class ZCL\_ESL\_MAKE\_XML definition  
 public  
 final  
 create public .   
public section.  
 type-pools ZGESL .   
 methods END\_LOG .  
 methods OUT\_DATA  
 exporting  
 !ET\_ROWS type ZESLSI\_HR\_PERSONNEL\_EMPLO\_TAB1  
 !ET\_HEAD type ZESLTT\_OUT\_HEAD .  
 methods CONSTRUCTOR  
 importing  
 !IV\_PROFILE type ZESLD\_XML\_PROFIL .  
 methods MAKE\_PARSE\_REESTR\_REQUEST  
 importing  
 !IS\_DATA type ZESLS\_XML\_PARSE\_REESTR  
 !IV\_HASHALG type SSFHASH .  
 methods SEND\_REQUEST  
 importing  
 !IV\_SNILS type ZESLD\_SNILS  
 !IV\_LN\_CODE type ZESLD\_LN\_CODE  
 !IV\_EMPL\_REG\_NO type P33\_FSSREGNO  
 !IV\_ENCRYPT type FLAG optional  
 !IV\_OGRN type P33\_KOGRN  
 !IS\_PARSE\_REESTR type ZESLS\_XML\_PARSE\_REESTR optional  
 !IV\_PARSE\_FLAG type FLAG optional  
 !IV\_HASHALG type SSFHASH optional .  
 class-methods FUNCTION\_SHA1  
 importing  
 !IV\_INPUT type STRING  
 !IV\_HEX type ABAP\_BOOL  
 returning  
 value(RV\_HASH) type STRING .  
protected section.  
private section.  
  
 data GOR\_LOG type ref to ZCL\_UNI\_SBAL .  
 data GT\_OUT\_ROWS type ZESLSI\_HR\_PERSONNEL\_EMPLO\_TAB1 .  
 data GT\_HEAD type ZESLTT\_OUT\_HEAD .  
 data GV\_PROFIL type ZESLD\_XML\_PROFIL .  
 data GV\_PATH type STRING .  
  
 methods DELETE\_XML\_CAPTION  
 changing  
 !CV\_XML type STRING .  
 methods EXPORT\_DATA  
 importing  
 !IS\_HEAD type ZESLS\_OUT\_HEAD .  
 methods GET\_ENCRYPTED\_DATA  
 importing  
 !IV\_XML type STRING  
 exporting  
 !EV\_ENCR\_DATA type STRING .  
 methods HANDLE\_ERRORS  
 importing  
 !IO\_ERR type ref to CX\_AI\_SYSTEM\_FAULT .  
 methods CONVERT\_DECRYPTED\_XML\_TOFIELDS  
 importing  
 !IV\_XML type STRING  
 exporting  
 !ES\_DECR\_FIELDS type ZESLS\_OUT\_HEAD .  
 methods CHANGE\_TAGS\_ENCRYPTED\_DATA  
 changing  
 !CS\_ENCR\_DATA type STRING .  
 methods PI\_TRANSFER  
 importing  
 !IS\_INPUT type ZESLENCRYPTED\_DATA\_HR  
 exporting  
 !ES\_OUTPUT type ZESLENCRYPTED\_DATA\_HR .  
 methods READ\_ENCRYPTED\_RESPONSE  
 importing  
 !IS\_DECR\_DATA type ZESLS\_OUT\_HEAD .  
 methods READ\_SIGNED\_RESPONSE  
 importing  
 !IS\_OUTPUT type ZESLMESSAGE\_ENVELOPE\_RESPONSE .  
 methods CHANGE\_TAGS\_DECRYPTED\_DATA  
 changing  
 !CS\_ENCR\_DATA type STRING .  
 methods NO\_ENCRYPT  
 importing  
 !IS\_INPUT type ZESLMESSAGE\_ENVELOPE\_REQUEST  
 !IV\_HASHALG type SSFHASH .  
 methods MAKE\_REQUEST1  
 importing  
 !IV\_SNILS type ZESLD\_SNILS  
 !IV\_LN\_CODE type ZESLD\_LN\_CODE  
 !IV\_EMPL\_REG\_NO type P33\_FSSREGNO  
 returning  
 value(RV\_DOC) type STRING .  
 methods MAKE\_REQUEST3  
 importing  
 !IS\_DATA type ZESLS\_XML\_PARSE\_REESTR  
 returning  
 value(RV\_DOC) type STRING .  
 methods MAKE\_REQUEST3\_FSS\_EXAMPLE  
 returning  
 value(RV\_DOC) type STRING .  
 methods MAKE\_REQUEST4  
 importing  
 !IV\_SNILS type ZESLD\_SNILS  
 !IV\_LN\_CODE type ZESLD\_LN\_CODE  
 !IV\_EMPL\_REG\_NO type P33\_FSSREGNO  
 !IV\_DIGEST type STRING  
 !IV\_SIGN type STRING  
 returning  
 value(RV\_DOC) type STRING .  
 methods SI\_ENCRYPT  
 importing  
 !IV\_SIGNED\_DATA type STRING  
 exceptions  
 ERROR\_ENCR .  
 methods MAKE\_REQUEST2  
 importing  
 !IV\_DIGEST type STRING  
 !IV\_EMPL\_REG\_NO type P33\_FSSREGNO  
 !IV\_LNCODE type ZESLD\_LN\_CODE  
 returning  
 value(RV\_DOC) type STRING .  
 class-methods GET\_PARAM  
 importing  
 !IV\_PARAM type ZESLD\_PARAMNAME  
 !IV\_PROFIL type ZESLD\_XML\_PROFIL optional  
 returning  
 value(RV\_VALUE) type ZESLD\_PARAMVALUE .  
ENDCLASS.  
  
