How to Create a Hierarchy Extractor for Employees and their Department Managers

Applies to:
This paper and the programs are checked on ECC 5 or higher (640 or 700) with plug-in PI 2005_1 (for 640 or 700).

Summary
In the BW Business Content, there’s no DataSource available for providing a hierarchical structure for an employee. In typical cases, in BW managers want to have access to their subordinate employees and/or the data they are responsible for/authorized to. This paper will show you how to create a Hierarchy DataSource for an employee object.

Author(s): Luc Boodts (luc.boodts@sap.com)
Company: SAP Belgium
Created on: October 2007

Author Bio
Luc Boodts has been working for SAP Belgium for 9 years, currently in the position of Principal Consultant NetWeaver BI. He has over 8 years of experience in the SAP HR domain in various modules. For SAP he has been working in the areas of HR, BW and EP and has good ABAP knowledge.
# Table of Contents

How to Create a Hierarchy Extractor for Employees and their Department Managers

- Business Scenario ........................................................................................................................................ 3
- Introduction ................................................................................................................................................ 3
- Limitations ................................................................................................................................................. 3

Step by step description................................................................................................................................4

- Create Function Group .............................................................................................................................. 4
- Create Function Module for Data Extractor ............................................................................................... 4
- Global Data Definition ................................................................................................................................6
- Create Subroutines in Subprograms-Include Program ................................................................................ 8
- Create Function Module for Hierarchy Catalog ....................................................................................... 12
- Customizing in Source System ............................................................................................................... 14
- Create the BW Datasource in the Source System .................................................................................... 15
- Create the Corresponding InfoObject in BW ........................................................................................... 16
- Create an InfoPackage to Test/Load Data ............................................................................................... 17

Appendix – DataSource Creation Program ................................................................................................. 21

Copyright ..................................................................................................................................................... 24
**Business Scenario**

You need to be able to grant access to grant managers access to the data their subordinates are responsible for. There’s no standard BI Content to provide you with an employee hierarchy. There’s a hierarchy provided in the BI Content for Organizational Unit, but this does not really suit, since you want the manager of the org.unit to be able to see the data, without using organization unit.

With the hierarchy on an employee object, you can grant managers access through the authorizations based on a hierarchy. This hierarchy will contain their personnel number, which you can then pre-fill with an exit-variable in the authorizations (RSECADMIN).

Possible scenarios are: give access to managers for cost centers for which their subordinates are responsible, profit centers, administrator groups in HR, projects, …

**Introduction**

The extractor/DataSource described hereafter is based on the Organization Unit Hierarchy DataSource, and uses a similar process to extract the data. It makes use of SAP ERP standard function modules for Human Resources.

In appendix a coding sample is provided, to make the necessary table updates for your hierarchy extractor to work.

**Limitations**

This document only describes how to set up the hierarchy extractor and not the authorization setup.

The extractor only allows having 1 manager for an organizational unit at one point in time, and one employee can only belong to one organizational unit. This is because of the limitations of a BW Hierarchy (one object can only occur once at one point in time). The extractor extracts the data on system date and does not include time dependency.
Step by step description

Create Function Group

Create function group ZBW_EMPL_HIERARCHY (use transaction SE37 : Goto – Function Groups – Create function group).

Create Function Module for Data Extractor

Create the function module by copying from the standard function module(easiest).

Make sure you have the correct import parameters

Make sure you have the correct tables specifications
Also check that following exceptions are specified.

Enter the source code below in the function module

**Source code**

```plaintext
FUNCTION Z_BW_EMPL_HIERARCHY_TRANSFER.
**"----------------------------------------------------------------------
**"Local interface:
**" IMPORTING
**"     VALUE(I_S_HIEBAS) TYPE RSAP_S_HIEBAS
**"     VALUE(I_S_HIEFLAG) TYPE RSAP_S_HIEFLAG
**"     VALUE(I_S_HIERSEL) TYPE RSAP_S_HIER_LIST
**"     TABLES
**"       I_T_LANGU TYPE SBIWA_T_LANGU
**"       E_T_HIETEXT TYPE RSAP_T_HIETEXT
**"       E_T_HIENODE TYPE RSAP_T_HIENODE
**"       E_T_FOLDERT TYPE RSAP_T_FOLDERT OPTIONAL
**"       E_T_HIEINTV TYPE RSAP_T_HIEINTV OPTIONAL
**"     EXCEPTIONS
**"       INVALID_CHABASNM_HCLASS
**"       INVALID_HIERARCHY_FLAG
**"       INVALID_HIERARCHY_SELECT
**"       LANGU_NOT_SUPPORTED
**"       HIERARCHY_TAB_NOT_FOUND
**"       APPLICATION_ERROR
**"----------------------------------------------------------------------

DATA: LOC_HIENODE LIKE ROSHIENODE OCCURS 0 WITH HEADER LINE,
     LOC_HIETEXT LIKE ROSHIETEXT OCCURS 0 WITH HEADER LINE.
```
DATA: SUBRC LIKE SY-SUBRC.

