  METHOD zpi\_ii\_rs\_bp\_update~rs\_bp\_update.  
  
\*\*\* Deklaration der Variablen \*\*\*  
    DATA: v\_gp      TYPE bu\_partner,  
          v\_gp\_type TYPE bu\_type,  
          v\_fail(1) TYPE c.  
  
\*\*\* Deklaration lokaler Tabellen \*\*\*  
    DATA: lt\_telefondatanonaddress   TYPE TABLE OF bapiadtel,  
          lt\_telefondatanonaddress\_x TYPE TABLE OF bapiadtelx,  
          lt\_e\_maildatanonaddress    TYPE TABLE OF bapiadsmtp,  
          lt\_e\_maildatanonaddress\_x  TYPE TABLE OF bapiadsmtx,  
          lt\_message\_detail          TYPE zpi\_bp\_update\_response\_mes\_tab,  
          lt\_tel\_change              TYPE  zpi\_bp\_update\_request\_tel\_\_tab,  
          lt\_mail\_change             TYPE  zpi\_bp\_update\_request\_e\_ma\_tab,  
          lt\_return                  TYPE TABLE OF bapiret2.  
  
\*\*\* Deklaration lokaler Strukturen \*\*\*  
    DATA: ls\_centraldata               TYPE bapibus1006\_central,  
          ls\_centraldataperson         TYPE bapibus1006\_central\_person,  
          ls\_centraldataorganization   TYPE bapibus1006\_central\_organ,  
          ls\_centraldatagroup          TYPE bapibus1006\_central\_group,  
          ls\_centraldata\_x             TYPE bapibus1006\_central\_x,  
          ls\_centraldataperson\_x       TYPE bapibus1006\_central\_person\_x,  
          ls\_centraldataorganization\_x TYPE bapibus1006\_central\_organ\_x,  
          ls\_centraldatagroup\_x        TYPE bapibus1006\_central\_group\_x,  
          ls\_message\_detail            LIKE LINE OF lt\_message\_detail,  
          ls\_message\_list              TYPE zpi\_bp\_update\_response\_message,  
          ls\_telefondatanonaddress     LIKE LINE OF lt\_telefondatanonaddress,  
          ls\_e\_maildatanonaddress      LIKE LINE OF lt\_e\_maildatanonaddress,  
          ls\_tel\_change                LIKE LINE OF lt\_tel\_change,  
          ls\_mail\_change               LIKE LINE OF lt\_mail\_change,  
          ls\_telefondatanonaddress\_x   LIKE LINE OF lt\_telefondatanonaddress\_x,  
          ls\_e\_maildatanonaddress\_x    LIKE LINE OF lt\_e\_maildatanonaddress\_x,  
          ls\_return                    LIKE LINE OF lt\_return.  
  
    v\_gp = input-bp\_update\_request-gp\_no. " Übergabe der Inputparameter  
  
\*\*\* Die GP Nummer konvertieren\*\*\*  
    CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
      EXPORTING  
        input  = v\_gp  
      IMPORTING  
        output = v\_gp.  
  
    SELECT SINGLE type FROM but000 INTO v\_gp\_type WHERE partner = v\_gp. "Select Abfrage zum auslesen des GP Typs  
  
    IF sy-subrc EQ '0'.  
\*\*\* Ändern der Namen / je nach GP Typ müssen die neuen Werte in andere Strukturen übergeben werden  
      CASE v\_gp\_type.  
        WHEN '1'. " Privatperosn  
          IF input-bp\_update\_request-name\_2 IS NOT INITIAL. " Falls neuer Wert für Name 2  
            ls\_centraldataperson-firstname = input-bp\_update\_request-name\_2. " neuer Wert der lokalen Struktur übergeben  
            ls\_centraldataperson\_x-firstname = 'X'. " X für Änderung setzen  
            ENDIF.  
          IF input-bp\_update\_request-name\_1 IS NOT INITIAL. " Falls neuer Wert für Name 1  
            ls\_centraldataperson-lastname = input-bp\_update\_request-name\_1.  
            ls\_centraldataperson\_x-lastname = 'X'.  
          ENDIF.  
  