CLASS ZCL\_ESL\_MAKE\_XML IMPLEMENTATION.  
*SIGNATURE>*  
 METHOD change\_tags\_decrypted\_data.  
  
 REPLACE ALL OCCURRENCES OF '&lt;' IN cs\_encr\_data WITH '<'.  
 REPLACE ALL OCCURRENCES OF '&gt;' IN cs\_encr\_data WITH '>'.  
  
 ENDMETHOD.  
*SIGNATURE>*  
 METHOD change\_tags\_encrypted\_data.  
  
 REPLACE ALL OCCURRENCES OF 'EncryptedData' IN cs\_encr\_data WITH 'xenc:EncryptedData'.  
 REPLACE ALL OCCURRENCES OF 'EncryptionMethod' IN cs\_encr\_data WITH 'xenc:EncryptionMethod'.  
 REPLACE ALL OCCURRENCES OF 'KeyInfo' IN cs\_encr\_data WITH 'ds:KeyInfo'.  
 REPLACE ALL OCCURRENCES OF 'EncryptedKey' IN cs\_encr\_data WITH 'xenc:EncryptedKey'.  
 REPLACE ALL OCCURRENCES OF 'X509Data' IN cs\_encr\_data WITH 'ds:X509Data'.  
 REPLACE ALL OCCURRENCES OF 'X509Certificate' IN cs\_encr\_data WITH 'ds:X509Certificate'.  
 REPLACE ALL OCCURRENCES OF 'CipherData' IN cs\_encr\_data WITH 'xenc:CipherData'.  
 REPLACE ALL OCCURRENCES OF 'CipherValue' IN cs\_encr\_data WITH 'xenc:CipherValue'.  
  
 REPLACE '<xenc:EncryptedData' IN cs\_encr\_data  
 WITH '<xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"'.  
  
 REPLACE '<ds:KeyInfo xmlns' IN cs\_encr\_data WITH '<ds:KeyInfo xmlns:ds'.  
  
 cs\_encr\_data = '<EncryptedDataHR>' && cs\_encr\_data && '</EncryptedDataHR>'.  
  
 ENDMETHOD. *SIGNATURE>*  
 METHOD constructor.  
  
 CREATE OBJECT gor\_log  
 EXPORTING  
 im\_object = zgesl\_bal\_object  
 im\_subobject = zgesl\_bal\_subobj.  
  
 gv\_profil = iv\_profile.  
  