* build hierarchy
PERFORM FILL_STRUCTURES TABLES LOC_HIENODE
   LOC_HIETEXT
   USING I_S_HIEBAS-CHABASNM
   I_S_HIERSEL
   CHANGING SUBRC.

   IF SUBRC = 0.
      APPEND LINES OF LOC_HIENODE TO E_T_HIENODE.
   * description of hierarchy
   PERFORM FILL_HIETEXT TABLES LOC_HIETEXT
      I_T_LANGU
      USING I_S_HIEBAS-CHABASNM
      I_S_HIEBAS-HCLASS
      I_S_HIERSEL
      CHANGING SUBRC.

   IF SUBRC = 0.
      APPEND LINES OF LOC_HIETEXT TO E_T_HIETEXT.
   ** description of virtual root = description of hierarchy
   * LOOP AT LOC_HIETEXT.
   * MOVE-CORRESPONDING LOC_HIETEXT TO E_T_FOLDERT.
   * E_T_FOLDERT-IOBJNM   = '0HIER_NODE'.
   * E_T_FOLDERT-NODENAME = 'ROOT'.
   * APPEND E_T_FOLDERT.
   * ENDLOOP.
   ENDIF.
else.
   raise hierarchy_tab_not_found.
ENDIF.
ENDFUNCTION.

Global Data Definition

Create the global data definitions (Goto – Main Program)

Edit the program LZW_EML_HIERARCHYTOP.
Paste the code below in the program LZBW_EMPL_HIERARCHYTOP-program.

Source code

```
FUNCTION-POOL ZBW_EMPL_HIERARCHY.  "MESSAGE-ID ..

INCLUDE RHODAT00.

* type-pool rhe inserted due to deco issues
  TYPE-POOLS: RSAP, SBIWA.

* RODCHABAS-RSHIE: Hierarchiesteuerkennzeichen zu Basismetkmalen
* HIETABFL Merkmal mit Hierarchien

* HIEVERFL Hierarchien können mehrere Versionen haben
  TYPES: RHIE_VERSION LIKE RSHIERTRSF-VERSION.
  CONSTANTS: RHIE_VERSION_ZERO TYPE RHIE_VERSION VALUE '000'.

* HIENMTFL Hierarchiename ist zeitabhängig
* HIENDTFL Hierarchiestruktur ist zeitabhängig
  TYPES: RHIE_DATE LIKE ROSHIENODE-DATETO.
  CONSTANTS: RHIE_C_MINDATE TYPE RHIE_DATE VALUE '19000101',
             RHIE_C_MAXDATE TYPE RHIE_DATE VALUE '99991231'.

* HIEINTFL Intervalle in Hierarchie zugelassen

* Sonstige type
  TYPES: RHIE_SUBRC LIKE SY-SUBRC.
  TYPES: RHIE_FORMNAME LIKE TFDIR-FUNCNAME.

* Dates
  DATA LOWDATE LIKE PLOG-ENDDA VALUE '19000101'.
  DATA HIGHDATE LIKE PLOG-ENDDA VALUE '99991231'.

  tables: hrp1000.
```
Create Subroutines in Subprograms-Include Program

Create subroutine collection by editing the program LZW_EMPL_HIERARCHYF01 and enter the source code below.

