess: Mailtext aufbereiten \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lt\_text type string\_table.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_text.  \*==========================================\*  \* Import \*  \*==========================================\*  if iv\_text\_type is initial.  return.  endif.  \*==========================================\*  \* Text \*  \*==========================================\*  refresh: lt\_text.  lt\_text = zficl\_nb=>mail\_ford\_get\_raw\_text(  iv\_text\_type = iv\_text\_type  iv\_langu = iv\_langu ).  if lt\_text is initial.  return.  endif.  \*==========================================\*  \* Parameter ersetzen \*  \*==========================================\*  if not it\_parameter\_value is initial.  et\_text = zficl\_nb=>mail\_replace\_parameter(  it\_parameter\_value = it\_parameter\_value  it\_text\_raw = lt\_text ).  else.  et\_text = lt\_text.  endif.  endmethod.  method mail\_ford\_send.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Versand Info-Mail an ETHIS-Benutzer \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: ls\_mail type zfi\_s\_nb\_ford\_mail.  data: lt\_parameter\_value type wdr\_simple\_name\_value\_map,  ls\_parameter\_value like line of lt\_parameter\_value,  ls\_adressdaten type zfi\_s\_nb\_adressdaten.  data: lt\_recipient\_to type string\_table,  lt\_recipient\_cc type string\_table,  lv\_sender type ad\_smtpadr,  lv\_empfaenger\_rw type ad\_smtpadr,  lv\_empfaenger\_hr type ad\_smtpadr,  lv\_langu type spras.  data: lt\_text\_hlp type string\_table,  lt\_text type string\_table,  ls\_text like line of lt\_text,  lt\_mailtext type bcsy\_text,  ls\_mailtext like line of lt\_mailtext.  data: lv\_subject type string,  lv\_string type string,  lv\_data\_xstring type xstring,  lt\_data\_xstring type table of xstring.  constants: co\_empf\_test type ad\_smtpadr value 'niederer@id.ethz.ch',  co\_empf\_rw type ad\_smtpadr value 'schlmich@ethz.ch',  co\_sender\_test type ad\_smtpadr value 'no-response@fc.ethz.ch',  co\_sender type ad\_smtpadr value 'debitoren@fc.ethz.ch'.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Import \*  \*==========================================\*  ls\_mail = is\_ford\_mail.  \*==========================================\*  \* Email aufbereiten \*  \*==========================================\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Sprache ablegen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  lv\_langu = 'D'.  \*--------------------------------------------\*  \* Mail aufbereiten \*  \*--------------------------------------------\*  \*\*\*------------------------------------------\*  \*\*\* Info an Benutzer (S4P), \*  \* RW (S4Q), \*  \* NIW (S4D) \*  \*\*\*------------------------------------------\*  refresh:  lt\_text\_hlp.  call method zficl\_nb=>mail\_ford\_information  exporting  is\_daten = ls\_mail  receiving  et\_text\_info = lt\_text\_hlp.  if not lt\_text\_hlp is initial.  insert lines of lt\_text\_hlp into table lt\_text.  endif.  \* Leerzeile  clear: lv\_string.  lv\_string = `<br>`.  insert lv\_string into table lt\_text.  \*===========================================\*  \* Mail versenden \*  \*===========================================\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Empfänger \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------\*  \* Empfänger TO \*  \*-------------------------------------------\*  refresh: lt\_recipient\_to.  \* Empfänger zuteilen  clear lv\_string.  case sy-sysid.  when 'S4P'. lv\_string = ls\_mail-mail.  when 'S4Q'. lv\_string = co\_empf\_test.  when 'S4D'. lv\_string = co\_empf\_test.  when others. lv\_string = co\_empf\_test.  endcase.  \* Empfänger übernehmen in Empfänger liste  insert lv\_string into table lt\_recipient\_to.  \*-------------------------------------------\*  \* Empfänger CC: Delegierte \*  \*-------------------------------------------\*  \*\* REFRESH: lt\_recipient\_cc.  \*-------------------------------------------\*  \* Subject / Betreff \*  \*-------------------------------------------\*  clear: lv\_subject.  case sy-sysid.  when 'S4P'.  lv\_subject = 'Offene Forderungen'.  when others.  lv\_subject = sy-sysid && ': ' && 'Offene Forderungen'.  endcase.  \*-------------------------------------------\*  \* Mailtext \*  \*-------------------------------------------\*  refresh: lt\_mailtext.  loop at lt\_text into ls\_text.  clear: ls\_mailtext.  ls\_mailtext-line = ls\_text.  insert ls\_mailtext into table lt\_mailtext.  endloop.  \*-------------------------------------------\*  \* Attachments für Mail \*  \*-------------------------------------------\*  \* einfach Tabelle weiterleiten,  \* Verarbeitung erfolgt in send\_bcs\_mail  \* keine Attachments !!  \*-------------------------------------------\*  \* Sender eruieren \*  \*-------------------------------------------\*  \* Absender festlegen fix (SAPMAIL\_FFM4)  case sy-sysid.  when 'S4P'.  lv\_sender = co\_sender.  when others.  lv\_sender = co\_sender\_test.  endcase.  \*-------------------------------------------\*  \* Mail verschicken \*  \*-------------------------------------------\*  try.  zficl\_nb=>send\_bcs\_mail(  exporting  iv\_email\_sender = lv\_sender  it\_body = lt\_mailtext  \* it\_attach = lt\_attach  iv\_subject = lv\_subject  iv\_send\_immediately = abap\_false "Mail mit Job verschicken  it\_recipient\_to = lt\_recipient\_to  it\_recipient\_cc = lt\_recipient\_cc  iv\_content\_type = 'HTM'  iv\_with\_commit\_work = 'X'  iv\_spras = lv\_langu ).  \* iv\_with\_commit\_work = iv\_with\_commit\_work ).  catch cx\_bcs ##no\_handler. " BCS: Allgemeine Ausnahmen  \* \*\*\* Ignorieren  endtry.  endmethod.  method mail\_inst\_abschluss.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Mail: Abschluss aufbauen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lt\_parameter\_value type wdr\_simple\_name\_value\_map,  ls\_parameter\_value like line of lt\_parameter\_value,  ls\_daten type zfi\_s\_nb\_benach\_mahndruck,  lt\_text type string\_table.  data: ls\_abschluss type string,  lv\_langu type langu.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Import Daten \*  \*==========================================\*  ls\_daten = is\_daten.  lv\_langu = ls\_daten-spras.  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_text\_abschluss.  \*\*  \*\*==========================================\*  \* Aufbau des Abschluss \*  \*===========================================\*  \*\*\*==========================================\*  \*\*\* Parametertabelle füllen \*  \*\*\*==========================================\*  refresh: lt\_parameter\_value.  \*\*  \*\*\* Mahner  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `MAHNER`.  ls\_parameter\_value-value = sy-uname.  insert ls\_parameter\_value into table lt\_parameter\_value.  \*==========================================\*  \* Export \*  \*==========================================\*  \*\*\*==========================================\*  \*\*\* Mailtext ermitteln \*  \*\*\*==========================================\*  refresh: lt\_text.  lt\_text = zficl\_nb=>mail\_inst\_process\_text(  iv\_text\_type = zficl\_nb=>co\_text\_abschluss  it\_parameter\_value = lt\_parameter\_value  iv\_langu = lv\_langu ).  if lt\_text is initial.  return.  endif.  \*==========================================\*  \* Export \*  \*==========================================\*  et\_text\_abschluss = lt\_text.  endmethod.  method mail\_inst\_begruessung.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Mail: Begrüssung aufbauen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lt\_parameter\_value type wdr\_simple\_name\_value\_map,  ls\_parameter\_value like line of lt\_parameter\_value,  ls\_daten type zfi\_s\_nb\_benach\_mahndruck,  lt\_text type string\_table.  data: ls\_anrede type string,  lv\_langu type langu.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Import Daten \*  \*==========================================\*  ls\_daten = is\_daten.  lv\_langu = ls\_daten-spras.  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_text\_anrede.  \*\*  \*\*==========================================\*  \* Aufbau der Anrede \*  \*===========================================\*  case lv\_langu.  when 'D'.  concatenate 'Sehr geehrte/r' ls\_daten-ename  into ls\_anrede separated by space.  when 'E'.  concatenate 'Dear' ls\_daten-ename  into ls\_anrede separated by space.  when others.  concatenate 'Sehr geehrte/r' ls\_daten-ename  into ls\_anrede separated by space.  endcase.  \*\*\*==========================================\*  \*\*\* Parametertabelle füllen \*  \*\*\*==========================================\*  refresh: lt\_parameter\_value.  \*\*  \*\*\* Anrede  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `ANREDE`.  ls\_parameter\_value-value = ls\_anrede.  insert ls\_parameter\_value into table lt\_parameter\_value.  \*==========================================\*  \* Export \*  \*==========================================\*  \*\*\*==========================================\*  \*\*\* Mailtext ermitteln \*  \*\*\*==========================================\*  refresh: lt\_text.  lt\_text = zficl\_nb=>mail\_inst\_process\_text(  iv\_text\_type = zficl\_nb=>co\_text\_anrede  it\_parameter\_value = lt\_parameter\_value  iv\_langu = lv\_langu ).  if lt\_text is initial.  return.  endif.  \*==========================================\*  \* Export \*  \*==========================================\*  et\_text\_anrede = lt\_text.  endmethod.  method mail\_inst\_get\_raw\_text.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Roh Text (also noch mit den Variablen) ermitteln \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lv\_langu type langu,  lv\_text\_type type string,  lv\_id type tdobname.  data: ls\_text like line of zficl\_nb=>gt\_text.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_raw\_text.  \*==========================================\*  \* Import \*  \*==========================================\*  clear: lv\_text\_type.  lv\_text\_type = iv\_text\_type.  lv\_langu = iv\_langu.  \* \*\*\* Sprache: Nur Deutsch und Englisch  clear: lv\_langu.  case iv\_langu.  when 'D' or 'E'.  lv\_langu = iv\_langu.  when others.  lv\_langu = 'D'.  endcase.  \*==========================================\*  \* Buffer lesen \*  \*==========================================\*  clear: ls\_text.  read table zficl\_nb=>gt\_text  into ls\_text  with table key text\_type = lv\_text\_type  langu = lv\_langu.  if sy-subrc eq 0.  et\_raw\_text = ls\_text-text\_tab.  return.  endif.  \*==========================================\*  \* Von der DB lesen \*  \*==========================================\*  clear: ls\_text.  ls\_text-text\_type = lv\_text\_type.  ls\_text-langu = lv\_langu.  clear: lv\_id.  case lv\_text\_type.  when 'ANREDE'.  lv\_id = 'ZFIINST\_MAIL\_ANREDE'.  when 'NACHFRAGE\_1'.  lv\_id = 'ZFIINST\_MAIL\_NACHFRAGE\_1'.  when 'NACHFRAGE\_2'.  lv\_id = 'ZFIINST\_MAIL\_NACHFRAGE\_2'.  when 'ABSCHLUSS'.  lv\_id = 'ZFIINST\_MAIL\_ABSCHLUSS'.  when others.  clear: lv\_id.  endcase.  refresh: ls\_text-text\_tab.  if not lv\_id is initial.  ls\_text-text\_tab = zbccl\_text=>read\_text\_to\_string\_table(  i\_name = lv\_id  i\_id = zficl\_nb=>co\_text\_id\_zins "ZINS für Benachr. Inst.  i\_object = zficl\_nb=>co\_text\_object\_zins "Text  i\_spras = lv\_langu ).  endif.  \*===========================================\*  \* Export \*  \*===========================================\*  et\_raw\_text = ls\_text-text\_tab.  \*\*\*  \*===========================================\*  \* Buffer \*  \*===========================================\*  insert ls\_text into table zficl\_nb=>gt\_text.  endmethod.  method mail\_inst\_nachfrage.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Mail: Nachfrage aufbauen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lt\_parameter\_value type wdr\_simple\_name\_value\_map,  ls\_parameter\_value like line of lt\_parameter\_value,  ls\_daten type zfi\_s\_nb\_benach\_mahndruck,  lt\_text type string\_table.  data: ls\_nachfrage type string,  lv\_langu type langu.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Import Daten \*  \*==========================================\*  ls\_daten = is\_daten.  lv\_langu = ls\_daten-spras.  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_text\_nachfrage.  \*\*  \*\*==========================================\*  \* Nachfrage übernehmen \*  \*===========================================\*  clear ls\_nachfrage.  \*==========================================\*  \* Parametertabelle füllen \*  \*==========================================\*  refresh: lt\_parameter\_value.  \*\*  \* Rechnung  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `BELNR`.  ls\_parameter\_value-value = ls\_daten-belnr.  insert ls\_parameter\_value into table lt\_parameter\_value.  \* Name1  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `NAME1`.  ls\_parameter\_value-value = ls\_daten-name1.  insert ls\_parameter\_value into table lt\_parameter\_value.  \* Mahner  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `MAHNER`.  ls\_parameter\_value-value = ls\_daten-mahner\_email.  insert ls\_parameter\_value into table lt\_parameter\_value.  \*\*\*==========================================\*  \*\*\* Mailtext ermitteln \*  \*\*\*==========================================\*  refresh: lt\_text.  \* Wenn Betrag > CHF 1000 Nachfrage\_2, sonst Nachfrage\_1  if ls\_daten-dmbtr > 1000.  lt\_text = zficl\_nb=>mail\_inst\_process\_text(  iv\_text\_type = zficl\_nb=>co\_text\_nachfrage\_2  it\_parameter\_value = lt\_parameter\_value  iv\_langu = lv\_langu ).  else.  lt\_text = zficl\_nb=>mail\_inst\_process\_text(  iv\_text\_type = zficl\_nb=>co\_text\_nachfrage\_1  it\_parameter\_value = lt\_parameter\_value  iv\_langu = lv\_langu ).  endif.  if lt\_text is initial.  return.  endif.  \*\*  \*\*\*==========================================\*  \*\*\* Export \*  \*\*\*==========================================\*  et\_text\_nachfrage = lt\_text.  endmethod.  method MAIL\_INST\_PROCESS\_TEXT.