 CALL METHOD zcl\_custom\_setting=>get\_param  
 EXPORTING  
 i\_name = zgesl\_tvarv-down\_path  
 IMPORTING  
 e\_param = gv\_path  
 EXCEPTIONS  
 no\_data = 1  
 conversion\_error = 2  
 type\_mismatch = 3  
 OTHERS = 4.  
 IF sy-subrc <> 0.  
 MESSAGE e001(zesl\_sick\_list) WITH zgesl\_tvarv-down\_path.  
 ENDIF.   
 ENDMETHOD. *SIGNATURE>*  
 METHOD convert\_decrypted\_xml\_tofields.  
  
 DATA: ls\_rif\_ex TYPE REF TO cx\_root,  
 ls\_var\_text TYPE string.  
  
 DATA: ls\_result\_xml TYPE abap\_trans\_resbind.  
  
 CLEAR es\_decr\_fields.  
  
*\*Получаем отдельные поля*  
 GET REFERENCE OF es\_decr\_fields INTO ls\_result\_xml-value.  
 ls\_result\_xml-name = 'OUT\_DATA'.  
  
 TRY.  
 CALL TRANSFORMATION zesl\_convert\_decrypted\_xml  
 SOURCE XML iv\_xml  
 RESULT out\_data = es\_decr\_fields.  
  
 CATCH cx\_root INTO ls\_rif\_ex.  
 ls\_var\_text = ls\_rif\_ex->get\_text( ).  
 MESSAGE e000(zesl\_h2h) WITH ls\_var\_text.  
 ENDTRY.  
  
*\*Получаем весь XML*  
 es\_decr\_fields-xml = iv\_xml.  
  
 ENDMETHOD.

Листинг Б.2 – Исходный код класса ZCL\_ESL\_XML\_PROVIDER

class ZCL\_ESL\_XML\_PROVIDER definition  
 public  
 final  
 create public .  
  
public section.  
  
 methods CONSTRUCTOR .  
*\*"\* public components of class ZCL\_ESL\_XML\_PROVIDER*  
*\*"\* do not include other source files here!!!*  
 class-methods SET\_DATA\_REQUEST  
 importing  
 !IT\_GETLNDATA type ZESLT\_XML\_GET\_LNDATA optional  
 !IS\_GETLNDATA type ZESLS\_XML\_GET\_LNDATA optional  
 !IT\_PARSE\_REESTR type ZESLTT\_XML\_PARSE\_REESTR optional  
 !IS\_PARSE\_REESTR type ZESLS\_XML\_PARSE\_REESTR optional .  
 class-methods EXPORT\_PROVIDER\_DATA  
 exporting  
 !ET\_GETLNDATA type ZESLT\_XML\_GET\_LNDATA  
 !ES\_GETLNDATA type ZESLS\_XML\_GET\_LNDATA  
 !ET\_PARSE\_REESTR type ZESLTT\_XML\_PARSE\_REESTR  
 !ES\_PARSE\_REESTR type ZESLS\_XML\_PARSE\_REESTR .  
 class-methods GENERATE\_REQUEST\_XML  
 importing  
 !IV\_HIER\_CODE type J\_3RF\_REP\_HIER\_KEY  
 !IT\_GETLNDATA type ZESLT\_XML\_GET\_LNDATA optional  
 !IS\_GETLNDATA type ZESLS\_XML\_GET\_LNDATA optional  
 !IT\_PARSE\_REESTR type ZESLTT\_XML\_PARSE\_REESTR optional  
 !IS\_PARSE\_REESTR type ZESLS\_XML\_PARSE\_REESTR optional  
 exporting  
 !ET\_XML\_OUT type J3RXR\_REP\_XML\_TAB .  
protected section.  
*\*"\* protected components of class ZCL\_ESL\_XML\_PROVIDER*  
*\*"\* do not include other source files here!!!*  
private section.  
  
 class-data GT\_GETLNDATA type ZESLT\_XML\_GET\_LNDATA .  
 class-data GS\_GETLNDATA type ZESLS\_XML\_GET\_LNDATA .  
 class-data GT\_PARSE\_REESTR type ZESLTT\_XML\_PARSE\_REESTR .  
 class-data GS\_PARSE\_REESTR type ZESLS\_XML\_PARSE\_REESTR .  
  
 class-methods GENERATE\_FILE\_NAME  
 exporting  
 !EV\_FILE\_ID type CHAR255 .  
ENDCLASS.  
  