Source code

```plaintext
*---------------------------------------------------------------------*
*       FORM FILL_STRUCTURES                                          *
*       ........                                                      *
*---------------------------------------------------------------------*
*  -->  HIENODE                                                       *
*  -->  HIETEXT                                                       *
*  -->  VALUE(I_CHABASNM)                                             *
*  -->  VALUE(I_HIEM)                                                *
*  -->  VALUE(I_VERSION)                                              *
*  -->  VALUE(I_HCLASS)                                               *
*  -->  VALUE(I_S_TIMEINT)                                            *
*  -->  SUBRC                                                         *
*---------------------------------------------------------------------*
FORM fill_structures TABLES hienode STRUCTURE roshienode
   hietext STRUCTURE roshietext
USING value(i_chabasnm) LIKE rssiobj-iobjnm
   value(i_hiersel) LIKE roshiersel
CHANGING subrc LIKE sy-subrc.

types : begin of x_orgman,
   orgunit like hrp1001-objid,
   person like hrp1001-objid,
end of x_orgman.

types : begin of x_manag.
   include structure hrp1001.

types : sobid2 like hrp1001-objid,
end of x_manag.

types : begin of x_manag2,
   sobid like hrp1001-sobid,
   sobid2 like hrp1001-objid,
end of x_manag2.

types : begin of shorthier,
   SEQNR(10), " like struct-SEQNR,
   LEVEL(10), " like struct-LEVEL,
end of shorthier.
```
OTYPE like struc-OTYPE,
OBJID like struc-OBJID,
PDOWN(10), " like struc-PDOWN,
DFLAG like struc-DFLAG,
VCOUNT(10), " like struc-VCOUNT,
PNEXT(10), " like struc-PNEXT,
PUP(10), " like struc-PUP,
PPREV(10), " like struc-PPREV,
VRSIGN like struc-VRSIGN,
VRELAT like struc-VRELAT,
end of shorthier.

data : t_hier type struc_t,
t_hier2 type struc_t,
t_hier3 type table of shorthier,
t_manag type table of x_manag,
t_manag2 type hashed table of x_manag2 with unique key sobid,
t_person like hashed table of hrp1001 with unique key objid,"endda begda sobid,
t_persons like table of hrp1001,
t_persons2 like table of hrp1001,
t_orgman type hashed table of x_orgman with unique key orgunit,"person,
w_manag type x_manag,
w_person like hrp1001,
w_orgman type x_orgman,
w_manag2 type x_manag2,
w_hier type struc,
w_hier3 type shorthier,
c_hier3 type shorthier,
w_hiepers like hrp1001-objid.

CALL FUNCTION 'HR_STRUCTURE_GET'
EXCEPTING
ROOT_PLVAR                = '01'
ROOT_OTYPE                = 'O'
ROOT_OBJID                = I_HIERSEL-HIENM
*   ROOT_OBJECTS              =
*   BEGDA                     = SY-DATUM
*   ENDDA                     = SY-DATUM
*   pathid                    = 'ZOS-PNOS'
*   PATHID_IS_INTERN          =
*   STRU_TECH_DEPTH           = 0
*   STRU_STATUS_VECTOR        = '1'
*   STRU_STATUS_OVERLAP       = '
*   PROVIDE_TEXT              = '
*   PROVIDE_RELAT             = 'X'
*   PROVIDE_DFLAG             = 'X'
*   RECURSION_CHECK           = 'X'
*   AUTHORITY_CHECK           = '
*   TEXT_BUFFER_FILL          = '
*   READ_MODE                 = 'F'
*   KEEP_ORDER                = 'X'

IMPORTING
*   RESULT_OBJECTS            =
*   RESULT_STRUCTURE          = t_hier
*   ROOT_COPY                 =

EXCEPTIONS
*   PLVAR_NOT_FOUND           = 1
*   ROOT_NOT_FOUND            = 2
*   PATH_NOT_FOUND            = 3
*   INTERNAL_ERROR            = 4
*   OTHERS                    = 5

IF sy-subrc <> 0.
*   MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
   WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.
SELECT * into table t_manag
FROM hrp1001
  WHERE relat = '012'
  AND sclas = 'S'
  AND otype = 'O'
  AND plvar = '01'
  AND endda ge sy-datum
  AND rsign = 'B'
  AND istat = '1'
  AND begda le sy-datum.
if sy-subrc ne 0.
endif.
loop at t_manag into w_manag.
  move w_manag-sobid to w_manag-sobid2.
  modify t_manag from w_manag.
endloop.