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Prozess: Mailtext aufbereiten \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lt\_text type string\_table.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_text.  \*==========================================\*  \* Import \*  \*==========================================\*  if iv\_text\_type is initial.  return.  endif.  \*==========================================\*  \* Text \*  \*==========================================\*  refresh: lt\_text.  lt\_text = zficl\_nb=>mail\_inst\_get\_raw\_text(  iv\_text\_type = iv\_text\_type  iv\_langu = iv\_langu ).  if lt\_text is initial.  return.  endif.  \*==========================================\*  \* Parameter ersetzen \*  \*==========================================\*  if not it\_parameter\_value is initial.  et\_text = zficl\_nb=>mail\_replace\_parameter(  it\_parameter\_value = it\_parameter\_value  it\_text\_raw = lt\_text ).  else.  et\_text = lt\_text.  endif.  endmethod.  method mail\_inst\_send.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Rechnungsversand unechte Forderungen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* 02.08.2020: Absender fix: 'debitoren@fc.ethz.ch'.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: ls\_daten type zfi\_s\_nb\_benach\_mahndruck,  lt\_attach type zfi\_t\_nb\_benach\_attachment,  ls\_attach type zfi\_s\_nb\_benach\_attachment,  lv\_test type char1,  lv\_langu type langu.  data: lt\_parameter\_value type wdr\_simple\_name\_value\_map,  ls\_parameter\_value like line of lt\_parameter\_value,  ls\_adressdaten type zfi\_s\_nb\_adressdaten.  data: lt\_recipient\_to type string\_table,  lt\_recipient\_cc type string\_table,  lv\_sender type ad\_smtpadr,  lv\_empfaenger type ad\_smtpadr.  data: l\_adr type ad\_addrnum,  l\_per type ad\_persnum.  data: lt\_text\_hlp type string\_table,  lt\_text type string\_table,  ls\_text like line of lt\_text,  lt\_mailtext type bcsy\_text,  ls\_mailtext like line of lt\_mailtext.  data: lv\_subject type string,  lv\_string type string,  lv\_data\_xstring type xstring,  lt\_data\_xstring type table of xstring.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Import \*  \*==========================================\*  ls\_daten = is\_daten.  lt\_attach = it\_attach.  lv\_test = iv\_test.  \*==========================================\*  \* Email aufbereiten \*  \*==========================================\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Sprache ablegen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  lv\_langu = ls\_daten-spras.  \*------------------------------------------\*  \* Mail aufbereiten \*  \*------------------------------------------\*  \* Begrüssung gemäss Sprache \*  \*------------------------------------------\*  refresh:  lt\_text\_hlp.  call method zficl\_nb=>mail\_inst\_begruessung  exporting  is\_daten = ls\_daten  receiving  et\_text\_anrede = lt\_text\_hlp.  \*\*  if not lt\_text\_hlp is initial.  insert lines of lt\_text\_hlp into table lt\_text.  endif.  \*\*\* leerzeile  clear: lv\_string.  lv\_string = `<br>`.  insert lv\_string into table lt\_text.  \*\*\*\*\*\*\*------------------------------------------\*  \*\*\*\*\*\*\* Nachfrage an Rechnungssteller \*  \*\*\*\*\*\*\*------------------------------------------\*  refresh:  lt\_text\_hlp.  call method zficl\_nb=>mail\_inst\_nachfrage  exporting  is\_daten = ls\_daten  receiving  et\_text\_nachfrage = lt\_text\_hlp.  if not lt\_text\_hlp is initial.  insert lines of lt\_text\_hlp into table lt\_text.  endif.  \* leerzeile  clear: lv\_string.  lv\_string = `<br>`.  insert lv\_string into table lt\_text.  \*------------------------------------------\*  \* Abschluss gemäss Sprache \*  \*------------------------------------------\*  call method zficl\_nb=>mail\_inst\_abschluss  exporting  is\_daten = ls\_daten  receiving  et\_text\_abschluss = lt\_text\_hlp.  if not lt\_text\_hlp is initial.  insert lines of lt\_text\_hlp into table lt\_text.  endif.  \*===========================================\*  \* Mail versenden \*  \*===========================================\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Empfänger \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------\*  \* Empfänger TO \*  \*-------------------------------------------\*  refresh: lt\_recipient\_to.  \* Empfänger bestimmen, (Test an Mahner, Echt an Erfasser)  clear lv\_string.  \* Test ja/nein  if lv\_test = 'X'. "mit Testflag an Mahner senden  call method zficl\_nb=>get\_adressdaten  exporting  iv\_bname = sy-uname  importing  es\_adressdaten = ls\_adressdaten.  \*  lv\_string = ls\_adressdaten-email.  else.  lv\_string = ls\_daten-mail.  endif.  \* Empfänger übernehmen in Empfänger liste  insert lv\_string into table lt\_recipient\_to.  \*-------------------------------------------\*  \* Empfänger CC: Delegierte \*  \*-------------------------------------------\*  \*\* REFRESH: lt\_recipient\_cc.  \*-------------------------------------------\*  \* Subject / Betreff \*  \*-------------------------------------------\*  clear: lv\_subject.  \* anlegen je nach Sprache  case lv\_langu.  when 'D'.  concatenate 'Ihre Rechnung wurde noch nicht bezahlt'  ls\_daten-belnr into lv\_subject separated by space.  when 'E'.  concatenate 'Unpaid Invoice'  ls\_daten-belnr into lv\_subject separated by space.  when others.  concatenate 'Ihre Rechnung wurde noch nicht bezahlt'  ls\_daten-belnr into lv\_subject separated by space.  endcase.  \*-------------------------------------------\*  \* Mailtext \*  \*-------------------------------------------\*  refresh: lt\_mailtext.  loop at lt\_text into ls\_text.  clear: ls\_mailtext.  ls\_mailtext-line = ls\_text.  insert ls\_mailtext into table lt\_mailtext.  endloop.  \*-------------------------------------------\*  \* Attachments für Mail (SD-Fakturen) \*  \*-------------------------------------------\*  \* einfach Tabelle weiterleiten,  \* Verarbeitung erfolgt in send\_bcs\_mail  \*-------------------------------------------\*  \* Sender eruieren \*  \*-------------------------------------------\*  \* Achtung Absender fix zugeteilt  \*\*\* Absender festlegen (SAPMAIL\_FFM4)  \*\* call method zficl\_nb=>get\_adressdaten  \*\* exporting  \*\* iv\_bname = co\_sender\_uname  \*\* importing  \*\* es\_adressdaten = ls\_adressdaten.  \*\*  \*\* lv\_sender = ls\_adressdaten-email.  lv\_sender = 'debitoren@fc.ethz.ch'.  \*-------------------------------------------\*  \* Mail verschicken \*  \*-------------------------------------------\*  try.  zficl\_nb=>send\_bcs\_mail(  exporting  iv\_email\_sender = lv\_sender  it\_body = lt\_mailtext  it\_attach = lt\_attach  iv\_subject = lv\_subject  iv\_send\_immediately = abap\_false "Mail mit Job verschicken  it\_recipient\_to = lt\_recipient\_to  it\_recipient\_cc = lt\_recipient\_cc  iv\_content\_type = 'HTM'  iv\_with\_commit\_work = 'X'  iv\_spras = is\_daten-spras ).  \* iv\_with\_commit\_work = iv\_with\_commit\_work ).  catch cx\_bcs ##no\_handler. " BCS: Allgemeine Ausnahmen  \* \*\*\* Ignorieren  endtry.  endmethod.  method mail\_natp\_get\_raw\_text.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Roh Text (also noch mit den Variablen) ermitteln \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lv\_langu type langu,  lv\_text\_type type string,  lv\_id type tdobname.  data: ls\_text like line of zficl\_nb=>gt\_text.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_raw\_text.  \*==========================================\*  \* Import \*  \*==========================================\*  clear: lv\_text\_type.  lv\_text\_type = iv\_text\_type.  lv\_langu = iv\_langu.  \* \*\*\* Sprache: Nur Deutsch und Englisch  clear: lv\_langu.  case iv\_langu.  when 'D' or 'E'.  lv\_langu = iv\_langu.  when others.  lv\_langu = 'D'.  endcase.  \*==========================================\*  \* Buffer lesen \*  \*==========================================\*  clear: ls\_text.  read table zficl\_nb=>gt\_text  into ls\_text  with table key text\_type = lv\_text\_type  langu = lv\_langu.  if sy-subrc eq 0.  et\_raw\_text = ls\_text-text\_tab.  return.  endif.  \*==========================================\*  \* Von der DB lesen \*  \*==========================================\*  clear: ls\_text.  ls\_text-text\_type = lv\_text\_type.  ls\_text-langu = lv\_langu.  clear: lv\_id.  case lv\_text\_type.  when 'NATP\_INFO\_RW'.  lv\_id = 'ZFI\_NATP\_INFO\_RW'.  when 'NATP\_INFO\_RW\_P'.  lv\_id = 'ZFI\_NATP\_INFO\_RW\_P'.  when others.  clear: lv\_id.  endcase.  refresh: ls\_text-text\_tab.  if not lv\_id is initial.  ls\_text-text\_tab = zbccl\_text=>read\_text\_to\_string\_table(  i\_name = lv\_id  i\_id = zficl\_nb=>co\_text\_id\_st "ST  i\_object = zficl\_nb=>co\_text\_object\_zins "Text  i\_spras = lv\_langu ).  endif.  \*===========================================\*  \* Export \*  \*===========================================\*  et\_raw\_text = ls\_text-text\_tab.  \*\*\*  \*===========================================\*  \* Buffer \*  \*===========================================\*  insert ls\_text into table zficl\_nb=>gt\_text.  endmethod.  method mail\_natp\_information\_rw.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Mail: Nachfrage aufbauen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lt\_parameter\_value type wdr\_simple\_name\_value\_map,  ls\_parameter\_value like line of lt\_parameter\_value,  ls\_daten type zfi\_s\_nb\_natp\_info,  lt\_text type string\_table.  data: ls\_info type string,  lv\_langu type langu.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Import Daten \*  \*==========================================\*  ls\_daten = is\_daten.  lv\_langu = co\_langu\_natp.  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_text\_info.  \*\*  \*\*==========================================\*  \* Information an RW aufbauen \*  \*===========================================\*  clear ls\_info.  \*==========================================\*  \* Parametertabelle füllen \*  \*==========================================\*  refresh: lt\_parameter\_value.  \*\*  \* System (SYSID)  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `SYSTEM`.  ls\_parameter\_value-value = sy-sysid.  insert ls\_parameter\_value into table lt\_parameter\_value.  \* Eintrag (Zaehler)  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `ZAEHLER`.  ls\_parameter\_value-value = ls\_daten-zaehler.  insert ls\_parameter\_value into table lt\_parameter\_value.  \* ENAME (Nachname, Vorname)  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `ENAME`.  ls\_parameter\_value-value = ls\_daten-ename.  insert ls\_parameter\_value into table lt\_parameter\_value.  \* Name\_Auftritt  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `AUFTRITT`.  ls\_parameter\_value-value = ls\_daten-name\_auftritt.  insert ls\_parameter\_value into table lt\_parameter\_value.  \*\*\*==========================================\*  \*\*\* Mailtext ermitteln \*  \*\*\*==========================================\*  refresh: lt\_text.  if sy-sysid = 'S4P'.  lt\_text = zficl\_nb=>mail\_natp\_process\_text(  iv\_text\_type = zficl\_nb=>co\_text\_natp\_info\_rw\_p  it\_parameter\_value = lt\_parameter\_value  iv\_langu = lv\_langu ).  else.  lt\_text = zficl\_nb=>mail\_natp\_process\_text(  iv\_text\_type = zficl\_nb=>co\_text\_natp\_info\_rw  it\_parameter\_value = lt\_parameter\_value  iv\_langu = lv\_langu ).  endif.  if lt\_text is initial.  return.  endif.  \*\*  \*\*\*==========================================\*  \*\*\* Export \*  \*\*\*==========================================\*  et\_text\_info = lt\_text.  endmethod.  method MAIL\_NATP\_PROCESS\_TEXT.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Prozess: Mailtext aufbereiten \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lt\_text type string\_table.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_text.  \*==========================================\*  \* Import \*  \*==========================================\*  if iv\_text\_type is initial.  return.  endif.  \*==========================================\*  \* Text \*  \*==========================================\*  refresh: lt\_text.  lt\_text = zficl\_nb=>mail\_natp\_get\_raw\_text(  iv\_text\_type = iv\_text\_type  iv\_langu = iv\_langu ).  if lt\_text is initial.  return.  endif.  \*==========================================\*  \* Parameter ersetzen \*  \*==========================================\*  if not it\_parameter\_value is initial.  et\_text = zficl\_nb=>mail\_replace\_parameter(  it\_parameter\_value = it\_parameter\_value  it\_text\_raw = lt\_text ).  else.  et\_text = lt\_text.  endif.  endmethod.  method mail\_natp\_send.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Versand Selbständigkeitsnachweis --> Anlegen BP \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: ls\_info type zfi\_s\_nb\_natp\_info,  lt\_attach type zfi\_t\_nb\_benach\_attachment,  ls\_attach type zfi\_s\_nb\_benach\_attachment.  data: lt\_parameter\_value type wdr\_simple\_name\_value\_map,  ls\_parameter\_value like line of lt\_parameter\_value,  ls\_adressdaten type zfi\_s\_nb\_adressdaten.  data: lt\_recipient\_to type string\_table,  lt\_recipient\_cc type string\_table,  lv\_sender type ad\_smtpadr,  lv\_empfaenger\_rw type ad\_smtpadr,  lv\_empfaenger\_hr type ad\_smtpadr,  lv\_langu type spras.  data: lt\_text\_hlp type string\_table,  lt\_text type string\_table,  ls\_text like line of lt\_text,  lt\_mailtext type bcsy\_text,  ls\_mailtext like line of lt\_mailtext.  data: lv\_subject type string,  lv\_string type string,  lv\_data\_xstring type xstring,  lt\_data\_xstring type table of xstring.  constants: co\_empf\_niw type ad\_smtpadr value 'niederer@id.ethz.ch',  co\_empf\_rw type ad\_smtpadr value 'vendor\_accounts@fc.ethz.ch',  co\_sender type ad\_smtpadr value 'no-response@fc.ethz.ch'.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  refresh lt\_attach. clear lt\_attach.  \*==========================================\*  \* Import \*  \*==========================================\*  ls\_info = is\_natp\_info.  \*==========================================\*  \* Email aufbereiten \*  \*==========================================\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Sprache ablegen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  lv\_langu = co\_langu\_natp.  \*------------------------------------------\*  \* Mail aufbereiten \*  \*------------------------------------------\*  \*\*\*------------------------------------------\*  \*\*\* Info an RW \*  \*\*\*------------------------------------------\*  refresh:  lt\_text\_hlp.  call method zficl\_nb=>mail\_natp\_information\_rw  exporting  is\_daten = ls\_info  receiving  et\_text\_info = lt\_text\_hlp.  if not lt\_text\_hlp is initial.  insert lines of lt\_text\_hlp into table lt\_text.  endif.  \* Leerzeile  clear: lv\_string.  lv\_string = `<br>`.  insert lv\_string into table lt\_text.  \*===========================================\*  \* Mail versenden \*  \*===========================================\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Empfänger \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------\*  \* Empfänger TO \*  \*-------------------------------------------\*  refresh: lt\_recipient\_to.  \* Empfänger ist fix  clear lv\_string.  if sy-sysid = 'S4P'.  lv\_string = co\_empf\_rw.  else.  lv\_string = co\_empf\_niw.  endif.  \* Empfänger übernehmen in Empfänger liste  insert lv\_string into table lt\_recipient\_to.  \*-------------------------------------------\*  \* Empfänger CC: Delegierte \*  \*-------------------------------------------\*  \*\* REFRESH: lt\_recipient\_cc.  \*-------------------------------------------\*  \* Subject / Betreff \*  \*-------------------------------------------\*  clear: lv\_subject.  lv\_subject = 'Selbständigen Nachweis'.  \*-------------------------------------------\*  \* Mailtext \*  \*-------------------------------------------\*  refresh: lt\_mailtext.  loop at lt\_text into ls\_text.  clear: ls\_mailtext.  ls\_mailtext-line = ls\_text.  insert ls\_mailtext into table lt\_mailtext.  endloop.  \*-------------------------------------------\*  \* Attachments für Mail \*  \*-------------------------------------------\*  \* einfach Tabelle weiterleiten,  \* Verarbeitung erfolgt in send\_bcs\_mail  \* keine Attachments !!  \*-------------------------------------------\*  \* Sender eruieren \*  \*-------------------------------------------\*  \* Absender festlegen fix (SAPMAIL\_FFM5)  lv\_sender = co\_sender.  \*-------------------------------------------\*  \* Mail verschicken \*  \*-------------------------------------------\*  try.  zficl\_nb=>send\_bcs\_mail(  exporting  iv\_email\_sender = lv\_sender  it\_body = lt\_mailtext  it\_attach = lt\_attach  iv\_subject = lv\_subject  iv\_send\_immediately = abap\_false "Mail mit Job verschicken  it\_recipient\_to = lt\_recipient\_to  it\_recipient\_cc = lt\_recipient\_cc  iv\_content\_type = 'HTM'  iv\_with\_commit\_work = 'X'  iv\_spras = lv\_langu ).  \* iv\_with\_commit\_work = iv\_with\_commit\_work ).  catch cx\_bcs ##no\_handler. " BCS: Allgemeine Ausnahmen  \* \*\*\* Ignorieren  endtry.  endmethod.  method MAIL\_REPLACE\_PARAMETER.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Parameter Ersetzung aus Roh Text \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  DATA: lt\_text TYPE string\_table,  ls\_parameter\_value LIKE LINE OF it\_parameter\_value.  DATA: lv\_name TYPE string.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Export \*  \*==========================================\*  REFRESH: et\_text\_format.  \*==========================================\*  \* Import \*  \*==========================================\*  REFRESH: lt\_text.  IF it\_parameter\_value IS INITIAL.  et\_text\_format = it\_text\_raw.  RETURN.  ENDIF.  lt\_text = it\_text\_raw.  \*==========================================\*  \* Parameter Ersetzung \*  \*==========================================\*  LOOP AT it\_parameter\_value INTO ls\_parameter\_value.  lv\_name = `[` && to\_upper( ls\_parameter\_value-name ) && `]`.  REPLACE ALL OCCURRENCES OF lv\_name IN TABLE lt\_text WITH ls\_parameter\_value-value.  ENDLOOP.  \*==========================================\*  \* Export \*  \*==========================================\*  et\_text\_format = lt\_text.  endmethod.  method mail\_unfo\_abschluss.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Mail: Abschluss aufbauen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lt\_parameter\_value type wdr\_simple\_name\_value\_map,  ls\_parameter\_value like line of lt\_parameter\_value,  ls\_daten type zfi\_s\_nb\_unechte\_mahndruck,  lt\_text type string\_table.  data: ls\_abschluss type string,  lv\_langu type langu.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Import Daten \*  \*==========================================\*  ls\_daten = is\_daten.  lv\_langu = ls\_daten-spras.  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_text\_abschluss.  \*\*  \*\*==========================================\*  \* Aufbau des Abschluss \*  \*===========================================\*  \*\*\*==========================================\*  \*\*\* Parametertabelle füllen \*  \*\*\*==========================================\*  refresh: lt\_parameter\_value.  \*\*  \*\*\* Mahner  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `MAHNER`.  ls\_parameter\_value-value = sy-uname.  insert ls\_parameter\_value into table lt\_parameter\_value.  \*==========================================\*  \* Export \*  \*==========================================\*  \*\*\*==========================================\*  \*\*\* Mailtext ermitteln \*  \*\*\*==========================================\*  refresh: lt\_text.  lt\_text = zficl\_nb=>mail\_unfo\_process\_text(  iv\_text\_type = zficl\_nb=>co\_text\_abschluss  it\_parameter\_value = lt\_parameter\_value  iv\_langu = lv\_langu ).  if lt\_text is initial.  return.  endif.  \*==========================================\*  \* Export \*  \*==========================================\*  et\_text\_abschluss = lt\_text.  endmethod.  method mail\_unfo\_begruessung.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Mail: Begrüssung aufbauen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lt\_parameter\_value type wdr\_simple\_name\_value\_map,  ls\_parameter\_value like line of lt\_parameter\_value,  ls\_daten type zfi\_s\_nb\_unechte\_mahndruck,  lt\_text type string\_table.  data: ls\_anrede type string,  lv\_langu type langu.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Import Daten \*  \*==========================================\*  ls\_daten = is\_daten.  lv\_langu = ls\_daten-spras.  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_text\_anrede.  \*\*  \*\*==========================================\*  \* Aufbau der Anrede \*  \*===========================================\*  case lv\_langu.  when 'D'.  concatenate 'Sehr geehrte/r' ls\_daten-ename  into ls\_anrede separated by space.  when 'E'.  concatenate 'Dear' ls\_daten-ename  into ls\_anrede separated by space.  when others.  concatenate 'Sehr geehrte/r' ls\_daten-ename  into ls\_anrede separated by space.  endcase.  \*\*\*==========================================\*  \*\*\* Parametertabelle füllen \*  \*\*\*==========================================\*  refresh: lt\_parameter\_value.  \*\*  \*\*\* Anrede  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `ANREDE`.  ls\_parameter\_value-value = ls\_anrede.  insert ls\_parameter\_value into table lt\_parameter\_value.  \*==========================================\*  \* Export \*  \*==========================================\*  \*\*\*==========================================\*  \*\*\* Mailtext ermitteln \*  \*\*\*==========================================\*  refresh: lt\_text.  lt\_text = zficl\_nb=>mail\_unfo\_process\_text(  iv\_text\_type = zficl\_nb=>co\_text\_anrede  it\_parameter\_value = lt\_parameter\_value  iv\_langu = lv\_langu ).  if lt\_text is initial.  return.  endif.  \*==========================================\*  \* Export \*  \*==========================================\*  et\_text\_anrede = lt\_text.  endmethod.  method mail\_unfo\_get\_raw\_text.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Roh Text (also noch mit den Variablen) ermitteln \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lv\_langu type langu,  lv\_text\_type type string,  lv\_id type tdobname.  data: ls\_text like line of zficl\_nb=>gt\_text.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_raw\_text.  \*==========================================\*  \* Import \*  \*==========================================\*  clear: lv\_text\_type.  lv\_text\_type = iv\_text\_type.  lv\_langu = iv\_langu.  \* \*\*\* Sprache: Nur Deutsch und Englisch  clear: lv\_langu.  case iv\_langu.  when 'D' or 'E'.  lv\_langu = iv\_langu.  when others.  lv\_langu = 'D'.  endcase.  \*==========================================\*  \* Buffer lesen \*  \*==========================================\*  clear: ls\_text.  read table zficl\_nb=>gt\_text  into ls\_text  with table key text\_type = lv\_text\_type  langu = lv\_langu.  if sy-subrc eq 0.  et\_raw\_text = ls\_text-text\_tab.  return.  endif.  \*==========================================\*  \* Von der DB lesen \*  \*==========================================\*  clear: ls\_text.  ls\_text-text\_type = lv\_text\_type.  ls\_text-langu = lv\_langu.  clear: lv\_id.  case lv\_text\_type.  when 'ANREDE'.  lv\_id = 'ZFIUNFO\_MAIL\_ANREDE'.  when 'NACHFRAGE'.  lv\_id = 'ZFIUNFO\_MAIL\_NACHFRAGE'.  when 'ABSCHLUSS'.  lv\_id = 'ZFIUNFO\_MAIL\_ABSCHLUSS'.  when others.  clear: lv\_id.  endcase.  refresh: ls\_text-text\_tab.  if not lv\_id is initial.  ls\_text-text\_tab = zbccl\_text=>read\_text\_to\_string\_table(  i\_name = lv\_id  i\_id = zficl\_nb=>co\_text\_id\_zunf "ZUNF für unechte Ford.  i\_object = zficl\_nb=>co\_text\_object\_zunf "Text  i\_spras = lv\_langu ).  endif.  \*===========================================\*  \* Export \*  \*===========================================\*  et\_raw\_text = ls\_text-text\_tab.  \*\*\*  \*===========================================\*  \* Buffer \*  \*===========================================\*  insert ls\_text into table zficl\_nb=>gt\_text.  endmethod.  method mail\_unfo\_nachfrage.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Mail: Nachfrage aufbauen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lt\_parameter\_value type wdr\_simple\_name\_value\_map,  ls\_parameter\_value like line of lt\_parameter\_value,  ls\_daten type zfi\_s\_nb\_unechte\_mahndruck,  lt\_text type string\_table.  data: ls\_nachfrage type string,  lv\_langu type langu.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Import Daten \*  \*==========================================\*  ls\_daten = is\_daten.  lv\_langu = ls\_daten-spras.  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_text\_nachfrage.  \*\*  \*\*==========================================\*  \* Nachfrage übernehmen \*  \*===========================================\*  clear ls\_nachfrage.  \*==========================================\*  \* Parametertabelle füllen \*  \*==========================================\*  refresh: lt\_parameter\_value.  \*\*  \* Rechnung  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `BELNR`.  ls\_parameter\_value-value = ls\_daten-belnr.  insert ls\_parameter\_value into table lt\_parameter\_value.  \*  \* Fonds  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `PSP`.  ls\_parameter\_value-value = ls\_daten-projk.  insert ls\_parameter\_value into table lt\_parameter\_value.  \* Name1  clear: ls\_parameter\_value.  ls\_parameter\_value-name = `NAME1`.  ls\_parameter\_value-value = ls\_daten-name1.  insert ls\_parameter\_value into table lt\_parameter\_value.  \*\*\*==========================================\*  \*\*\* Mailtext ermitteln \*  \*\*\*==========================================\*  refresh: lt\_text.  lt\_text = zficl\_nb=>mail\_unfo\_process\_text(  iv\_text\_type = zficl\_nb=>co\_text\_nachfrage  it\_parameter\_value = lt\_parameter\_value  iv\_langu = lv\_langu ).  if lt\_text is initial.  return.  endif.  \*\*  \*\*\*==========================================\*  \*\*\* Export \*  \*\*\*==========================================\*  et\_text\_nachfrage = lt\_text.  endmethod.  method mail\_unfo\_process\_text.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Prozess: Mailtext aufbereiten \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lt\_text type string\_table.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Export \*  \*==========================================\*  refresh: et\_text.  \*==========================================\*  \* Import \*  \*==========================================\*  if iv\_text\_type is initial.  return.  endif.  \*==========================================\*  \* Text \*  \*==========================================\*  refresh: lt\_text.  lt\_text = zficl\_nb=>mail\_unfo\_get\_raw\_text(  iv\_text\_type = iv\_text\_type  iv\_langu = iv\_langu ).  if lt\_text is initial.  return.  endif.  \*==========================================\*  \* Parameter ersetzen \*  \*==========================================\*  if not it\_parameter\_value is initial.  et\_text = zficl\_nb=>mail\_replace\_parameter(  it\_parameter\_value = it\_parameter\_value  it\_text\_raw = lt\_text ).  else.  et\_text = lt\_text.  endif.  endmethod.  method mail\_unfo\_send.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Rechnungsversand unechte Forderungen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* 02.08.2020: Absender fix: 'debitoren@fc.ethz.ch'.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: ls\_daten type zfi\_s\_nb\_unechte\_mahndruck,  lt\_attach type zfi\_t\_nb\_unechte\_attachment,  ls\_attach type zfi\_s\_nb\_unechte\_attachment,  lv\_test type char1,  lv\_langu type langu.  data: lt\_parameter\_value type wdr\_simple\_name\_value\_map,  ls\_parameter\_value like line of lt\_parameter\_value,  ls\_adressdaten type zfi\_s\_nb\_adressdaten.  data: lt\_recipient\_to type string\_table,  lt\_recipient\_cc type string\_table,  lv\_sender type ad\_smtpadr,  lv\_empfaenger type ad\_smtpadr.  