  
  
CLASS ZCL\_ESL\_XML\_PROVIDER IMPLEMENTATION.

*SIGNATURE>*  
 method CONSTRUCTOR.  
 endmethod.

*SIGNATURE>*  
METHOD export\_provider\_data.  
  
 es\_getlndata = gs\_getlndata.  
  
 CLEAR et\_getlndata.  
 APPEND gs\_getlndata TO et\_getlndata.  
  
 es\_parse\_reestr = gs\_parse\_reestr.  
  
 CLEAR et\_parse\_reestr.  
 APPEND gs\_parse\_reestr TO et\_parse\_reestr.  
  
ENDMETHOD.  
*SIGNATURE>*  
method GENERATE\_FILE\_NAME.  
endmethod.  
*SIGNATURE>*  
METHOD GENERATE\_REQUEST\_XML.  
  
*\*... §1 Локальные определения*  
*\* type-pools: j3rxr.*  
  
 DATA:  
 lt\_subst TYPE j3rxr\_role\_subst\_tab,  
 lv\_xml\_name type string,  
 ls\_subst TYPE j3rxr\_role\_line,  
 lr\_renderer TYPE REF TO if\_j3rf\_rep\_xml\_render.  
  
 CHECK sy-batch EQ abap\_false.  
*\*... §2 XML-predata Request/PayDocRu*  
 set\_data\_request( EXPORTING  
 it\_getlndata = it\_getlndata  
 is\_getlndata = is\_getlndata  
 it\_parse\_reestr = it\_parse\_reestr  
 is\_parse\_reestr = is\_parse\_reestr  
 ).  
*\* gv\_file\_name = lv\_xml\_name.*  
  
  
*\*... §3 Потоковая передача*  
 CREATE OBJECT lr\_renderer TYPE j\_3rf\_rep\_stream\_contswitch.  
  
 CLEAR: ls\_subst, lt\_subst.  
*\* ls\_subst-role\_name = 'XMLFILENAME'.*  
*\* ls\_subst-role\_value = lv\_xml\_name.*  
*\* INSERT ls\_subst INTO TABLE lt\_subst.*  
*\**  
*\* ls\_subst-role\_name = 'XMLPATH'.*  
*\* ls\_subst-role\_value = 'D:' . "iv\_file\_path.*  
*\* INSERT ls\_subst INTO TABLE lt\_subst.*  
  
 ls\_subst-role\_name = 'FLUSH\_LIMIT'.  
 ls\_subst-role\_value = '524288'. *" 512 Kb chunks*  
 INSERT ls\_subst INTO TABLE lt\_subst.  
  
 j\_3rf\_report=>generate\_xml( EXPORTING  
 in\_hier\_key = iv\_hier\_code  
 in\_default\_fname = lv\_xml\_name  
*\* in\_supress\_selscreen = abap\_true*  
 in\_supress\_frontend\_save = abap\_true  
 in\_subst\_table = lt\_subst  
 in\_rend\_obj = lr\_renderer *" Subst. generation engine*  
 IMPORTING  
*\* out\_hier = " Instance of the actual exporting hierarchy*  
 out\_xmltab = et\_xml\_out *" Table with generated XML files*  
 ).  
  
ENDMETHOD.   
METHOD set\_data\_request.  
  
 gt\_getlndata = it\_getlndata.  
 gs\_getlndata = is\_getlndata.  
  
 gt\_parse\_reestr = it\_parse\_reestr.  
 gs\_parse\_reestr = is\_parse\_reestr.  
  
ENDMETHOD.  
ENDCLASS.

Листинг Б.3 – Исходный код класса ZCL\_ESL\_UTILS

Class ZCL\_ESL\_UTIL definition  
 public  
 final  
 create public .  
  
public section.  
  