**************
loop at t_manag into w_manag.
  w_manag2-sobid = w_manag-sobid.
  w_manag2-sobid2 = w_manag-sobid2.
  insert w_manag2 into table t_manag2.
endloop.
sort t_manag2 by sobid.
delete adjacent duplicates from t_manag2.
SELECT * into table t_persons
FROM hrp1001
for all entries in t_manag2
  WHERE relat = '008'
  AND sclas = 'P'
  AND otype = 'S'
  AND plvar = '01'
  AND endda ge sy-datum
  AND rsign = 'A'
  AND istat = '1'
  and objid = t_manag2-sobid2
  AND begda le sy-datum.
if sy-subrc ne 0.
endif.
t_persons2[] = t_persons[].
loop at t_persons transporting no fields
  where objid = '53008773'
  and sobid = '00044568'.
    delete t_persons.
endloop.
delete adjacent duplicates from t_persons comparing objid.
t_person[] = t_persons[].
if t_persons2[] ne t_persons[].
  CALL FUNCTION 'RSAL_LOG_WRITE'
    EXPORTING
      I_MSGTY = 'W'
      I.MSGID = 'RJ'
      I_MSGNO = '053'
      I.MSGV1 = 'Duplicate managers found'
      I.MSGV2 = ''.
endif.
loop at t_manag into w_manag.
  read table t_person into w_person
    with table key objid = w_manag-sobid.
  check sy-subrc eq 0.
move w_manag-objid to w_orgman-orgunit.
move w_person-sobid to w_orgman-person.
insert w_orgman into table t_orgman.
endloop.
*loop at t_hier into w_hier.
  if w_hier-otype eq 'O '.
  *endloop.

loop at t_hier into w_hier.
  move-corresponding w_hier to w_hier3.
  if w_hier3-otype eq 'O '.
    clear w_hiepers.
    read table t_orgman into w_orgman
    with table key orgunit = w_hier3-objid.
    if sy-subrc eq 0.
      move w_hier3 to c_hier3.
      move w_orgman-person to :w_hiepers, w_hier3-objid.
      move 'P ' to w_hier3-otype.
    endif.
  elseif w_hier3-otype eq 'P '.
    if w_hier3-objid eq w_hiepers.
      move c_hier3-otype to w_hier3-otype.
      move c_hier3-objid to w_hier3-objid.
      clear w_hiepers.
    endif.
  endif.
  append w_hier3 to t_hier3.
endloop.
loop at t_hier3 into w_hier3.
  move w_hier3-seqnr to hienode-NODEID.  "Internal ID number of a hierarchy node
  if w_hier3-otype eq 'P '.
    move i_chabasnm to hienode-IOBJNM.    "InfoObject
  elseif w_hier3-otype eq 'O '.
    move '0ORGUNIT' to hienode-IOBJNM.  "InfoObject
  endif.
  move w_hier3-objid to hienode-NODENAME. "Name of the hierarchy node
  move w_hier3-level to hienode-TLEVEL.   "Level of a hierarchy node
  *LINK "Link indicator for a hierarchy node
  move w_hier3-pup to hienode-PARENTID. "Parent ID for a hierarchy node
  move w_hier3-pdown to hienode-CHILDID. "Child ID of a hierarchy node
  move w_hier3-pnext to hienode-NEXTID.   "Next ID of a hierarchy node
  *DATEFROM"Valid-From Date
  *DATETO"Valid-to date
  *INTERVL"Flag: Node is interval
  append hienode.
*write : /
  ** w_hier3-SEQNR,
  ** w_hier3-LEVEL,
  * w_hier3-OTYPE,
  * w_hier3-OBJID,
  * w_hier3-PDOWN,
  ** w_hier3-DFLAG,
  ** w_hier3-VCOUNT,
  * w_hier3-PNEXT,
  * w_hier3-PUP,
  * w_hier3-PPREV,
  * w_hier3-VRSIGN,
  * w_hier3-VRELAT.
endform.                    " FILL_STRUCTURES
*---------------------------------------------------------------------*
*       FORM FILL_HIETEXT                                             *
*---------------------------------------------------------------------*
FORM fill_hietext TABLES hietext   STRUCTURE roshietext
How to Create a Hierarchy Extractor for Employees and their Department Managers

Create Function Module for Hierarchy Catalog

There is also a catalog function necessary to be able to view the available hierarchies in your source system (also visible from within BI). Copy the function module below to your own one (Z_BW_EMPL_HIERARCHY_CATALOG).