        WHEN '2'. " Firma  
          IF input-bp\_update\_request-name\_1 IS NOT INITIAL. " Falls neuer Wert für Name 1  
            ls\_centraldataorganization-name1 = input-bp\_update\_request-name\_1.  
            ls\_centraldataorganization\_x-name1 = 'X'.  
          ENDIF.  
          IF input-bp\_update\_request-name\_2 IS NOT INITIAL. " Falls neuer Wert für Name 2  
            ls\_centraldataorganization-name2 = input-bp\_update\_request-name\_2.  
            ls\_centraldataorganization\_x-name2 = 'X'.  
          ENDIF.  
  
        WHEN '3'. " Gruppe  
          IF input-bp\_update\_request-name\_1 IS NOT INITIAL. " Falls neuer Wert für Name 1  
            ls\_centraldatagroup-namegroup1 = input-bp\_update\_request-name\_1.  
            ls\_centraldatagroup\_x-namegroup1 = 'X'.  
          ENDIF.  
          IF input-bp\_update\_request-name\_2 IS NOT INITIAL. " Falls neuer Wert für Name 2  
            ls\_centraldatagroup-namegroup2 = input-bp\_update\_request-name\_2.  
            ls\_centraldatagroup\_x-namegroup2 = 'X'.  
          ENDIF.  
      ENDCASE.  
  
\*\*\* Fals Privatperson überprüfen ob neue Anrede \*\*\*  
      IF v\_gp\_type EQ '1' AND input-bp\_update\_request-title IS NOT INITIAL.  
  
        CASE input-bp\_update\_request-title.  
          WHEN 'FRAU'.  
            ls\_centraldata-title\_key = '0001'.  
            ls\_centraldata\_x-title\_key = 'X'.  
          WHEN 'HERR'.  
            ls\_centraldata-title\_key = '0002'.  
            ls\_centraldata\_x-title\_key = 'X'.  
          WHEN 'HERR U. FRAU'.  
            ls\_centraldata-title\_key = '0004'.  
            ls\_centraldata\_x-title\_key = 'X'.  
          WHEN 'FRAUEN'.  
            ls\_centraldata-title\_key = '0005'.  
            ls\_centraldata\_x-title\_key = 'X'.  
          WHEN 'HERREN'.  
            ls\_centraldata-title\_key = '0006'.  
            ls\_centraldata\_x-title\_key = 'X'.  
          WHEN OTHERS.  
            ls\_message\_detail-message\_id = 'I'.  
            ls\_message\_detail-message\_text = 'Anrede konnte nicht geändert werden. Ungültige Anredeform.'.  
            APPEND ls\_message\_detail TO lt\_message\_detail.  
            v\_fail = 'X'.  
        ENDCASE.  
      ENDIF.  
  
\*\*\* Neue Werte für Telefon auslesen und übergeben \*\*\*  
      IF input-bp\_update\_request-tel\_change\_list IS NOT INITIAL.  
        LOOP AT input-bp\_update\_request-tel\_change\_list-tel\_change\_detail INTO ls\_tel\_change.  
          ls\_telefondatanonaddress-telephone = ls\_tel\_change-tel\_number.  
          ls\_telefondatanonaddress-consnumber = ls\_tel\_change-tel\_id.  
          APPEND ls\_telefondatanonaddress TO lt\_telefondatanonaddress.  
  
        ls\_telefondatanonaddress\_x-telephone = 'X'.  
        ls\_telefondatanonaddress\_x-consnumber = 'X'.  
        APPEND ls\_telefondatanonaddress\_x TO lt\_telefondatanonaddress\_x.  
        ENDLOOP.  
      ENDIF.  
  