 class-methods STR\_TO\_XSTR  
 importing  
 !IV\_STRING type STRING  
 returning  
 value(RV\_XSTRING) type XSTRING .  
 class-methods XSTR\_TO\_STR  
 importing  
 !IV\_XSTRING type XSTRING  
 returning  
 value(RV\_STRING) type STRING .  
 class-methods GET\_OGRN  
 importing  
 !IV\_PERNR type P\_PERNR  
 returning  
 value(RV\_OGRN) type P33\_KOGRN .  
 class-methods STR\_TO\_BASE64  
 importing  
 !IV\_STR type STRING  
 returning  
 value(RV\_STR\_BASE64) type STRING .  
 class-methods DOWNLOAD\_DATA  
 importing  
 !IV\_DATA type STRING  
 !IV\_FILE type STRING  
 !IV\_PREF type STRING .  
 class-methods BASE64\_TO\_STR  
 importing  
 !IV\_BASE64STR type STRING  
 returning  
 value(RV\_STR) type STRING .  
 class-methods XML\_TO\_TEXT  
 importing  
 !IT\_XML type J3RXR\_REP\_XML\_TAB  
 returning  
 value(RT\_RESPONSE) type SDOKCNTASCS .  
 class-methods BINTAB\_TO\_BASE64\_STR  
 importing  
 !IT\_BINTAB type RMPS\_SSFBIN  
 !IV\_LEN type INT4  
 returning  
 value(RE\_DATA) type STRING .  
protected section.  
private section.  
ENDCLASS.  
  
  
  
CLASS ZCL\_ESL\_UTIL IMPLEMENTATION.  
  
*SIGNATURE>*  
METHOD base64\_to\_str.  
  
 DATA:  
 lv\_data\_str TYPE string,  
 lt\_bintab TYPE rmps\_ssfbin,  
 lv\_data\_len TYPE i,  
 lv\_data TYPE xstring.  
  
 CALL FUNCTION 'SSFC\_BASE64\_DECODE'  
 EXPORTING  
 b64data = iv\_base64str  
 IMPORTING  
 bindata = lv\_data  
 EXCEPTIONS  
 ssf\_krn\_error = 1  
 ssf\_krn\_noop = 2  
 ssf\_krn\_nomemory = 3  
 ssf\_krn\_opinv = 4  
 ssf\_krn\_input\_data\_error = 5  
 ssf\_krn\_invalid\_par = 6  
 ssf\_krn\_invalid\_parlen = 7  
 OTHERS = 8.  
 IF sy-subrc <> 0.  
 MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
 WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
 ENDIF.  
  
 CALL FUNCTION 'SCMS\_XSTRING\_TO\_BINARY'  
 EXPORTING  
 buffer = lv\_data  
 IMPORTING  
 output\_length = lv\_data\_len  
 TABLES  
 binary\_tab = lt\_bintab.  
  
 CALL FUNCTION 'SCMS\_BINARY\_TO\_STRING'  
 EXPORTING  
 input\_length = lv\_data\_len  
 IMPORTING  
 text\_buffer = rv\_str  
 TABLES  
 binary\_tab = lt\_bintab  
 EXCEPTIONS  
 failed = 1  
 OTHERS = 2.  
 IF sy-subrc <> 0.  
 MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
 WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
 ENDIF.  
  
ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*SIGNATURE>*  
METHOD BINTAB\_TO\_BASE64\_STR.  
  
 DATA: lv\_xsign TYPE xstring.  
  
 CALL FUNCTION 'SCMS\_BINARY\_TO\_XSTRING'  
 EXPORTING  
 input\_length = iv\_len  
 IMPORTING  
 buffer = lv\_xsign  
 TABLES  
 binary\_tab = it\_bintab  
 EXCEPTIONS  
 failed = 1  
 OTHERS = 2.  
 IF sy-subrc <> 0.  
 MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
 WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
 ENDIF.  
  
*\*Кодируем в Base64 подпись*  
 CALL FUNCTION 'SSFC\_BASE64\_ENCODE'  
 EXPORTING  
 bindata = lv\_xsign  
 binleng = iv\_len  
 IMPORTING  
 b64data = re\_data  
 EXCEPTIONS  
 ssf\_krn\_error = 1  
 ssf\_krn\_noop = 2  
 ssf\_krn\_nomemory = 3  
 ssf\_krn\_opinv = 4  
 ssf\_krn\_input\_data\_error = 5  
 ssf\_krn\_invalid\_par = 6  
 ssf\_krn\_invalid\_parlen = 7  
 OTHERS = 8.  
 IF sy-subrc <> 0.  
 MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
 ENDIF.  
  
ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method ZCL\_ESL\_UTIL=>DOWNLOAD\_DATA*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IV\_DATA TYPE STRING*  
*\* | [--->] IV\_FILE TYPE STRING*  
*\* | [--->] IV\_PREF TYPE STRING*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
 METHOD DOWNLOAD\_DATA.  
  
 DATA: lt\_temptable TYPE table\_of\_strings,  
 lv\_path TYPE string,  
 lv\_len TYPE i,  
 lv\_pref TYPE string,  
 lv\_ext TYPE string,  
 lv\_data TYPE string,  
 lv\_xstring TYPE xstring,  
 lt\_xtable TYPE solix\_tab,  
 lv\_string TYPE string,  
 lv\_date\_time TYPE string,  
 lv\_data\_str TYPE string,  
 lv\_data\_utf8 TYPE string.  
 DATA: lor\_conv TYPE REF TO cl\_abap\_conv\_in\_ce.  
  
 lv\_path = iv\_file.  
 lv\_len = strlen( lv\_path ).  
  
 CONCATENATE lv\_path '\_sy-datum '\_' sy-uzeit '\_' iv\_pref '.xml' INTO lv\_path.  
  
*\*выгрузка в текстовом формате*  
 CALL FUNCTION 'SCMS\_STRING\_TO\_XSTRING'  
 EXPORTING  
 text = iv\_data  
 IMPORTING  
 buffer = lv\_xstring  
 EXCEPTIONS  
 failed = 1  
 OTHERS = 2.  
 IF sy-subrc <> 0.  
 MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
 ENDIF.  
  
 CONCATENATE cl\_abap\_char\_utilities=>byte\_order\_mark\_utf8  
 lv\_xstring  
 INTO lv\_xstring IN BYTE MODE.  
  
 CALL FUNCTION 'SCMS\_XSTRING\_TO\_BINARY'  
 EXPORTING  
 buffer = lv\_xstring  
 IMPORTING  
 output\_length = lv\_len  
 TABLES  
 binary\_tab = lt\_xtable.  
  
 CALL FUNCTION 'GUI\_DOWNLOAD'  
 EXPORTING  
 filetype = 'BIN'  
 bin\_filesize = lv\_len  
 filename = lv\_path  
 TABLES  
 data\_tab = lt\_xtable.  
  
 ENDMETHOD.  
  
  
*SIGNATURE>*  
 METHOD GET\_OGRN.  
  
 DATA:  
 lt\_struc TYPE TABLE OF struc,  
 lt\_objec TYPE TABLE OF objec,  
 ls\_objec TYPE objec.  
  
 CALL FUNCTION 'RH\_STRUC\_GET'  
 EXPORTING  
 act\_otype = 'P'  
 act\_objid = iv\_pernr  
 act\_wegid = 'P-O-OR'  
 TABLES  
 result\_objec = lt\_objec  
 result\_struc = lt\_struc  
 EXCEPTIONS  
 no\_plvar\_found = 1  
 no\_entry\_found = 2  
 OTHERS = 3.  
 IF sy-subrc <> 0.  
 MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
 WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
 ENDIF.  
  
 LOOP AT lt\_objec INTO ls\_objec  
 WHERE  
 otype = 'OR' AND  
 begda <= sy-datum AND  
 endda >= sy-datum.  
  
 SELECT SINGLE kogrn  
 INTO rv\_ogrn  
 FROM hrp1655  
 WHERE objid = ls\_objec-objid AND  
 otype = 'OR' AND  
 plvar = '01'.  
  
 EXIT.  
 ENDLOOP.  
  
 ENDMETHOD.  
  
*SIGNATURE>*  
 METHOD STR\_TO\_BASE64.  
  
 DATA: lv\_datalen TYPE i,  
 lt\_data TYPE rmps\_ssfbin,  
 lv\_data TYPE xstring.  
  
 CALL FUNCTION 'SCMS\_STRING\_TO\_XSTRING'  
 EXPORTING  
 text = iv\_str  
 IMPORTING  
 buffer = lv\_data  
 EXCEPTIONS  
 failed = 1  
 OTHERS = 2.  
 IF sy-subrc <> 0.  
 MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
 ENDIF.  
  