Check the import parameters

Check the tables specified

Check the exceptions
Copy the source code below into your function module

**Source code**

```plaintext
FUNCTION Z_BW_EMPL_HIERARCHY_CATALOG.
**"Local interface:"**
**" IMPORTING**
**"     VALUE(I_S_HIEBAS) TYPE  RSAP_S_HIEBAS**
**"     VALUE(I_S_HIEFLAG) TYPE  RSAP_S_HIEFLAG**
**"     VALUE(I_S_HIERSEL) TYPE  RSAP_S_HIER_LIST OPTIONAL**
**" TABLES**
**"     I_T_LANGU TYPE  SBIWA_T_LANGU**
**"     E_T_HIERS TYPE  RSAP_T_HIERS**
**" EXCEPTIONS**
**"     INVALID_CHABASNM_HCLASS**
**"     INVALID_HIERARCHY_FLAG**
**"     INVALID_HIERARCHY_SELECT**
**"     LANGU_NOT_SUPPORTED**
**"     HIERARCHY_NOT_FOUND**
**"     APPLICATION_ERROR**
**"**

DATA: I_RSHIERTRSF LIKE RSHIERTRSF OCCURS 0 WITH HEADER LINE.
DATA: SUBRC LIKE SY-SUBRC.
data : t_roots like table of HRROOTOB with header line.
* LOOP AT I_T_LANGU.
  CLEAR: I_RSHIERTRSF, I_RSHIERTRSF[].
  PERFORM FILL_I_RSHIERTRSF TABLES I_RSHIERTRSF
    USING I_S_HIEBAS-CHABASNM
    I_S_HIEBAS-HCLASS
    I_T_LANGU-LANGU
    I_S_HIERSEL
    CHANGING SUBRC.
  IF SUBRC = 0.
    APPEND LINES OF I_RSHIERTRSF TO E_T_HIERS.
  ENDIF.
  CALL FUNCTION 'RH_STRUC_HELP_ROOTS_GET'
    EXPORTING
      search_plvar = '01'
      search_otype = '0'
      SEARCH_WEGID =
      SEARCH_WEGID_INT =
      SEARCH_SVECT =
      SEARCH_BEGDA =
```
Customizing in Source System

The standard function modules that are used in the source code specified in this document are making use of “evaluation paths”.

Goto transaction SM30, specify table T778A and press “Maintain”. This action will prompt you for a customizing request. Make sure that this customizing is done in agreement with the responsables for the Human Resources module.
How to Create a Hierarchy Extractor for Employees and their Department Managers

**Maintain Table Views: Initial Screen**

![Table View Screen]

Click on “Create new entries”-button and enter the line specified below.

**New Entries: Overview of Added Entries**

![New Entries Screen]

Then in the left column, double-click on “Evaluation path (individual …) and enter following values :

**New Entries: Overview of Added Entries**

![Data Entry Screen]

Create the BW Datasource in the Source System

Before you can execute this, please create and run the program mentioned in appendix.

When you run the program, enter a new InfoObject name (not an existing one, since the program might overwrite existing datasource definitions) in field P_IOBJ.

As second parameter enter a name for your DataSource in field P_DSCRCE.
How to Create a Hierarchy Extractor for Employees and their Department Managers

If you want this program to actually create the DataSource and update the necessary tables, check the box for P_UPDATE.

Create the Corresponding InfoObject in BW

Use transaction RSA1 or RSD1 to create the InfoObject you have specified in the previous step. Specify for this object that it is using “Hierarchies” – “Hierarchies, not time-dependent”.

Change Characteristic ZRSPNSBL: Detail

Then use the button “External Chars. in Hierarchies”. This to include also the Organization Unit object, so that these are included in your hierarchy and which allow you to verify the result with the Organizational Structure used in the source system.

From the list of Characteristics, choose “0ORGUNIT”.

Then use the button “External Chars. in Hierarchies”. This to include also the Organization Unit object, so that these are included in your hierarchy and which allow you to verify the result with the Organizational Structure used in the source system.

From the list of Characteristics, choose “0ORGUNIT”.

© 2007 SAP AG
Create an InfoPackage to Test/Load Data

Create an InfoPackage for the Hierarchy on the InfoObject and press the button "Available hierarchies from OLTP". This will show you the available "root nodes" from your source system.

