



Bankleitzahlen einlesen

Mal wieder ein Thema aus dem FI-Bereich. Diesmal geht es um die Aktualisierung der Bankleitzahlen. Die Bundesbank bietet die aktuellen Bankleitzahlen zum Download an. Weitere Infos gibt es hier.

Das Einlesen der Bankleitzahlen ist dabei gar nicht die größte Herausforderung, sondern das Finden der aktuellen Bankleitzahlendatei. Markus Völker hat hierfür eine kleine Klasse geschrieben, mit der

1. die URL der Bankleitzahlendatei ermittelt wird
2. Die Bankleitzahlen ins SAP-System eingelesen werden

Verbesserungspotential

Die Klasse ermittelt und liest die Bankleitzahlen ein. Eventuelle wäre es jedoch wünschenswert zu sehen, welche und wie viele Debitoren von den Änderungen betroffen sind. Es müsste dafür eine Prüfung gegen die Tabelle KNBK gemacht werden.

Proxy

Falls du dich mit User und Kennwort am Proxy anmelden musst, dann muss du nach der Methode `cl_http_client=>create_by_url` den folgenden Aufruf einbauen:

```
lo_client->propertytype_logon_popup = http_client->co_disabled.  
call method lo_client->authenticate  
    exporting  
        username = 'myProxyUser'  
        password = 'myProxyPassword'.
```

Dementsprechend muss der CONSTRUCTOR um die beiden Parameter `username` und `password` erweitert werden.

Coding

```
REPORT  zrep_bbank_vs_bnka.  
**&-----*  
**& Report  ZREP_BBANK_VS_BNKA  
**&  
**&-----*  
**& Dieser Report demonstriert die Benutzung der Klasse lcl_cmp_bnka_to_bbank  
**& Author: Markus Völker  
**&-----*
```

```
PARAMETERS p_proxy TYPE string LOWER CASE DEFAULT '123.100.100.88'.
PARAMETERS p_pport TYPE string          DEFAULT `80`.
```

```
CLASS lcl_cmp_bnka_to_bbank DEFINITION.
  PUBLIC SECTION.
```

```
  TYPES:
```

```
    BEGIN OF mts_bnka_result,
      status TYPE char01,
      bankl  TYPE bankl,
    END OF mts_bnka_result .
```

```
  TYPES:
```

```
    mtt_bnka_result TYPE STANDARD TABLE OF mts_bnka_result WITH DEFAULT KEY
```

```
  .
```

```
**Vergleiche:
```

```
**http://www.bundesbank.de/Redaktion/DE/Downloads/Aufgaben/Unbarer\_Zahlungsverkehr/Bankleitzahlen/merkblatt\_bankleitzahldatei.pdf?\_\_blob=publicationFile
```

```
  TYPES:
```

```
    BEGIN OF mts_bundesbank,
      blz(8), "Bankleitzahl
      merkmal(1), "Merkmal, ob bankleitzahlführender Zahlungsdienstleister
      ("1") oder nicht ("2")
      bezeichnung(58), "Bezeichnung des Zahlungsdienstleisters
      plz(5), "Postleitzahl
      ort(35), "Ort
      kurzbezeichnung(27), "Kurzbezeichnung des Zahlungsdienstleisters mit
      Ort
```

```
      pan(5), "Institutsnummer für PAN
      bic(11), "Business Identifier Code (BIC)
      pz(2), "Kennzeichen für Prüfzifferberechnungsmethode
      datensatz(6), "Nummer des Datensatzes
      aenderungs(1), "Änderungskennzeichen
      blz_loesch(1), "Hinweis auf eine beabsichtigte Bankleitzahllöschung
      nach_blz(8), "Hinweis auf Nachfolge-Bankleitzahl
      knz_iban(6), "Kennzeichen für die IBAN-Regel
```

```
    END OF mts_bundesbank .
```

```
  TYPES:
```

```
    mtt_bundesbank TYPE STANDARD TABLE OF mts_bundesbank WITH DEFAULT KEY .
```

```
  CONSTANTS mc_bank_changes TYPE char01 VALUE 'C'.      "#EC NOTEXT "Die
  Bank ist zur Änderung vorgesehen
```

```
  CONSTANTS mc_bank_correct TYPE char01 VALUE 'O'.      "#EC NOTEXT "Die
  Bank ist in Ordnung
```

```
  CONSTANTS mc_bank_missing TYPE char01 VALUE 'M'.      "#EC NOTEXT "Die
  Bank existiert nicht mehr
```

```
  CLASS-DATA mv_proxy_host TYPE string .                "Proxy IP
```

```
  CLASS-DATA mv_proxy_service TYPE string .             "Proxy Port
```

```
  DATA      mt_bbank_datei TYPE mtt_bundesbank .      "Datei der
  Bundesbank
```

```

METHODS constructor
  IMPORTING
    iv_proxy_host    TYPE string OPTIONAL
    iv_proxy_service TYPE string OPTIONAL .
METHODS set_bbank_file
  IMPORTING
    iv_path          TYPE csequence OPTIONAL
  EXPORTING
    ev_subrc         TYPE sy-subrc
    ev_err_message  TYPE string.
TYPE-POOLS abap .
METHODS compare_bbank_with_bnka
  IMPORTING
    iv_with_missing    TYPE flag DEFAULT abap_true
    iv_with_changes    TYPE flag DEFAULT abap_true
    iv_with_correct    TYPE flag DEFAULT abap_true
  RETURNING
    VALUE(rt_result_tab) TYPE mtt_bnka_result .
PROTECTED SECTION.
METHODS get_bbank_file_from_web
  EXPORTING
    ev_subrc         TYPE sy-subrc
    ev_err_message  TYPE string
    et_filetab      TYPE table .