 CALL FUNCTION 'SCMS\_XSTRING\_TO\_BINARY'  
 EXPORTING  
 buffer = lv\_data  
 IMPORTING  
 output\_length = lv\_datalen  
 TABLES  
 binary\_tab = lt\_data.  
  
 CALL FUNCTION 'SSFC\_BASE64\_ENCODE'  
 EXPORTING  
 bindata = lv\_data  
 binleng = lv\_datalen  
 IMPORTING  
 b64data = rv\_str\_base64  
 EXCEPTIONS  
 ssf\_krn\_error = 1  
 ssf\_krn\_noop = 2  
 ssf\_krn\_nomemory = 3  
 ssf\_krn\_opinv = 4  
 ssf\_krn\_input\_data\_error = 5  
 ssf\_krn\_invalid\_par = 6  
 ssf\_krn\_invalid\_parlen = 7  
 OTHERS = 8.  
 IF sy-subrc <> 0.  
 MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
 ENDIF.  
  
 ENDMETHOD.  
  
  
*SIGNATURE>*  
 method STR\_TO\_XSTR.  
  
 CALL FUNCTION 'SCMS\_STRING\_TO\_XSTRING'  
 EXPORTING  
 text = iv\_string  
 IMPORTING  
 buffer = rv\_xstring  
 EXCEPTIONS  
 failed = 1  
 OTHERS = 2.  
 IF sy-subrc <> 0.  
 MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
 ENDIF.  
  
 endmethod.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method ZCL\_ESL\_UTIL=>XML\_TO\_TEXT*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IT\_XML TYPE J3RXR\_REP\_XML\_TAB*  
*\* | [<-()] RT\_RESPONSE TYPE SDOKCNTASCS*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
METHOD xml\_to\_text.  
  
 DATA:  
 lt\_bintab TYPE rmps\_ssfbin,  
 lv\_datalen TYPE i,  
 ls\_xml TYPE j3rxr\_rep\_xml\_tline.  
  
*\*перегоним xml в текстовую табличку*  
 LOOP AT it\_xml INTO ls\_xml.  
  
 CALL FUNCTION 'SCMS\_XSTRING\_TO\_BINARY'  
 EXPORTING  
 buffer = ls\_xml-xml\_cont  
 IMPORTING  
 output\_length = lv\_datalen  
 TABLES  
 binary\_tab = lt\_bintab.  
  
 CALL FUNCTION 'SCMS\_BINARY\_TO\_TEXT'  
 EXPORTING  
 input\_length = lv\_datalen  
 TABLES  
 binary\_tab = lt\_bintab  
 text\_tab = rt\_response  
 EXCEPTIONS  
 failed = 1  
 OTHERS = 2.  
 IF sy-subrc <> 0.  
 MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
 WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
 ENDIF.  
 ENDLOOP.  
  
ENDMETHOD.  
  
  
*\* <SIGNATURE>---------------------------------------------------------------------------------------+*  
*\* | Static Public Method ZCL\_ESL\_UTIL=>XSTR\_TO\_STR*  
*\* +-------------------------------------------------------------------------------------------------+*  
*\* | [--->] IV\_XSTRING TYPE XSTRING*  
*\* | [<-()] RV\_STRING TYPE STRING*  
*\* +--------------------------------------------------------------------------------------</SIGNATURE>*  
 METHOD xstr\_to\_str.  
  
 DATA:  
 lt\_bintab TYPE rmps\_ssfbin,  
 lv\_data\_len TYPE i.  
  
 CALL FUNCTION 'SCMS\_XSTRING\_TO\_BINARY'  
 EXPORTING  
 buffer = iv\_xstring  
 IMPORTING  
 output\_length = lv\_data\_len  
 TABLES  
 binary\_tab = lt\_bintab.  
  
 CALL FUNCTION 'SCMS\_BINARY\_TO\_STRING'  
 EXPORTING  
 input\_length = lv\_data\_len  
 IMPORTING  
 text\_buffer = rv\_string  
 TABLES  
 binary\_tab = lt\_bintab  
 EXCEPTIONS  
 failed = 1  
 OTHERS = 2.  
 IF sy-subrc <> 0.  
 MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
 WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
 ENDIF.  
  
 ENDMETHOD.  
ENDCLASS.\_