METHOD zpi\_ii\_rs\_bp\_search~rs\_bp\_search.  
  
\*\*\* Deklaration interne Tabellen \*\*\*  
    DATA: lt\_searchresult   TYPE TABLE OF bapibus1006\_bp\_addr,  
          lt\_return         TYPE TABLE OF bapiret2,  
          lt\_person\_details TYPE          zpi\_bp\_search\_response\_per\_tab.  
  
  
\*\*\* Deklaration interne Strukturen \*\*\*  
    DATA: ls\_searchresult              LIKE LINE OF   lt\_searchresult,  
          ls\_return                    LIKE LINE OF   lt\_return,  
          ls\_i\_addressdata             TYPE           bapibus1006\_addr\_search,  
          ls\_i\_centraldata             TYPE           bapibus1006\_central\_search,  
          ls\_person\_details            LIKE LINE OF   lt\_person\_details,  
          ls\_persons\_list              TYPE           zpi\_bp\_search\_response\_persons,  
          ls\_e\_centraldataperson       TYPE           bapibus1006\_central\_person,  
          ls\_e\_centraldataorganization TYPE           bapibus1006\_central\_organ,  
          ls\_e\_centraldatagroup        TYPE           bapibus1006\_central\_group,  
          ls\_message                   TYPE           zpi\_bp\_search\_response\_message.  
  
\*\*\* Deklaration Variablen \*\*\*  
    DATA: v\_gp\_typ TYPE string,  
          v\_count  TYPE i.  
   
  
\*\*\* Übergabe der Inputparameter der lokalen Strukturen \*\*\*  
    ls\_i\_centraldata-partner = input-bp\_search\_request-gp\_no.  
    ls\_i\_centraldata-mc\_name1 = input-bp\_search\_request-name\_1.  
    ls\_i\_centraldata-mc\_name2 = input-bp\_search\_request-name\_2.  
    ls\_i\_centraldata-partnercategory = input-bp\_search\_request-gp\_type.  
    ls\_i\_addressdata-city1 = input-bp\_search\_request-city.  
    ls\_i\_addressdata-street = input-bp\_search\_request-street.  
    ls\_i\_addressdata-house\_no = input-bp\_search\_request-house\_no.  
    ls\_i\_addressdata-postl\_cod1 =  input-bp\_search\_request-plz.

\*\*\* Aufruf Methode BAPI\_BUPA\_SEARCH\_2 zur Suche von GPs  
    CALL FUNCTION 'BAPI\_BUPA\_SEARCH\_2'  
      EXPORTING  
        addressdata  = ls\_i\_addressdata  
        centraldata  = ls\_i\_centraldata  
      TABLES  
        searchresult = lt\_searchresult " Tabelle mit allen gefunden Resultaten  
        return       = lt\_return. " Tabelle mit Infomrationen zur abgeschlossenen Funktiosaufruf (Fehlermedlungen)  
  
    IF lt\_searchresult IS NOT INITIAL. " Überprüfen ob ein GP gefunden wurde  
      LOOP AT lt\_searchresult INTO ls\_searchresult. " Loop über alle gefunden GPs  
\*\*\* Aufruf Methode BAPI\_BUPA\_CENTRAL\_GETDETAIL zur Beschaffung vom GP Name \*\*\*  
        CALL FUNCTION 'BAPI\_BUPA\_CENTRAL\_GETDETAIL'  
          EXPORTING  
            businesspartner         = ls\_searchresult-partner  
  
          IMPORTING  
  
            centraldataperson       = ls\_e\_centraldataperson  
            centraldataorganization = ls\_e\_centraldataorganization  
            centraldatagroup        = ls\_e\_centraldatagroup  
  
          TABLES  
            return                  = lt\_return.  
          
        IF lt\_return IS INITIAL. " Überprüfung der Fehlermeldungen / Falls Fehlermeldung vorhanden weiter in Else  
\*\*\* Falls kein GP Typ mitgegeben von Webservice : Select auf der BUT000 Tabelle und GP TYP auslesen  
          IF input-bp\_search\_request-gp\_type IS INITIAL.  
            SELECT SINGLE type FROM but000 INTO v\_gp\_typ WHERE partner = ls\_searchresult-partner.  
          ENDIF.  
  
\*\*\* Firmen- Gruppen- oder Personenname auslesen abhänig von GP Typ \*\*\*\*  
          CASE v\_gp\_typ .  
            WHEN '1' .  
              ls\_person\_details-name\_1 = ls\_e\_centraldataperson-lastname.  
              ls\_person\_details-name\_2 = ls\_e\_centraldataperson-firstname.  
            WHEN '2'.  
              ls\_person\_details-name\_1 = ls\_e\_centraldataorganization-name1.  
              ls\_person\_details-name\_2 = ls\_e\_centraldataorganization-name2.  
            WHEN '3'.  
              ls\_person\_details-name\_1 = ls\_e\_centraldatagroup-namegroup1.  
              ls\_person\_details-name\_2 = ls\_e\_centraldatagroup-namegroup2.  
            WHEN OTHERS.  
          ENDCASE.

\*\*\* Personeninformationen in Personentabelle übergeben\*\*\*\*  
          ls\_person\_details-gp\_no = ls\_searchresult-partner.  
          ls\_person\_details-gp\_type = v\_gp\_typ.  
          APPEND ls\_person\_details TO lt\_person\_details.  
  
        ELSE.  
        "  Bei einem Fehler während der Funktion auftritt: Über die Fehlermeldung Loppen  
        "  und Fehlermeldung dem Output übergeben. Loop verlassen und keine Personendaten dem Output übergeben.  
          LOOP AT lt\_return INTO ls\_return.  
            output-bp\_search\_response-message-message\_text = ls\_return-message.  
            output-bp\_search\_response-message-message\_type = ls\_return-type.  
          ENDLOOP.  
          Clear lt\_person\_details.  
          Exit.  
        ENDIF.  
  
\*\*\* Sortieren der Personentabelle und löschen doppelter Einträge  
        SORT lt\_person\_details ASCENDING BY gp\_no.  
        DELETE ADJACENT DUPLICATES FROM lt\_person\_details COMPARING gp\_no.  
  
\*\*\* Die Personentabelle dem Output übergeben  
        ls\_persons\_list-person\_details = lt\_person\_details.  
        output-bp\_search\_response-persons\_list = ls\_persons\_list.  
      ENDLOOP.  
  
    ELSE.  
\*\*\* Fall keinen GPS gefunden wurden : return Nachricht dem Output übergeben.  
      LOOP AT lt\_return INTO ls\_return.  
        output-bp\_search\_response-message-message\_type = ls\_return-type.  
        output-bp\_search\_response-message-message\_text = ls\_return-message.  
      ENDLOOP.  
    ENDIF.  
  
\*\*\* Anzahl Einträge in der Perosnentabelle zählen und dem output übergeben  
    DESCRIBE TABLE lt\_person\_details LINES v\_count.  
    output-bp\_search\_response-result\_count = v\_count.  
  
  ENDMETHOD.