```

ENDCLASS.

CLASS lcl_cmp_bnka_to_bbank IMPLEMENTATION.

```

* -----+-----
* | Instance Public Method COMPARE_BBANK_WITH_BNKA
* +-----+-----
* | [--->] IV_WITH_MISSING          TYPE          FLAG (default
=ABAP_TRUE)
* | [--->] IV_WITH_CHANGES        TYPE          FLAG (default
=ABAP_TRUE)
* | [--->] IV_WITH_CORRECT        TYPE          FLAG (default
=ABAP_TRUE)
* | [<-()] RT_RESULT_TAB          TYPE          MTT_BNKA_RESULT
* +-----+-----

```

METHOD compare_bbank_with_bnka.

```

DATA: lt_bnka  TYPE TABLE OF bnka,
      lv_subrc TYPE sy-subrc.

```

```

FIELD-SYMBOLS: <bnka>  TYPE bnka,
               <bbank> TYPE mts_bundesbank,

```

<result> TYPE mts_bnka_result.

DEFINE append_result.

append initial line to rt_result_tab assigning <result>.

<result>-status = &1.

<result>-bankl = &2.

END-OF-DEFINITION.

* Wenn wir noch keine Daten haben, holen wir halt welche

IF mt_bbank_datei IS INITIAL.

set_bbank_file(IMPORTING ev_subrc = lv_subrc).

IF lv_subrc <> 0.

RETURN.

ENDIF.

ENDIF.

* Alle Banken holen

SELECT * FROM bnka INTO TABLE lt_bnka WHERE banks = 'DE'.

* Vergleich der Daten

*/-----\

-----\

LOOP AT lt_bnka ASSIGNING <bnka>.

* Lese die Vergleichsdaten

READ TABLE mt_bbank_datei ASSIGNING WITH KEY blz = <bnka>-bnklz. "#EC

WARNOK.

IF sy-subrc <> 0.

* Die Bank existiert nicht mehr

append_result mc_bank_missing <bnka>-bankl.

ELSEIF -blz_loesch = '1'.

* Die Bank ist zum Löschen vorgemerkt

append_result mc_bank_changes <bnka>-bankl.

ELSE.

* Keine Veränderungen festzustellen

append_result mc_bank_correct <bnka>-bankl.

ENDIF.

ENDLOOP.

*/-----\

-----/

* Lösche ungewünschtes

*/-----\

-----\

IF iv_with_missing = abap_false.

DELETE rt_result_tab WHERE status = mc_bank_missing.

ENDIF.

IF iv_with_changes = abap_false.

DELETE rt_result_tab WHERE status = mc_bank_changes.

ENDIF.

IF iv_with_correct = abap_false.