After you have selected one of the hierarchies and successfully loaded on, go and check the result (either via the monitor of the InfoPackage (using the tree-button) or via the InfoObject maintenance using the button 'Maintain Hierarchies')
How to Create a Hierarchy Extractor for Employees and their Department Managers

After loading, this is the result. The hierarchy contains personnel numbers, mixed with the IDs of the Organizational Unit included. If you want to have the description for the org. units, you can load the texts using the standard extractor.
How to Create a Hierarchy Extractor for Employees and their Department Managers

This is what it looks like in ERP:

<table>
<thead>
<tr>
<th>CORPORATE_RESPONSIBL</th>
<th>InfoObject</th>
<th>Node n</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>20000001</td>
<td>0ORGUNIT</td>
<td>20000001</td>
<td></td>
</tr>
<tr>
<td>20000002</td>
<td>0ORGUNIT</td>
<td>20000002</td>
<td></td>
</tr>
<tr>
<td>20000003</td>
<td>0ORGUNIT</td>
<td>20000003</td>
<td></td>
</tr>
<tr>
<td>20000004</td>
<td>0ORGUNIT</td>
<td>20000004</td>
<td></td>
</tr>
<tr>
<td>1157</td>
<td>ZRSPNSBL</td>
<td>00001157</td>
<td></td>
</tr>
<tr>
<td>20000005</td>
<td>0ORGUNIT</td>
<td>20000005</td>
<td></td>
</tr>
<tr>
<td>1158</td>
<td>ZRSPNSBL</td>
<td>00001158</td>
<td></td>
</tr>
<tr>
<td>1159</td>
<td>ZRSPNSBL</td>
<td>00001159</td>
<td></td>
</tr>
<tr>
<td>1162</td>
<td>ZRSPNSBL</td>
<td>00001162</td>
<td></td>
</tr>
<tr>
<td>1160</td>
<td>ZRSPNSBL</td>
<td>00001160</td>
<td></td>
</tr>
<tr>
<td>1650</td>
<td>ZRSPNSBL</td>
<td>00001650</td>
<td></td>
</tr>
<tr>
<td>20000006</td>
<td>0ORGUNIT</td>
<td>20000006</td>
<td></td>
</tr>
<tr>
<td>20000007</td>
<td>0ORGUNIT</td>
<td>20000007</td>
<td></td>
</tr>
<tr>
<td>20000008</td>
<td>0ORGUNIT</td>
<td>20000008</td>
<td></td>
</tr>
<tr>
<td>20000009</td>
<td>0ORGUNIT</td>
<td>20000009</td>
<td></td>
</tr>
<tr>
<td>20000014</td>
<td>0ORGUNIT</td>
<td>20000014</td>
<td></td>
</tr>
<tr>
<td>1150</td>
<td>ZRSPNSBL</td>
<td>00001150</td>
<td></td>
</tr>
<tr>
<td>20000015</td>
<td>0ORGUNIT</td>
<td>20000015</td>
<td></td>
</tr>
<tr>
<td>1153</td>
<td>ZRSPNSBL</td>
<td>00001153</td>
<td></td>
</tr>
<tr>
<td>1152</td>
<td>ZRSPNSBL</td>
<td>00001152</td>
<td></td>
</tr>
<tr>
<td>1151</td>
<td>ZRSPNSBL</td>
<td>00001151</td>
<td></td>
</tr>
<tr>
<td>1155</td>
<td>ZRSPNSBL</td>
<td>00001155</td>
<td></td>
</tr>
<tr>
<td>1166</td>
<td>ZRSPNSBL</td>
<td>00001166</td>
<td></td>
</tr>
<tr>
<td>1161</td>
<td>ZRSPNSBL</td>
<td>00001161</td>
<td></td>
</tr>
<tr>
<td>1154</td>
<td>ZRSPNSBL</td>
<td>00001154</td>
<td></td>
</tr>
<tr>
<td>20000016</td>
<td>0ORGUNIT</td>
<td>20000016</td>
<td></td>
</tr>
<tr>
<td>20000017</td>
<td>0ORGUNIT</td>
<td>20000017</td>
<td></td>
</tr>
<tr>
<td>20000018</td>
<td>0ORGUNIT</td>
<td>20000018</td>
<td></td>
</tr>
<tr>
<td>20000019</td>
<td>0ORGUNIT</td>
<td>20000019</td>
<td></td>
</tr>
</tbody>
</table>