data: l\_adr type ad\_addrnum,  l\_per type ad\_persnum.  data: lt\_text\_hlp type string\_table,  lt\_text type string\_table,  ls\_text like line of lt\_text,  lt\_mailtext type bcsy\_text,  ls\_mailtext like line of lt\_mailtext.  data: lv\_subject type string,  lv\_string type string,  lv\_data\_xstring type xstring,  lt\_data\_xstring type table of xstring.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Import \*  \*==========================================\*  ls\_daten = is\_daten.  lt\_attach = it\_attach.  lv\_test = iv\_test.  \*==========================================\*  \* Email aufbereiten \*  \*==========================================\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Sprache ablegen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  lv\_langu = ls\_daten-spras.  \*------------------------------------------\*  \* Mail aufbereiten \*  \*------------------------------------------\*  \* Begrüssung gemäss Sprache \*  \*------------------------------------------\*  refresh:  lt\_text\_hlp.  call method zficl\_nb=>mail\_unfo\_begruessung  exporting  is\_daten = ls\_daten  receiving  et\_text\_anrede = lt\_text\_hlp.  \*\*  if not lt\_text\_hlp is initial.  insert lines of lt\_text\_hlp into table lt\_text.  endif.  \*\*\* leerzeile  clear: lv\_string.  lv\_string = `<br>`.  insert lv\_string into table lt\_text.  \*\*\*\*------------------------------------------\*  \*\*\*\* Nachfrage an Rechnungssteller \*  \*\*\*\*------------------------------------------\*  refresh:  lt\_text\_hlp.  call method zficl\_nb=>mail\_unfo\_nachfrage  exporting  is\_daten = ls\_daten  receiving  et\_text\_nachfrage = lt\_text\_hlp.  if not lt\_text\_hlp is initial.  insert lines of lt\_text\_hlp into table lt\_text.  endif.  \*\*\* leerzeile  clear: lv\_string.  lv\_string = `<br>`.  insert lv\_string into table lt\_text.  \*------------------------------------------\*  \* Abschluss gemäss Sprache \*  \*------------------------------------------\*  call method zficl\_nb=>mail\_unfo\_abschluss  exporting  is\_daten = ls\_daten  receiving  et\_text\_abschluss = lt\_text\_hlp.  if not lt\_text\_hlp is initial.  insert lines of lt\_text\_hlp into table lt\_text.  endif.  \*===========================================\*  \* Mail versenden \*  \*===========================================\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Empfänger \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------\*  \* Empfänger TO \*  \*-------------------------------------------\*  refresh: lt\_recipient\_to.  \* Empfänger bestimmen, (Test an Mahner, Echt an Erfasser)  clear lv\_string.  \* Test ja/nein  if lv\_test = 'X'. "mit Testflag an Mahner senden  call method zficl\_nb=>get\_adressdaten  exporting  iv\_bname = sy-uname  importing  es\_adressdaten = ls\_adressdaten.  \*  lv\_string = ls\_adressdaten-email.  else.  lv\_string = ls\_daten-mail.  endif.  \* Empfänger übernehmen in Empfänger liste  insert lv\_string into table lt\_recipient\_to.  \*-------------------------------------------\*  \* Empfänger CC: Delegierte \*  \*-------------------------------------------\*  \*\* REFRESH: lt\_recipient\_cc.  \*-------------------------------------------\*  \* Subject / Betreff \*  \*-------------------------------------------\*  clear: lv\_subject.  \* anlegen je nach Sprache  case lv\_langu.  when 'D'.  concatenate 'Ihre Rechnung wurde noch nicht bezahlt'  ls\_daten-belnr into lv\_subject separated by space.  when 'E'.  concatenate 'Unpaid Invoice'  ls\_daten-belnr into lv\_subject separated by space.  when others.  concatenate 'Ihre Rechnung wurde noch nicht bezahlt'  ls\_daten-belnr into lv\_subject separated by space.  endcase.  \*-------------------------------------------\*  \* Mailtext \*  \*-------------------------------------------\*  refresh: lt\_mailtext.  loop at lt\_text into ls\_text.  clear: ls\_mailtext.  ls\_mailtext-line = ls\_text.  insert ls\_mailtext into table lt\_mailtext.  endloop.  \*-------------------------------------------\*  \* Attachments für Mail (SD-Fakturen) \*  \*-------------------------------------------\*  \* einfach Tabelle weiterleiten,  \* Verarbeitung erfolgt in send\_bcs\_mail  \*-------------------------------------------\*  \* Sender eruieren \*  \*-------------------------------------------\*  \* Achtung; es wird fix zugeteilt  lv\_sender = 'debitoren@fc.ethz.ch'.  \*-------------------------------------------\*  \* Mail verschicken \*  \*-------------------------------------------\*  try.  zficl\_nb=>send\_bcs\_mail(  exporting  iv\_email\_sender = lv\_sender  it\_body = lt\_mailtext  it\_attach = lt\_attach  iv\_subject = lv\_subject  iv\_send\_immediately = abap\_false "Mail mit Job verschicken  it\_recipient\_to = lt\_recipient\_to  it\_recipient\_cc = lt\_recipient\_cc  iv\_content\_type = 'HTM'  iv\_with\_commit\_work = 'X'  iv\_spras = is\_daten-spras ).  \* iv\_with\_commit\_work = iv\_with\_commit\_work ).  catch cx\_bcs ##no\_handler. " BCS: Allgemeine Ausnahmen  \* \*\*\* Ignorieren  endtry.  endmethod.  method send\_bcs\_mail.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Mailversand mittels BCS Funktionalität \*  \* - Absender: \*  \* - IV\_EMAIL\_SENDER: Emailadresse eines im SAP vor- \*  \* handenen Benutzers \*  \* - IV\_USER\_SENDER: Den Benutzer als Absender nehmen \*  \* ==> Ist keine Angabe vorhanden, so wird der aktuelle \*  \* Benutzer als Absender genommen \*  \* - Mailtext: Dieser muss als Tabelle mittels Parameter \*  \* IT\_BODY übergeben werden. Ist die Tabelle leer, so \*  \* wird auch kein Mail erstellt \*  \* - Betreff: Dieser muss als Parameter IV\_SUBJECT über- \*  \* geben werden. Extra lange Betreff bis 255 Zeichen \*  \* werden unterstützt \*  \* - Wichtigkeit: Diese muss als Parameter IV\_IMPORTANCE \*  \* übergeben werden. Dabei ist 1 wichtig, 5 Standard \*  \* und 9 unwichtig \*  \* - Mail sofort verschicken: Im Normalfall wird das Mail \*  \* mit dem regelmässig eingeplanten SEND\_MAIL Job ver- \*  \* sendet. Soll das Mail aber sofort verschickt werden, \*  \* so muss der Parameter IV\_SEND\_IMMEDIATELY gesetzt \*  \* werden \*  \* - Empfänger: Als Empfänger werden mit dieser Methode \*  \* nur Emailadressen unterstützt. Mehrfach vorhandene \*  \* Emailadressen werden aussortiert. Hinweis: Die \*  \* Tabelle unterstützen im Prinzip unendlich lange \*  \* Emailadressen, die maximal unterstütze Länge ist je- \*  \* doch auf 241 Zeichen beschränkt. \*  \* - IT\_RECIPIENT\_TO: Hauptempfänger (TO) \*  \* - IT\_RECIPIENT\_CC: CC-Empfänger \*  \* - IT\_RECIPIENT\_BCC: Blind Empfänger (BCC) \*  \* - Content Type: Nur die Typen RAW und HTM werden unter-\*  \* stützt. Parameter IV\_CONTENT\_TYPE \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinition \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  types: begin of ls\_ty\_recipient,  mail type ad\_smtpadr,  end of ls\_ty\_recipient.  data: lr\_send\_request type ref to cl\_bcs,  lr\_document type ref to cl\_document\_bcs,  lr\_sender type ref to cl\_cam\_address\_bcs,  lr\_sender\_user type ref to cl\_sapuser\_bcs,  lr\_receiver type ref to cl\_cam\_address\_bcs.  data: lv\_subject type string,  lv\_subject\_hlp type so\_obj\_des,  lv\_importance type bcs\_docimp,  lv\_recipient\_string type string,  lv\_content\_type type so\_obj\_tp,  lv\_email\_sender type ad\_smtpadr,  lv\_uname type sy-uname,  lv\_spras type langu.  data: pdf\_size type so\_obj\_len,  pdf\_content type solix\_tab,  pdf\_xstring type xstring.  data: lt\_recipient\_to type standard table of ls\_ty\_recipient,  lt\_recipient\_cc type standard table of ls\_ty\_recipient,  lt\_recipient\_bcc type standard table of ls\_ty\_recipient,  ls\_recipient type ls\_ty\_recipient,  lt\_recipient\_multiple type hashed table of ls\_ty\_recipient  with unique key mail,  lt\_attach type zfi\_t\_nb\_unechte\_attachment,  ls\_attach type zfi\_s\_nb\_unechte\_attachment,  iv\_data\_xstring type xstring.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*==========================================\*  \* Import \*  \*==========================================\*  \* \*\*\* Mailtext  if it\_body is initial.  raise exception type cx\_bcs  exporting  error\_type = cx\_bcs=>parameter\_error.  endif.  \* \*\*\* Content Type  lv\_content\_type = iv\_content\_type.  if not ( lv\_content\_type = 'RAW' or  lv\_content\_type = 'HTM' ).  raise exception type cx\_bcs  exporting  error\_type = cx\_bcs=>parameter\_error.  endif.  \* \*\*\* Betreff  clear: lv\_subject,  lv\_subject\_hlp.  lv\_subject = iv\_subject.  lv\_subject\_hlp = iv\_subject.  \* \*\*\* Wichtigkeit  clear: lv\_importance.  if iv\_importance = '1' or  iv\_importance = '5' or  iv\_importance = '9'.  lv\_importance = iv\_importance.  else.  lv\_importance = '5'. "Default  endif.  \* \*\*\* Sprache  lv\_spras = iv\_spras.  \* \*\*\* Empfänger  refresh: lt\_recipient\_to,  lt\_recipient\_cc,  lt\_recipient\_bcc,  lt\_recipient\_multiple.  loop at it\_recipient\_to into lv\_recipient\_string.  \* \*\*\* Länge überprüfen  if strlen( lv\_recipient\_string ) > 241.  continue.  endif.  \* \*\*\* Multiple Check  ls\_recipient-mail = to\_upper( lv\_recipient\_string ).  insert ls\_recipient into table lt\_recipient\_multiple.  if sy-subrc eq 0.  \* \*\*\* Alles in Ordnung, in Tabelle schreiben  ls\_recipient-mail = lv\_recipient\_string.  insert ls\_recipient into table lt\_recipient\_to.  else.  continue.  endif.  endloop.  loop at it\_recipient\_cc into lv\_recipient\_string.  \* \*\*\* Länge überprüfen  if strlen( lv\_recipient\_string ) > 241.  continue.  endif.  \* \*\*\* Multiple Check  ls\_recipient-mail = to\_upper( lv\_recipient\_string ).  insert ls\_recipient into table lt\_recipient\_multiple.  if sy-subrc eq 0.  \* \*\*\* Alles in Ordnung, in Tabelle schreiben  ls\_recipient-mail = lv\_recipient\_string.  insert ls\_recipient into table lt\_recipient\_cc.  else.  continue.  endif.  endloop.  loop at it\_recipient\_bcc into lv\_recipient\_string.  \* \*\*\* Länge überprüfen  if strlen( lv\_recipient\_string ) > 241.  continue.  endif.  \* \*\*\* Multiple Check  ls\_recipient-mail = to\_upper( lv\_recipient\_string ).  insert ls\_recipient into table lt\_recipient\_multiple.  if sy-subrc eq 0.  \* \*\*\* Alles in Ordnung, in Tabelle schreiben  ls\_recipient-mail = lv\_recipient\_string.  insert ls\_recipient into table lt\_recipient\_bcc.  else.  continue.  endif.  endloop.  if lt\_recipient\_multiple is initial.  \* \*\*\* Keine Empfänger!  raise exception type cx\_bcs  exporting  error\_type = cx\_bcs=>parameter\_error.  else.  free: lt\_recipient\_multiple.  endif.  \*==========================================\*  \* Mail aufbauen \*  \*==========================================\*  \*------------------------------------------\*  \* BCS Instanz \*  \*------------------------------------------\*  clear: lr\_send\_request.  try.  lr\_send\_request = cl\_bcs=>create\_persistent( ).  endtry.  \*------------------------------------------\*  \* Dokument (Mailtext) \*  \*------------------------------------------\*  clear: lr\_document.  try.  lr\_document = cl\_document\_bcs=>create\_document(  i\_type = lv\_content\_type  i\_subject = lv\_subject\_hlp  i\_importance = lv\_importance  i\_text = it\_body ).  endtry.  try.  lr\_send\_request->set\_document( lr\_document ).  endtry.  \*------------------------------------------\*  \* PDF-als Document ausgeben (BCS\_EXAMPLE\_8)\*  \*------------------------------------------\*  \* Holen der Dokumentes aus Archiv und erstellen XString  loop at it\_attach into ls\_attach.  call method zficl\_nb=>get\_data\_from\_archiv  exporting  iv\_archiv\_id = ls\_attach-archiv\_id  iv\_arc\_doc\_id = ls\_attach-arc\_doc\_id  iv\_document\_type = 'PDF'  receiving  ev\_data\_xstring = iv\_data\_xstring.  \* Attachment erstellen  pdf\_content = cl\_document\_bcs=>xstring\_to\_solix( iv\_data\_xstring ).  case lv\_spras.  when 'D'.  call method lr\_document->add\_attachment  exporting  i\_attachment\_type = 'PDF'  i\_attachment\_subject = 'RE\_' && ls\_attach-belnr  i\_att\_content\_hex = pdf\_content.  when 'E'.  call method lr\_document->add\_attachment  exporting  i\_attachment\_type = 'PDF'  i\_attachment\_subject = 'Invoice\_' && ls\_attach-belnr  i\_att\_content\_hex = pdf\_content.  when others.  call method lr\_document->add\_attachment  exporting  i\_attachment\_type = 'PDF'  i\_attachment\_subject = 'RE\_' && ls\_attach-belnr  i\_att\_content\_hex = pdf\_content.  endcase.  endloop.  \*------------------------------------------\*  \* Extralanger Betreff \*  \*------------------------------------------\*  try.  lr\_send\_request->set\_message\_subject( lv\_subject ).  endtry.  \*------------------------------------------\*  \* Sender \*  \*------------------------------------------\*  if not iv\_email\_sender is initial.  \* \*\*\* Email Absender  clear: lv\_email\_sender.  lv\_email\_sender = iv\_email\_sender.  try.  clear: lr\_sender.  lr\_sender = cl\_cam\_address\_bcs=>create\_internet\_address(  i\_address\_string = lv\_email\_sender ).  endtry.  try.  lr\_send\_request->set\_sender( i\_sender = lr\_sender ).  endtry.  else.  \* \*\*\* Benutzer Absender  clear: lv\_uname.  if not iv\_user\_sender is initial.  lv\_uname = iv\_user\_sender.  else.  lv\_uname = sy-uname.  endif.  clear: lr\_sender\_user.  try.  lr\_sender\_user = cl\_sapuser\_bcs=>create( lv\_uname ).  endtry.  try.  lr\_send\_request->set\_sender( i\_sender = lr\_sender\_user ).  endtry.  endif.  \*------------------------------------------\*  \* Empfänger \*  \*------------------------------------------\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* TO \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  loop at lt\_recipient\_to into ls\_recipient.  try.  clear: lr\_receiver.  lr\_receiver = cl\_cam\_address\_bcs=>create\_internet\_address( ls\_recipient-mail ).  endtry.  try.  lr\_send\_request->add\_recipient( exporting i\_recipient = lr\_receiver ).  endtry.  endloop.