DELETE rt_result_tab WHERE status = mc_bank_correct.

```

ENDIF.
*\-----
-----/

ENDMETHOD.
"compare_bbank_with_bnka

* -----
-----+
* | Instance Public Method ZZ_CMP_BNKA_TO_BBANK->CONSTRUCTOR
* +-----
-----+
* | [--->] IV_PROXY_HOST                TYPE          STRING(optional)
* | [--->] IV_PROXY_SERVICE              TYPE          STRING(optional)
* +-----
-----
METHOD constructor.
* Das hier ist nur die Erinnerung, eventuell einen Proxy setzen zu müssen
  mv_proxy_host    = iv_proxy_host .
  mv_proxy_service = iv_proxy_service.
ENDMETHOD.                                "constructor

* -----
-----+
* | Instance Protected Method GET_BBANK_FILE_FROM_WEB
* +-----
-----+
* | [<---] EV_SUBRC                      TYPE          SY-SUBRC
* | [<---] EV_ERR_MESSAGE                 TYPE          STRING
* | [<---] ET_FILETAB                     TYPE          TABLE
* +-----
-----
METHOD get_bbank_file_from_web.
* Hole die Datei der Bundesbank per HTTP
  DATA: lv_url      TYPE string,
         lv_datvon   TYPE d,
         lv_datbis   TYPE d,
         lo_http     TYPE REF TO if_http_client,
         lv_recdata  TYPE xstring,
         lt_tmpbin   TYPE TABLE OF raw256,
         lv_tmplen   TYPE i.

*****
* DATUMSBERECHNUNG
*****

*      Monat/Jahr von:
*/-----
-----\

```

```

lv_datvon = sy-datum.
DO.
    lv_datvon = cl_hrpdate_computations=>subtract_months_from_date(
start_date = lv_datvon          " Datum von dem Monate subtrahiert
werden
months      = 1 ).              " Anzahl Monate
    IF lv_datvon+4(2) MOD 3 = 0.
        EXIT.
    ENDIF.
ENDDO.
*   Tag von ...
lv_datvon+6(2) = '01'.
DO.
*   Finde den ersten Samstag im Monat ...
    IF cl_hrpdate_computations=>get_weekday_number( lv_datvon ) = 6.
"Samstag
        EXIT.
    ENDIF.
    lv_datvon = lv_datvon + 1.
ENDDO.
    lv_datvon = lv_datvon + 2.              "Wir brauchen aber
den Montag
*\-----/
----/
*   Monat/Jahr bis:
*/-----/
----\
    lv_datbis = sy-datum.
DO.
    IF lv_datbis+4(2) MOD 3 = 0.
        EXIT.
    ENDIF.
    lv_datbis = cl_hrpdate_computations=>add_months_to_date( start_date
= lv_datbis          " Datum von dem Monate subtrahiert werden
months
= 1 ).              " Anzahl Monate
ENDDO.
*   Tag von ...
lv_datbis+6(2) = '01'.
DO.
*   Finde den ersten Samstag im Monat ...
    IF cl_hrpdate_computations=>get_weekday_number( lv_datbis ) = 6.
"Samstag
        EXIT.
    ENDIF.
    lv_datbis = lv_datbis + 1.
ENDDO.
    lv_datbis = lv_datbis + 1.              "geht bis Sonntag
*\-----/
----/

```

```

* URL Zusammenbauen
*/-----\
-----\
    lv_url =
'http://www.bundesbank.de/Redaktion/DE/Downloads/Aufgaben/Unbarer_Zahlungsver
kehr/Bankleitzahlen/' ##no_text
    && |{ lv_datbis(4) }_{ lv_datbis+4(2) }_{ lv_datbis+6(2) }/blz_{
lv_datvon(4) }_{ lv_datvon+4(2) }_{ lv_datvon+6(2)
}_txt.txt?__blob=publicationFile|.
*\-----\
-----/

*****
* HTTP - Verbindung
*****
* Erstellen des Verbindungsobjekts
    cl_http_client=>create_by_url( EXPORTING url          = lv_url
                                proxy_host       = mv_proxy_host
" logische Destination (Wird bei Funktionsaufruf angegeben)
                                proxy_service    = mv_proxy_service
" Portnummer
                                IMPORTING
                                client          = lo_http ).
" HTTP Client Abstraction

* Request senden
    lo_http->send( EXCEPTIONS OTHERS = 1 ).
    IF sy-subrc <> 0.
        lo_http->get_last_error( IMPORTING code      = ev_subrc
                                message           = ev_err_message ).

        RETURN.
    ENDIF.

* Daten annehmen
    lo_http->receive( EXCEPTIONS OTHERS = 1 ).
    IF sy-subrc <> 0.
        lo_http->get_last_error( IMPORTING code      = ev_subrc
                                message           = ev_err_message ).

        RETURN.
    ENDIF.

* Datei reinladen...
    lv_recddata = lo_http->response->get_data( ).
    IF lv_recddata IS INITIAL.
        ev_subrc = 1024.
        RETURN.
    ENDIF.

* Daten in eine "Datei" - Tabelle konvertieren
*/-----\
-----\