\*\*\* Neue Werte für Mail auslesen und übergeben \*\*\*  
      IF input-bp\_update\_request-e\_mail\_change\_list IS NOT INITIAL.  
        LOOP AT input-bp\_update\_request-e\_mail\_change\_list-e\_mail\_detail INTO ls\_mail\_change.  
          ls\_e\_maildatanonaddress-e\_mail = ls\_mail\_change-e\_mail\_adress.  
          ls\_e\_maildatanonaddress-consnumber = ls\_mail\_change-e\_mail\_id.  
          APPEND ls\_e\_maildatanonaddress TO lt\_e\_maildatanonaddress.  
  
        ls\_e\_maildatanonaddress\_x-e\_mail = 'X'.  
        ls\_e\_maildatanonaddress\_x-consnumber = 'X'.  
        Append ls\_e\_maildatanonaddress\_x to lt\_e\_maildatanonaddress\_x.  
        ENDLOOP.  
      ENDIF.  
  
\*\*\* Funktionsaufruf von BAPI\_BUPA\_CENTRAL\_CHANGE mit Übergabe der Strukturen mit den neuen Werte  
        CALL FUNCTION 'BAPI\_BUPA\_CENTRAL\_CHANGE'  
          EXPORTING  
            businesspartner           = v\_gp  
            centraldata               = ls\_centraldata  
            centraldataperson         = ls\_centraldataperson  
            centraldataorganization   = ls\_centraldataorganization  
            centraldatagroup          = ls\_centraldatagroup  
            centraldata\_x             = ls\_centraldata\_x  
            centraldataperson\_x       = ls\_centraldataperson\_x  
            centraldataorganization\_x = ls\_centraldataorganization\_x  
            centraldatagroup\_x        = ls\_centraldatagroup\_x  
            duplicate\_message\_type    = '-'  
  
          TABLES  
          telefondatanonaddress     = lt\_telefondatanonaddress  
          e\_maildatanonaddress      = lt\_e\_maildatanonaddress  
          telefondatanonaddressx    = lt\_telefondatanonaddress\_x  
          e\_maildatanonaddressx     = lt\_e\_maildatanonaddress\_x  
          return                    = lt\_return.  
  
        IF sy-subrc EQ '0' AND lt\_return IS INITIAL. " Fals Funktionsaufruf erfolgreich und ohne Fehlermeldung  
          CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'. " Befehl zur Speichern der neuen Werte  
          IF sy-subrc = '0'. " Speicherung erfolgreich  
            ls\_message\_detail-message\_id = 'S'.  
            ls\_message\_detail-message\_text = 'Daten erfolgreich geändert'.  
            APPEND ls\_message\_detail TO lt\_message\_detail.  
            ls\_message\_list-message\_details = lt\_message\_detail.  
          ELSE. " Speicherung nicht erfolgreich  
            ls\_message\_detail-message\_id = 'I'.  
            ls\_message\_detail-message\_text = 'Beim speichern der Daten ist ein Fehler aufgetreten.'.  
          ENDIF.  
  
        ELSE. " Funktionsaufruf nicht erfolgreich / mit Fehlermeldung  
          IF lt\_return IS INITIAL.  
            ls\_message\_detail-message\_id = 'I'.  
            ls\_message\_detail-message\_text = 'Beim ändern der Daten ist ein Fehler aufgetreten'.  
          ELSE.  
  
            LOOP AT lt\_return INTO ls\_return.  
              ls\_message\_detail-message\_id = ls\_return-type.  
              ls\_message\_detail-message\_text = ls\_return-message.  
            ENDLOOP.  
          ENDIF.  
  
          APPEND ls\_message\_detail TO lt\_message\_detail." Tabelle mit Fehlermeldungen an lokale Struktur übergeben  
          ls\_message\_list-message\_details = lt\_message\_detail.  
        ENDIF.  
    ELSE. " Fehler beim ermitteln des GP Typs  
  
      ls\_message\_detail-message\_id = 'I'.  
      ls\_message\_detail-message\_text = 'Fehler beim ermitteln des Geschäftpartner-Typs'.  
      APPEND ls\_message\_detail TO lt\_message\_detail.  
  
    ENDIF.  
  
    ls\_message\_list-message\_details = lt\_message\_detail.  
    output-bp\_update\_response-message\_list = ls\_message\_list." lokale Struktur mit allen Meldungen an Output übergeben  
  ENDMETHOD.