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* CC \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  loop at lt\_recipient\_cc into ls\_recipient.  try.  clear: lr\_receiver.  lr\_receiver = cl\_cam\_address\_bcs=>create\_internet\_address( ls\_recipient-mail ).  endtry.  try.  lr\_send\_request->add\_recipient( exporting i\_recipient = lr\_receiver  i\_copy = abap\_true ).  endtry.  endloop.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* BCC \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  loop at lt\_recipient\_bcc into ls\_recipient.  try.  clear: lr\_receiver.  lr\_receiver = cl\_cam\_address\_bcs=>create\_internet\_address( ls\_recipient-mail ).  endtry.  try.  lr\_send\_request->add\_recipient( exporting i\_recipient = lr\_receiver  i\_blind\_copy = abap\_true ).  endtry.  endloop.  \*------------------------------------------\*  \* Sofort versenden? \*  \*------------------------------------------\*  try.  lr\_send\_request->set\_send\_immediately( iv\_send\_immediately ).  endtry.  \*------------------------------------------\*  \* Verschlüsselung \*  \*------------------------------------------\*  try.  try.  lr\_send\_request->set\_sign( i\_sign = ' ' ).  \* exporting  \* i\_sign = 'X'.  catch cx\_send\_req\_bcs.  endtry.  endtry.  \*==========================================\*  \* Mail versenden \*  \*==========================================\*  try.  if lr\_send\_request->send( ) = abap\_true.  if iv\_with\_commit\_work = abap\_true.  commit work and wait.  endif.  else.  if iv\_with\_commit\_work = abap\_true.  rollback work. "#EC CI\_ROLLBACK  endif.  endif.  endtry.  endmethod.  method SPERREN\_ENTSPERREN\_10K.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Hier werden die Programmsperren abgehandelt ZFIKFED10K \*  \* Mögliche Aktionen: \*  \* 'S' --> sperren \*  \* 'E' --> entsperren \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_action type char1.  data: lt\_enqueue type table of seqg3,  lv\_garg type seqg3-garg.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_action = iv\_action.  clear: ev\_lock\_uname,  rv\_gesperrt.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  case lv\_action.  when 'S'. "Sperren  call function 'ENQUEUE\_EZFI\_KRED10K'  exporting  mode\_zfi\_kred\_10k = 'E'  mandt = sy-mandt  bukrs = iv\_bukrs  gjahr = iv\_gjahr  belnr = iv\_belnr  buzei = iv\_buzei  \* X\_BUKRS = ' '  \* X\_GJAHR = ' '  \* X\_BELNR = ' '  \* X\_BUZEI = ' '  \_scope = '2'  \_wait = ' '  \_collect = ' '  exceptions  foreign\_lock = 1  system\_failure = 2  others = 3.  if sy-subrc <> 0.  rv\_gesperrt = abap\_true.  else.  rv\_gesperrt = abap\_false.  endif.  when 'E'. "Entsperren  call function 'DEQUEUE\_EZFI\_KRED10K'  exporting  mode\_zfi\_kred\_10k = 'E'  mandt = sy-mandt  bukrs = iv\_bukrs  gjahr = iv\_gjahr  belnr = iv\_belnr  buzei = iv\_buzei  \* X\_BUKRS = ' '  \* X\_GJAHR = ' '  \* X\_BELNR = ' '  \* X\_BUZEI = ' '  \_scope = '2'  \_synchron = ' '  \_collect = ' '.  when others.  lv\_garg = sy-mandt && iv\_bukrs && iv\_gjahr && iv\_belnr && iv\_buzei.  call function 'ENQUEUE\_READ'  exporting  gname = 'ZFI\_KRED\_10K'  garg = lv\_garg  guname = '\*'  tables  enq = lt\_enqueue  exceptions  communication\_failure = 1  system\_failure = 2  others = 3.  if sy-subrc <> 0.  rv\_gesperrt = abap\_false.  else.  if lt\_enqueue is initial.  rv\_gesperrt = abap\_false.  else.  read table lt\_enqueue into data(ls\_enqueue) index 1.  ev\_lock\_uname = ls\_enqueue-guname.  rv\_gesperrt = abap\_true.  endif.  endif.  endcase.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  endmethod.  method sperren\_entsperren\_ru.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Hier werden die Programmsperren abgehandelt ZFIKREDRU \*  \* Mögliche Aktionen: \*  \* 'S' --> sperren \*  \* 'E' --> entsperren \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_action type char1.  data: lt\_enqueue type table of seqg3,  lv\_garg type seqg3-garg.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_action = iv\_action.  clear: ev\_lock\_uname,  rv\_gesperrt.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  case lv\_action.  when 'S'. "Sperren  call function 'ENQUEUE\_EZFI\_KREDRU'  exporting  mode\_zfi\_kred\_ru = 'E'  mandt = sy-mandt  bukrs = iv\_bukrs  gjahr = iv\_gjahr  belnr = iv\_belnr  buzei = iv\_buzei  \* X\_BUKRS = ' '  \* X\_GJAHR = ' '  \* X\_BELNR = ' '  \* X\_BUZEI = ' '  \_scope = '2'  \_wait = ' '  \_collect = ' '  exceptions  foreign\_lock = 1  system\_failure = 2  others = 3.  if sy-subrc <> 0.  rv\_gesperrt = abap\_true.  else.  rv\_gesperrt = abap\_false.  endif.  when 'E'. "Entsperren  call function 'DEQUEUE\_EZFI\_KREDRU'  exporting  mode\_zfi\_kred\_ru = 'E'  mandt = sy-mandt  bukrs = iv\_bukrs  gjahr = iv\_gjahr  belnr = iv\_belnr  buzei = iv\_buzei  \* X\_BUKRS = ' '  \* X\_GJAHR = ' '  \* X\_BELNR = ' '  \* X\_BUZEI = ' '  \_scope = '2'  \_synchron = ' '  \_collect = ' '.  when others.  lv\_garg = sy-mandt && iv\_bukrs && iv\_gjahr && iv\_belnr && iv\_buzei.  call function 'ENQUEUE\_READ'  exporting  gname = 'ZFI\_KRED\_RU'  garg = lv\_garg  guname = '\*'  tables  enq = lt\_enqueue  exceptions  communication\_failure = 1  system\_failure = 2  others = 3.  if sy-subrc <> 0.  rv\_gesperrt = abap\_false.  else.  if lt\_enqueue is initial.  rv\_gesperrt = abap\_false.  else.  read table lt\_enqueue into data(ls\_enqueue) index 1.  ev\_lock\_uname = ls\_enqueue-guname.  rv\_gesperrt = abap\_true.  endif.  endif.  endcase.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  endmethod.  method upload\_dokument\_echt.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Upload Dokument \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinitionen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lv\_bp type bu\_partner,  lv\_filename\_in type string,  lv\_filename\_out type string,  lv\_xstring type xstring,  lv\_type\_ok type char1.  data: lt\_return type table of bapiret2,  ls\_return type bapiret2,  ls\_return1 type bapiret1.  data: lv\_msgv1 type sy-msgv1,  lv\_index type i,  lv\_info type string.  constants: co\_sap\_object\_bus1006 type saeobjid value 'BUS1006',  co\_docart\_zbpselbstn type saeobjart value 'ZBPSELBSTN'.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Daten übernehmen  lv\_bp = iv\_bp.  \* -----------------------------------------------------  \* Dokument holen ab Verzeichnis und als Xstring übernehmen  \* -----------------------------------------------------  call method zficl\_nb=>get\_fs\_openfile  exporting  iv\_filename = space  importing  ev\_filename = lv\_filename\_out  ev\_xstring = lv\_xstring.  \* -----------------------------------------------------  \* Prüfen, ob PDF-Dokument  \* -----------------------------------------------------  call method zbcattcl\_file=>check\_is\_extension\_xstring  exporting  iv\_file\_xstring = lv\_xstring  iv\_file\_extension = 'PDF'  \* iv\_do\_mapping = ABAP\_TRUE  \* IMPORTING  \* ev\_file\_extension\_effective =  receiving  rv\_ok = lv\_type\_ok.  if lv\_type\_ok = abap\_true.  \* -----------------------------------------------------  \* Xstring übernehmen in temp. Archiv  \* -----------------------------------------------------  call method zbcarchcl\_archive=>do\_insert\_data\_into\_archive  exporting  iv\_xstring = lv\_xstring  iv\_sap\_object = co\_sap\_object\_bus1006  iv\_docart = co\_docart\_zbpselbstn  iv\_objectid = lv\_bp  iv\_documenttype = 'PDF'  iv\_with\_commit\_work = 'X'  \* importing  \* es\_outdoc =  \* ev\_ok =  \* es\_return =  \* ev\_rc =  .  \*\*\* catch zcx\_bc\_ixos\_webviewer .  \*\*\* clear ls\_return.  \*\*\* ls\_return-type = 'E'.  \*\*\* ls\_return-id = 'ZFI\_NB'.  \*\*\* ls\_return-number = 075.  \*\*\* ls\_return-message = 'Dokument konnte nicht in internes Archiv übernommen werden'.  \*\*\* append ls\_return to lt\_return.  \*\*\*  \*\*\* endtry.  \*\*\*  \* -----------------------------------------------------  \* Aufbauen Fehlermeldung  \* -----------------------------------------------------  else.  clear ls\_return.  ls\_return-type = 'E'.  ls\_return-id = 'ZFI\_NB'.  ls\_return-number = 076.  ls\_return-message = 'Dokument ist kein PDF-Dokument'.  append ls\_return to lt\_return.  endif.  \*------------------------------------------------------  \* Export Parameter  \*------------------------------------------------------  et\_return = lt\_return.  endmethod.  method upload\_dokument\_tmp.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Upload Dokument \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Lokale Datendefinitionen \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  data: lv\_zaehler type zfi\_e\_nb\_laufnr,  lv\_filename\_in type string,  lv\_filename\_out type string,  lv\_xstring type xstring,  lv\_type\_ok type char1.  data: lt\_return type table of bapiret2,  ls\_return type bapiret2,  ls\_return1 type bapiret1,  ls\_temporary type zarc\_temporary.  data: lv\_msgv1 type sy-msgv1,  lv\_index type i,  lv\_info type string.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Coding \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Daten übernehmen  lv\_zaehler = iv\_zaehler.  \* -----------------------------------------------------  \* Dokument holen ab Verzeichnis und als Xstring übernehmen  \* -----------------------------------------------------  call method zficl\_nb=>get\_fs\_openfile  exporting  iv\_filename = space  importing  ev\_filename = lv\_filename\_out  ev\_xstring = lv\_xstring.  \* -----------------------------------------------------  \* Prüfen, ob PDF-Dokument  \* -----------------------------------------------------  call method zbcattcl\_file=>check\_is\_extension\_xstring  exporting  iv\_file\_xstring = lv\_xstring  iv\_file\_extension = 'PDF'  \* iv\_do\_mapping = ABAP\_TRUE  \* IMPORTING  \* ev\_file\_extension\_effective =  receiving  rv\_ok = lv\_type\_ok.  if lv\_type\_ok = abap\_true.  \* -----------------------------------------------------  \* Xstring übernehmen in temp. Archiv  \* -----------------------------------------------------  try.  call method zbcarchcl\_archive=>do\_insert\_data\_into\_temp\_arch  exporting  iv\_xstring = lv\_xstring  iv\_doc\_type = 'PDF'  iv\_application\_id = 'ZFI\_NP'  iv\_application\_key = lv\_zaehler  iv\_expire\_in = 2  iv\_time\_unit = 'JHR'  iv\_additional\_info = 'Selbständigkeitserklärung'  iv\_with\_commit\_work = 'X'  importing  es\_temporary = ls\_temporary.  catch zcx\_bcgen.  clear ls\_temporary.  endtry.  if ls\_temporary is initial.  ls\_return-type = 'E'.  ls\_return-id = 'ZFI\_NB'.  ls\_return-number = 075.  ls\_return-message = 'Dokument konnte nicht in internes Archiv übernommen werden'.  endif.  \* -----------------------------------------------------  \* Symbol IXOS-dokument aktivieren Zeile.  \* -----------------------------------------------------  else.  clear ls\_return.  ls\_return-type = 'E'.  ls\_return-id = 'ZFI\_NB'.  ls\_return-number = 076.  ls\_return-message = 'Dokument ist kein PDF-Dokument'.  append ls\_return to lt\_return.  endif.  \*------------------------------------------------------  \* Export Parameter  \*------------------------------------------------------  et\_return = lt\_return.  endmethod.  method get\_forderungen\_detail.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* es werden die Details des Forderungsbeleges geholt  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lt\_detail type zfi\_t\_nb\_forderungen\_details,  ls\_detail type zfi\_s\_nb\_forderungen\_details.  constants: co\_bukrs\_0330 type bukrs value '0330',  co\_spras\_de type spras value 'D',  co\_ktopl\_ethz type ktopl value 'ETHZ'.  field-symbols: <ls\_detail> type zfi\_s\_nb\_forderungen\_details.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  \*-----------------------------------------------------------------------  \* Tabellen übernehmen --------------------------------------------------  \*-----------------------------------------------------------------------  if iv\_belnr IS NOT INITIAL and iv\_gjahr IS NOT INITIAL and iv\_buzei IS NOT INITIAL.  select \* from bseg into corresponding fields of table lt\_detail  where bukrs = co\_bukrs\_0330  and belnr = iv\_belnr  and gjahr = iv\_gjahr  and buzei = iv\_buzei  and koart = 'S'.  elseif iv\_belnr IS NOT INITIAL and iv\_gjahr IS NOT INITIAL.  select \* from bseg into corresponding fields of table lt\_detail  where bukrs = co\_bukrs\_0330  and belnr = iv\_belnr  and gjahr = iv\_gjahr  and koart = 'S'.  else.  select \* from bseg into corresponding fields of table lt\_detail UP TO 10 ROWS  where bukrs = co\_bukrs\_0330  and koart = 'S'.  endif.  sort lt\_detail.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  \* ergänzen mit Bezeichnungen.  loop at lt\_detail assigning <ls\_detail>.  \* Sachkontenbezeichnung  select txt50 from skat into <ls\_detail>-hkont\_txt  where spras = co\_spras\_de  and ktopl = co\_ktopl\_ethz  and saknr = <ls\_detail>-hkont.  endselect.  \* Bezeichnung PSP  select posid post1 from prps into ( <ls\_detail>-posid , <ls\_detail>-projk\_txt )  where pspnr = <ls\_detail>-projk.  endselect.  endloop.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  et\_detail[] = lt\_detail[].  endmethod.  method get\_vbund\_from\_bp.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Es wird das Feld VBUND (IC-Partner aufgrunc BP geholt  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*-------------------------------------------------------------  \* Datendefinitonen  \*-------------------------------------------------------------  data: lv\_bp type bu\_partner.  \*-------------------------------------------------------------  \* Import-Parameter übernehmen  \*-------------------------------------------------------------  lv\_bp = iv\_bp.  \*-------------------------------------------------------------  \* Verarbeitung  \*-------------------------------------------------------------  clear ev\_vbund.  select vbund from bp001  where partner = @iv\_bp  into @ev\_vbund.  endselect.  \*-------------------------------------------------------------  \* Export-Parameter  \*-------------------------------------------------------------  endmethod.  ENDCLASS. |