```

```
CALL FUNCTION 'SCMS_XSTRING_TO_BINARY'
  EXPORTING
    buffer      = lv_recddata
  IMPORTING
    output_length = lv_tmplen
  TABLES
    binary_tab   = lt_tmpbin.
```

```
CALL FUNCTION 'SCMS_BINARY_TO_TEXT'
  EXPORTING
    input_length = lv_tmplen
  TABLES
    binary_tab   = lt_tmpbin
    text_tab     = et_filetab.
```

```
*\-----/
-----/
```

```
ENDMETHOD.
"GET_BBANK_FILE_FROM_HTML
```

```
* -----/
-----+
* | Instance Public Method ZZ_CMP_BNKA_TO_BBANK->SET_BBANK_FILE
* +-----/
-----+
* | [--->] IV_PATH                TYPE          CSEQUENCE(optional)
* | [<---] EV_SUBRC                TYPE          SY-SUBRC
* | [<---] EV_ERR_MESSAGE          TYPE          STRING
* | [EXC!] FAILED_VIA_UPLOAD
* +-----/
```

```
METHOD set_bbank_file.
```

```
DATA: lt_filetab TYPE TABLE OF char512.
FIELD-SYMBOLS: <file> TYPE char512,
               <bbank> TYPE mts_bundesbank.
```

```
IF iv_path IS NOT INITIAL AND cl_gui_frontend_services=>file_exist(
iv_path ) = abap_true.
```

```
* Datei einfach vom Frontend hochladen
  cl_gui_frontend_services=>gui_upload( EXPORTING
                                         filename = iv_path
" Name der Datei
                                         CHANGING
                                         data_tab = lt_filetab
" Übergabetabelle für Datei-Inhalt
                                         EXCEPTIONS
                                         OTHERS   = 1 ).
  ev_subrc = sy-subrc.
ELSE.
```



```

* Datei per HTTP holen
  me->get_bbank_file_from_web( IMPORTING ev_subrc      = ev_subrc
                               ev_err_message = ev_err_message
                               et_filetab    = lt_filetab ).

  ENDIF.

* Wenn die Datei leer ist machen wir hier nichts
  IF lt_filetab IS INITIAL.
    RETURN.
  ENDIF.

* In die richtige Struktur kippen:
  LOOP AT lt_filetab ASSIGNING <file>.
    APPEND INITIAL LINE TO mt_bbank_datei ASSIGNING <bbank>.
    <bbank> = <file>.
  ENDLOOP.
  ENDMETHOD.                                "set_bbank_file
ENDCLASS.

DATA gr_bbank TYPE REF TO lcl_cmp_bnka_to_bbank.
DATA gv_subrc TYPE sy-subrc.
DATA gv_err   TYPE string.
DATA gt_result TYPE lcl_cmp_bnka_to_bbank=>mtt_bnka_result.
FIELD-SYMBOLS TYPE lcl_cmp_bnka_to_bbank=>mts_bnka_result.

START-OF-SELECTION.

* Objekt instanziiieren
  CREATE OBJECT gr_bbank
    EXPORTING
      iv_proxy_host    = p_proxy           " logische
Destination
      iv_proxy_service = p_pport.         " Portnummer

* Datei innerhalb der Klasse organisieren
  gr_bbank->set_bbank_file( IMPORTING
                               ev_subrc      = gv_subrc      " Rückgabewert von
ABAP-Anweisungen
                               ev_err_message = gv_err        " Error - Message
                               ).

  IF gv_subrc <> 0.
    WRITE: / gv_err.
    RETURN.
  ENDIF.

* Vergleich der Daten aus der Bundesbank und SAP holen
  gt_result = gr_bbank->compare_bbank_with_bnka( ).

* primitive Ausgabe... :)
  LOOP AT gt_result ASSIGNING .

```

```
CASE -status.  
  WHEN lcl_cmp_bnka_to_bbank=>mc_bank_missing.  
    WRITE: / 'MISSING'.  
  WHEN lcl_cmp_bnka_to_bbank=>mc_bank_changes.  
    WRITE: / 'CHANGING'.  
  WHEN lcl_cmp_bnka_to_bbank=>mc_bank_correct.  
    WRITE: / 'OK'.  
ENDCASE.  
WRITE: 12 '|', -bankl.  
ENDLOOP.
```