|  |
| --- |
| @ObjectModel.query.implementedBy: 'ABAP:ZCL\_GET\_DETAIL\_POSITION'  @EndUserText.label: 'Gets Dunn Position Details'  @Search.searchable: false  @UI: {  headerInfo: {  typeName: 'Überfällige Forderungen Positionen',  typeNamePlural: 'Überfällige Forderungen Positionen',  title: {  type: #STANDARD,  label: 'Überfällige Forderungen Positionen',  value: 'BELNR'  }  }  }  **define** **custom** **entity** ZFI\_I\_POSITION\_DETAILS\_2  **{**  **key** belnr **:** *belnr\_d***;**  **key** gjahr **:** *gjahr***;**  **key** buzei **:** *buzei***;**  **key** bukrs **:** *bukrs***;**  @UI.lineItem : [{position: 40, cssDefault.width: '10rem'}]  hkont **:** *z\_fi\_hkont***;**  @UI.lineItem : [{position: 50, cssDefault.width: '12rem'}]  hkont\_txt **:** *z\_fi\_hkont\_txt***;**  @UI.lineItem : [{position: 10, cssDefault.width: '12rem'}]  prctr **:** *z\_fi\_prctr***;**  @EndUserText.label: 'Projekt'  projk **:** *char8***;**  @UI.lineItem : [{position: 20, cssDefault.width: '12rem', label: 'Kontierung'}]  posid **:** *z\_fi\_posid***;**  @UI.lineItem : [{position: 30, cssDefault.width: '21rem'}]  projk\_txt **:** *z\_fi\_ps\_post1***;**  @Semantics.amount.currencyCode: 'H\_WAERS'  @UI.lineItem : [{position: 60, cssDefault.width: '16rem'}]  wrbtr **:** *z\_fi\_wrbtr***;**  @Semantics.currencyCode: true  h\_waers **:** *waers***;**  \_Header **:** **association** **to** **parent** ZI\_DUNN\_ABAP\_CALL\_2  **on** **$projection.**belnr **=** \_Header**.**belnr  **and** **$projection.**gjahr **=** \_Header**.**gjahr  **and** **$projection.**bukrs **=** \_Header**.**bukrs  **and** **$projection.**buzei **=** \_Header**.**BUZEI**;**  **}** |

Class: ZCL\_GET\_DETAIL\_POSITION

|  |
| --- |
| CLASS zcl\_get\_detail\_position DEFINITION  PUBLIC  FINAL  CREATE PUBLIC .  PUBLIC SECTION.  INTERFACES if\_rap\_query\_provider .  PROTECTED SECTION.  PRIVATE SECTION.  ENDCLASS.  CLASS ZCL\_GET\_DETAIL\_POSITION IMPLEMENTATION.  METHOD if\_rap\_query\_provider~select.  FIELD-SYMBOLS: <fs\_tab> TYPE STANDARD TABLE.  DATA: lt\_result TYPE zfi\_t\_nb\_forderungen\_details,  ls\_result LIKE LINE OF lt\_result, "xstring,  lv\_belnr TYPE belnr\_d,  lv\_gjahr TYPE gjahr,  lv\_buzei TYPE buzei,  lcl\_zficl\_root TYPE REF TO zficl\_root,  lv\_tabref TYPE REF TO data,  lv\_error TYPE flag.  MOVE sy-datum(4) TO lv\_gjahr.  TRY.  TRY.  TRY.  IF io\_request->is\_data\_requested( ).  io\_request->get\_paging( ).  DATA(lt\_filter\_cond) = io\_request->get\_parameters( ).  DATA(lt\_filter\_cond2) = io\_request->get\_filter( )->get\_as\_ranges( ).  IF lt\_filter\_cond IS NOT INITIAL.  lv\_belnr = VALUE #( lt\_filter\_cond[ parameter\_name = 'BELNO' ]-value OPTIONAL ). "Fetching the parameter value  lv\_gjahr = VALUE #( lt\_filter\_cond[ parameter\_name = 'GEJAHR' ]-value OPTIONAL ). "Fetching the parameter value  ELSEIF lt\_filter\_cond2 IS NOT INITIAL.  lv\_belnr = VALUE #( lt\_filter\_cond2[ name = 'BELNR' ]-range[ 1 ]-low OPTIONAL ). "Fetching the parameter value  lv\_gjahr = VALUE #( lt\_filter\_cond2[ name = 'GJAHR' ]-range[ 1 ]-low OPTIONAL ). "Fetching the parameter value  \* lv\_buzei = VALUE #( lt\_filter\_cond2[ name = 'BUZEI' ]-range[ 1 ]-low OPTIONAL ). "Fetching the parameter value  ENDIF.  CREATE OBJECT lcl\_zficl\_root.  IF lv\_error IS INITIAL.  TRY.  CALL METHOD zficl\_nb=>get\_forderungen\_detail(  EXPORTING  iv\_Belnr = lv\_belnr  iv\_gjahr = lv\_gjahr  \* iv\_buzei = lv\_buzei  IMPORTING  et\_detail = lt\_result ).  CATCH zcx\_bcgen.  \* do nothing  ENDTRY.  ENDIF.  ENDIF.  \* =====================================================================  \* IF lt\_result IS NOT INITIAL.  "APPEND wf\_pdf\_x TO lt\_result.  io\_response->set\_data( lt\_result ).  DATA(lv\_offset) = io\_request->get\_paging( )->get\_offset( ).  DATA(lv\_page\_size) = io\_request->get\_paging( )->get\_page\_size( ).  DATA(lv\_sort) = io\_request->get\_sort\_elements( ).  " io\_response->set\_total\_number\_of\_records( lines( lt\_result ) ).  "IF\_RAP\_QUERY\_RESPONSE~SET\_DATA->set\_data( lt\_result ).  " io\_response->set\_data( lt\_result ).  io\_response->set\_total\_number\_of\_records( lines( lt\_result ) ).  "io\_response->set\_total\_number\_of\_records( lo\_response->get\_count( ) ).  \* ENDIF.  CATCH cx\_rap\_query\_provider INTO DATA(lx\_exc).  ENDTRY.  CATCH cx\_rfc\_dest\_provider\_error INTO DATA(lx\_dest).  ENDTRY.  CATCH cx\_rap\_query\_filter\_no\_range.  ENDTRY.  ENDMETHOD.  ENDCLASS. |

|  |
| --- |
| @ObjectModel.query.implementedBy: 'ABAP:ZCL\_DUNN\_ACTION'  @EndUserText.label: 'Execute Actions'  @Search.searchable: false  **define** **custom** **entity** ZFI\_DUNN\_ACTION\_2  // with parameters  // bukrs : bukrs,  // belnr : belnr\_d,  // gjahr : gjahr,  // buzei : buzei,  // action : zfi\_e\_nb\_atyp,  // bemerkung : char50  **{**  **key** belnr **:** *belnr\_d***;**  **key** bukrs **:** *bukrs***;**  **key** gjahr **:** *gjahr***;**  **key** buzei **:** *buzei***;**  **key** subrc **:** *abap***.***char***(** 2 **);**  action **:** *zfi\_e\_nb\_atyp***;**  bemerkung **:** *char50***;**  \_Header **:** **association** **to** **parent** ZI\_DUNN\_ABAP\_CALL\_2  **on** **$projection.**belnr **=** \_Header**.**belnr  **and** **$projection.**gjahr **=** \_Header**.**gjahr  **and** **$projection.**buzei **=** \_Header**.**BUZEI  **and** **$projection.**bukrs **=** \_Header**.**bukrs**;**  **}** |

|  |
| --- |
| class ZCL\_DUNN\_ACTION definition  public  final  create public .  public section.  interfaces IF\_RAP\_QUERY\_PROVIDER .  PROTECTED SECTION.  PRIVATE SECTION.  ENDCLASS.  CLASS ZCL\_DUNN\_ACTION IMPLEMENTATION.  METHOD if\_rap\_query\_provider~select.  FIELD-SYMBOLS: <fs\_tab> TYPE STANDARD TABLE.  DATA: ls\_input\_struc TYPE ZFI\_S\_NB\_FORD\_INBOX\_ETHIS,  lv\_aktion TYPE ZFI\_E\_NB\_ATYP,  ls\_key TYPE ZFI\_S\_NB\_FORDERUNG\_KEY,  lv\_subrc TYPE subrc,  ls\_result TYPE ZFI\_S\_DUNN\_ACTION\_RETURN,  lt\_result like table of ls\_result,  lv\_bukrs TYPE bukrs,  lv\_gjahr TYPE gjahr,  lv\_belnr TYPE belnr\_d,  lv\_buzei TYPE buzei,  lv\_bemerkung TYPE char50.  try.  try.  if IO\_REQUEST->IS\_DATA\_REQUESTED( ). "Fetching incoming data  IO\_REQUEST->GET\_PAGING( ).  data(LT\_FILTER\_COND) = IO\_REQUEST->GET\_PARAMETERS( ). "Setting the filter condition, fetching parameter names from data definition  " DATA(lt\_filter\_cond2) = io\_request->get\_filter( )->get\_as\_ranges( ).  lv\_bukrs = value #( LT\_FILTER\_COND[ PARAMETER\_NAME = 'BUKRS' ]-VALUE optional ).  lv\_gjahr = value #( LT\_FILTER\_COND[ PARAMETER\_NAME = 'GJAHR' ]-VALUE optional ).  lv\_belnr = value #( LT\_FILTER\_COND[ PARAMETER\_NAME = 'BELNR' ]-VALUE optional ).  lv\_buzei = value #( LT\_FILTER\_COND[ PARAMETER\_NAME = 'BUZEI' ]-VALUE optional ).  lv\_aktion = value #( LT\_FILTER\_COND[ PARAMETER\_NAME = 'ACTION' ]-VALUE optional ).  lv\_bemerkung = value #( LT\_FILTER\_COND[ PARAMETER\_NAME = 'BEMERKUNG' ]-VALUE optional ).  endif.  clear ls\_key.  ls\_key-bukrs = lv\_bukrs. "ls\_input\_struc-bukrs.  ls\_key-belnr = lv\_belnr. "ls\_input\_struc-belnr.  ls\_key-gjahr = lv\_gjahr. "ls\_input\_struc-gjahr.  ls\_key-buzei = lv\_buzei. "ls\_input\_struc-buzei.  call method zficl\_nb=>create\_forderung\_aktion  exporting  is\_forderung\_key = ls\_key  iv\_aenderer = sy-uname  iv\_aktion = lv\_aktion  iv\_bemerkung = lv\_bemerkung  iv\_datum = sy-datum "added 27102022 MEN  importing  ev\_subrc = lv\_subrc.  " MOVE-CORRESPONDING lt\_forderungen to et\_entityset.  MOVE lv\_subrc TO ls\_result-subrc.  APPEND ls\_result TO lt\_result.  " io\_response->set\_data( lt\_result ).  io\_response->set\_data( lt\_result ).  DATA(lv\_offset) = io\_request->get\_paging( )->get\_offset( ).  DATA(lv\_page\_size) = io\_request->get\_paging( )->get\_page\_size( ).  " io\_response->set\_total\_number\_of\_records( lines( lt\_result ) ).  "IF\_RAP\_QUERY\_RESPONSE~SET\_DATA->set\_data( lt\_result ).  " io\_response->set\_data( lt\_result ).  io\_response->set\_total\_number\_of\_records( lines( lt\_result ) ).  "io\_response->set\_total\_number\_of\_records( lo\_response->get\_count( ) ).  " ENDIF.  CATCH cx\_rap\_query\_provider INTO DATA(lx\_exc).  ENDTRY.  CATCH cx\_rfc\_dest\_provider\_error INTO DATA(lx\_dest).  ENDTRY.  " CATCH cx\_rap\_query\_filter\_no\_range.  " ENDTRY.  ENDMETHOD.  ENDCLASS. |

1. Behavior

|  |
| --- |
| **unmanaged** **implementation** **in** **class** zbp\_i\_dunn\_abap\_call\_2 **unique;**  **strict;** //Comment this line in to enable strict mode. The strict mode is prerequisite to be future proof regarding syntax and to be able to release your BO.  **define** **behavior** **for** ZI\_DUNN\_ABAP\_CALL\_2 //alias <alias\_name>  //late numbering  **lock** **master**  **authorization** **master** **(** **instance** **)**  //etag master <field\_name>  **{**  // create;  // update;  // delete;  **association** \_Actions**;**  **association** \_Positions**;**  **action** inkasso **result[1]** **$self;**  **action** mahnen **result[1]** **$self;**  **action** verlaengern **result[1]** **$self;**  **action** ausbuchen **parameter** ZFI\_DUNN\_AUSBUCHEN **result[1]** **$self;**  **}**  **define** **behavior** **for** ZFI\_DUNN\_ACTION\_2 //alias <alias\_name>  //late numbering  **lock** **dependent** **by** \_Header  **authorization** **dependent** **by** \_Header  //etag master <field\_name>  **{**  // update;  // delete;  **field** **(** **readonly** **)** belnr**,** gjahr**,** buzei**,** bukrs**;**  **association** \_Header**;**  **}**  **define** **behavior** **for** ZFI\_I\_POSITION\_DETAILS\_2 //alias <alias\_name>  //late numbering  **lock** **dependent** **by** \_Header  **authorization** **dependent** **by** \_Header  //etag master <field\_name>  **{**  // update;  // delete;  **field** **(** **readonly** **)** belnr**,** gjahr**,** bukrs**,** buzei**;**  **association** \_Header**;**  **}** |
|  |

1. Behavior Class

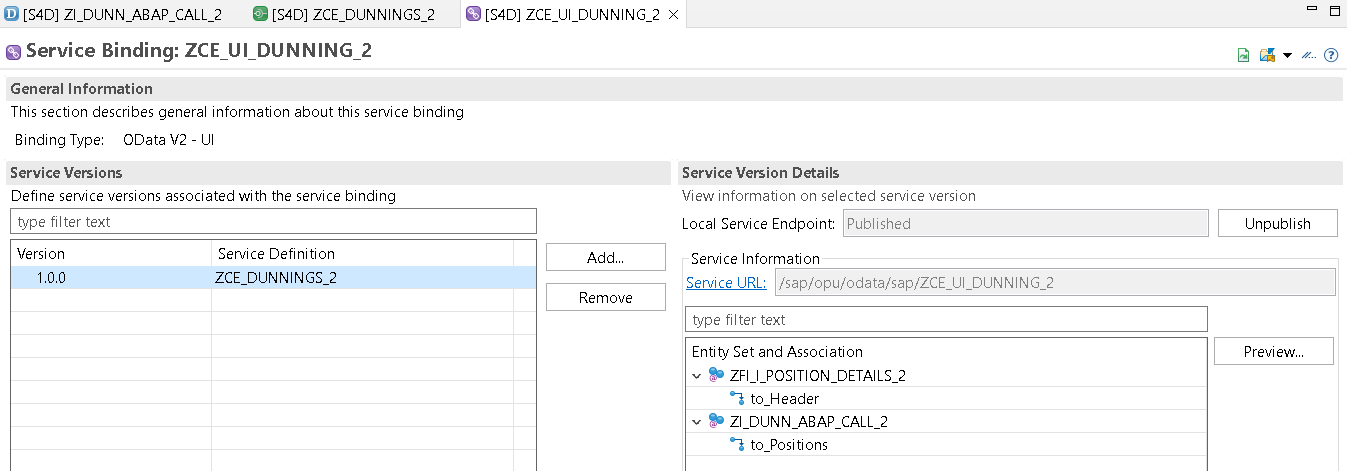
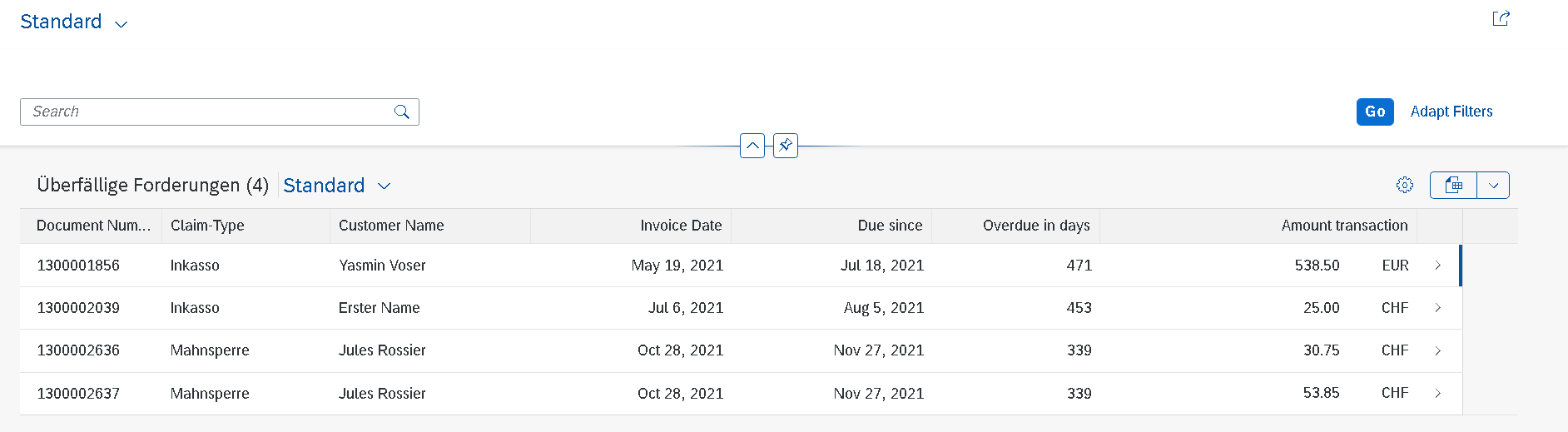
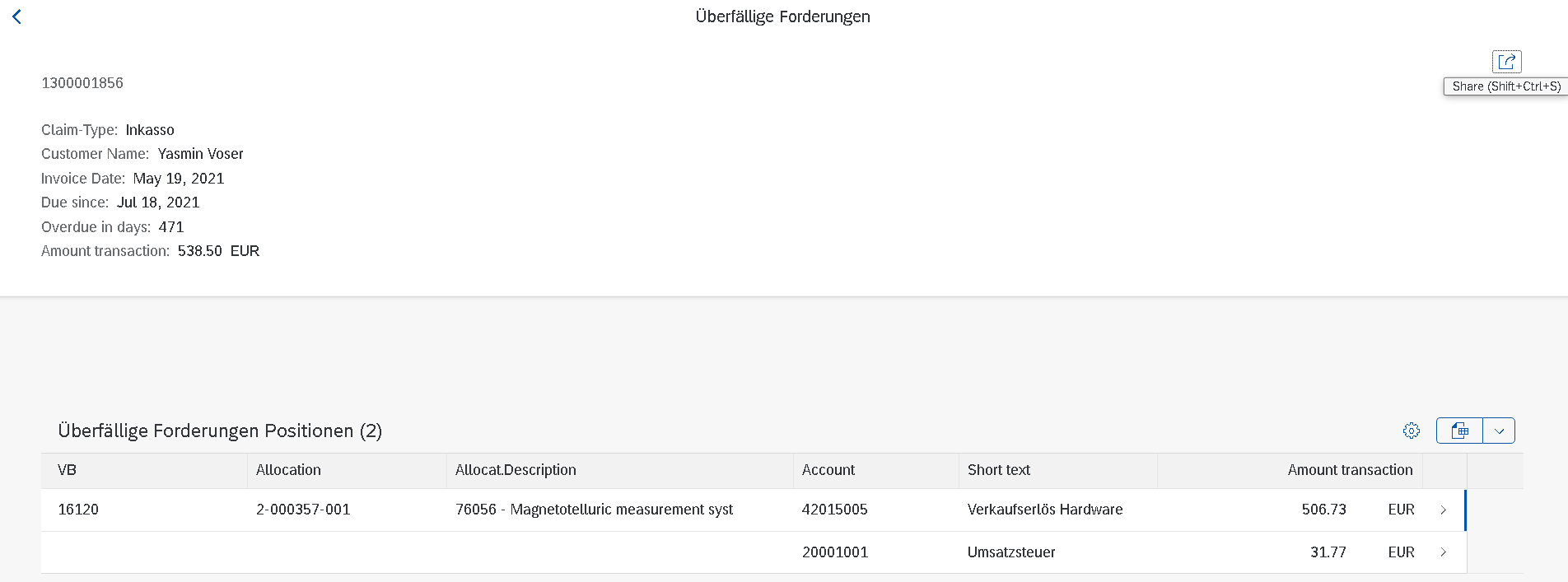
|  |
| --- |
| CLASS lhc\_ZI\_DUNN\_ABAP\_CALL\_2 DEFINITION INHERITING FROM cl\_abap\_behavior\_handler.  PRIVATE SECTION.  METHODS get\_instance\_authorizations FOR INSTANCE AUTHORIZATION  IMPORTING keys REQUEST requested\_authorizations FOR zi\_dunn\_abap\_call\_2 RESULT result.  METHODS read FOR READ  IMPORTING keys FOR READ zi\_dunn\_abap\_call\_2 RESULT result.  METHODS lock FOR LOCK  IMPORTING keys FOR LOCK zi\_dunn\_abap\_call\_2.  METHODS rba\_Actions FOR READ  IMPORTING keys\_rba FOR READ zi\_dunn\_abap\_call\_2\\_Actions FULL result\_requested RESULT result LINK association\_links.  METHODS rba\_Positions FOR READ  IMPORTING keys\_rba FOR READ zi\_dunn\_abap\_call\_2\\_Positions FULL result\_requested RESULT result LINK association\_links.  METHODS ausbuchen FOR MODIFY  IMPORTING keys FOR ACTION zi\_dunn\_abap\_call\_2~ausbuchen RESULT result.  METHODS inkasso FOR MODIFY  IMPORTING keys FOR ACTION zi\_dunn\_abap\_call\_2~inkasso RESULT result.  METHODS mahnen FOR MODIFY  IMPORTING keys FOR ACTION zi\_dunn\_abap\_call\_2~mahnen RESULT result.  METHODS verlaengern FOR MODIFY  IMPORTING keys FOR ACTION zi\_dunn\_abap\_call\_2~verlaengern RESULT result.  ENDCLASS.  CLASS lhc\_ZI\_DUNN\_ABAP\_CALL\_2 IMPLEMENTATION.  METHOD get\_instance\_authorizations.  ENDMETHOD.  METHOD read.  ENDMETHOD.  METHOD lock.  ENDMETHOD.  METHOD rba\_Actions.  ENDMETHOD.  METHOD rba\_Positions.  ENDMETHOD.  METHOD ausbuchen.  READ TABLE keys ASSIGNING FIELD-SYMBOL(<fs\_keys>) INDEX 1.  CHECK sy-subrc = 0.  CALL METHOD zficl\_nb=>create\_forderung\_aktion  EXPORTING  is\_forderung\_key = VALUE #( belnr = <fs\_keys>-belnr bukrs = <fs\_keys>-bukrs buzei = <fs\_keys>-buzei gjahr = <fs\_keys>-gjahr )  iv\_aenderer = sy-uname  iv\_aktion = 'A'  iv\_bemerkung = <fs\_keys>-%param-bemerkung  is\_rap\_behavior = abap\_true  IMPORTING  ev\_subrc = DATA(lv\_subrc).  IF lv\_subrc IS INITIAL.  result = CORRESPONDING #( keys ).  ENDIF.  ENDMETHOD.  METHOD inkasso.  READ TABLE keys ASSIGNING FIELD-SYMBOL(<fs\_keys>) INDEX 1.  CHECK sy-subrc = 0.  CALL METHOD zficl\_nb=>create\_forderung\_aktion  EXPORTING  is\_forderung\_key = VALUE #( belnr = <fs\_keys>-belnr bukrs = <fs\_keys>-bukrs buzei = <fs\_keys>-buzei gjahr = <fs\_keys>-gjahr )  iv\_aenderer = sy-uname  iv\_aktion = 'B'  iv\_bemerkung = space  is\_rap\_behavior = abap\_true  IMPORTING  ev\_subrc = DATA(lv\_subrc).  IF lv\_subrc IS INITIAL.  result = CORRESPONDING #( keys ).  ENDIF.  ENDMETHOD.  METHOD mahnen.  READ TABLE keys ASSIGNING FIELD-SYMBOL(<fs\_keys>) INDEX 1.  CHECK sy-subrc = 0.  CALL METHOD zficl\_nb=>create\_forderung\_aktion  EXPORTING  is\_forderung\_key = VALUE #( belnr = <fs\_keys>-belnr bukrs = <fs\_keys>-bukrs buzei = <fs\_keys>-buzei gjahr = <fs\_keys>-gjahr )  iv\_aenderer = sy-uname  iv\_aktion = 'M'  iv\_bemerkung = space  is\_rap\_behavior = abap\_true  IMPORTING  ev\_subrc = DATA(lv\_subrc).  IF lv\_subrc IS INITIAL.  result = CORRESPONDING #( keys ).  ENDIF.  ENDMETHOD.  METHOD verlaengern.  READ TABLE keys ASSIGNING FIELD-SYMBOL(<fs\_keys>) INDEX 1.  CHECK sy-subrc = 0.  CALL METHOD zficl\_nb=>create\_forderung\_aktion  EXPORTING  is\_forderung\_key = VALUE #( belnr = <fs\_keys>-belnr bukrs = <fs\_keys>-bukrs buzei = <fs\_keys>-buzei gjahr = <fs\_keys>-gjahr )  iv\_aenderer = sy-uname  iv\_aktion = 'V'  iv\_bemerkung = space  iv\_datum = sy-datum  is\_rap\_behavior = abap\_true  IMPORTING  ev\_subrc = DATA(lv\_subrc).  IF lv\_subrc IS INITIAL.  result = CORRESPONDING #( keys ).  ENDIF.  ENDMETHOD.  ENDCLASS.  CLASS lhc\_ZFI\_DUNN\_ACTION\_2 DEFINITION INHERITING FROM cl\_abap\_behavior\_handler.  PRIVATE SECTION.  METHODS read FOR READ  IMPORTING keys FOR READ zfi\_dunn\_action\_2 RESULT result.  METHODS rba\_Header FOR READ  IMPORTING keys\_rba FOR READ zfi\_dunn\_action\_2\\_Header FULL result\_requested RESULT result LINK association\_links.  ENDCLASS.  CLASS lhc\_ZFI\_DUNN\_ACTION\_2 IMPLEMENTATION.  METHOD read.  ENDMETHOD.  METHOD rba\_Header.  ENDMETHOD.  ENDCLASS.  CLASS lhc\_ZFI\_I\_POSITION\_DETAILS\_2 DEFINITION INHERITING FROM cl\_abap\_behavior\_handler.  PRIVATE SECTION.  METHODS read FOR READ  IMPORTING keys FOR READ zfi\_i\_position\_details\_2 RESULT result.  METHODS rba\_Header FOR READ  IMPORTING keys\_rba FOR READ zfi\_i\_position\_details\_2\\_Header FULL result\_requested RESULT result LINK association\_links.  ENDCLASS.  CLASS lhc\_ZFI\_I\_POSITION\_DETAILS\_2 IMPLEMENTATION.  METHOD read.  ENDMETHOD.  METHOD rba\_Header.  ENDMETHOD.  ENDCLASS.  CLASS lsc\_ZI\_DUNN\_ABAP\_CALL\_2 DEFINITION INHERITING FROM cl\_abap\_behavior\_saver.  PROTECTED SECTION.  METHODS finalize REDEFINITION.  METHODS check\_before\_save REDEFINITION.  METHODS save REDEFINITION.  METHODS cleanup REDEFINITION.  METHODS cleanup\_finalize REDEFINITION.  ENDCLASS.  CLASS lsc\_ZI\_DUNN\_ABAP\_CALL\_2 IMPLEMENTATION.  METHOD finalize.  ENDMETHOD.  METHOD check\_before\_save.  ENDMETHOD.  METHOD save.  ENDMETHOD.  METHOD cleanup.  ENDMETHOD.  METHOD cleanup\_finalize.  ENDMETHOD.  ENDCLASS. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

1. Service Definition

|  |
| --- |
| @EndUserText.label: 'Service interface for Dunning'  **define** **service** ZCE\_DUNNINGS\_2 **{**  **expose** ZI\_DUNN\_ABAP\_CALL\_2**;**  **expose** ZFI\_I\_POSITION\_DETAILS\_2**;**  **}** |

1. Service Binding  
     
   
2. Preview  
     
   
3. App – Check the video  
     
   