This is what it looks like in ERP.
<table>
<thead>
<tr>
<th>Staff Assignments (Structure)</th>
<th>ID</th>
<th>Chief</th>
<th>Workflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC4YOU Corporate Group</td>
<td>O 20000001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC4YOU Corporate Office</td>
<td>O 20000002</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Germany</td>
<td>O 20000004</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Germany: Personal Computers</td>
<td>O 20000005</td>
<td>Gerlinde Habermas</td>
<td></td>
</tr>
<tr>
<td>Operator</td>
<td>S 00100273</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group Leader</td>
<td>S 20000007</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gerlinde Habermas</td>
<td>P 00011157</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sachbearbeiter Administrativ</td>
<td>20000009</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sachbearbeiter Administrativ</td>
<td>20000010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator Administration (D)</td>
<td>20000017</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator Administration (D)</td>
<td>20000019</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator Administration (D)</td>
<td>50000291</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Germany: Peripherals</td>
<td>O 20000006</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Germany: Smart PC Security</td>
<td>O 20000007</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Germany: Smart PC Home</td>
<td>O 20000008</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spain</td>
<td>O 20000009</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td>O 20000014</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USA: Personal Computers</td>
<td>O 20000015</td>
<td>Carola Wissing</td>
<td></td>
</tr>
<tr>
<td>Group Leader</td>
<td>S 20000006</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Carola Wissing</td>
<td>P 00011150</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sachbearbeiter Netzwerk (D)</td>
<td>S 20000003</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sachbearbeiter Netzwerk (D)</td>
<td>S 20000011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sachbearbeiter Netzwerk (D)</td>
<td>S 20000013</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator Netzwerk (D)</td>
<td>S 20000001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator Netzwerk (D)</td>
<td>S 20000002</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator Netzwerk (D)</td>
<td>S 20000008</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator Netzwerk (D)</td>
<td>S 20000016</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USA: Peripherals</td>
<td>O 20000016</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USA: Smart PC Security</td>
<td>O 20000017</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USA: Smart PC Home</td>
<td>O 20000018</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shared Services</td>
<td>O 20000019</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix – DataSource Creation Program

Create the program below and include the source code specified. This program will update the necessary table entries and thus actually create your DataSource.

```abap
REPORT ZBW_CREATE_EMPL_HIER_EXTRACTOR.

tables :
  roohiecat,
  roohiecom,
  rodcha,
  rodchabas,
  roosfield,
  roosource,
  roosourcet,
  roosseg,
  rohiebas.

data : w_domain like rodchabas-domanm value 'PERNR',
        w_dtelnm like rodcha-dtelnm value 'P_PERNR',
        w_iarea  like rodcha-applnm value 'PA-PA-IO',
        w_apco   like roosource-applnm value '0PA_OS-IO',
        w_text   like roosourcet-txtsh value 'Employee'.

data : t_rodcha     like table of rodcha with header line,
        t_rodchabas  like table of rodchabas with header line,
        t_roosource  like table of roosource with header line,
        t_roosourcet like table of roosourcet with header line,
        t_roosfield  like table of roosfield with header line,
        t_roohiecom  like table of roohiecom with header line,
        t_roohiebas  like table of rohiebas with header line.

parameters :   p_iobj  like rodcha-CHABASNM, "BW InfoObject
                p_dsrce like roosource-OLTPSOURCE, "Datasource name
                P_update as checkbox.  "Actually update ?

start-of-selection.

  * Update of table rodcha
    move p_iobj   to : t_rodcha-chabasnm, t_rodcha-chanm.
    move w_dtelnm to t_rodcha-dtelnm.
    move w_iarea  to t_rodcha-applnm.
    append t_rodcha.

  * Update of table rodchabas
    move p_iobj   to t_rodchabas-chabasnm.
    move w_domain to t_rodchabas-domanm.
    move 'X'      to t_rodchabas-hietabfl.
    append t_rodchabas.
```
* Update of table roosource
  move p_dsorce to t_roosource-OLTPSOURCE.
  move 'A' to t_roosource-objvers.
  move 'HIER' to t_roosource-type.
  move w_apco to t_roosource-applnm.
  move p_iobj to t_roosource-basosource.
  move 'F1' to t_roosource-EXMETHOD.
  move 'Z_BW_EMPL_HIERARCHY_TRANSFER' to t_roosource-EXTRACTOR.
  move 'ROHIEROS' to t_roosource-EXSTRUCT.
  move SY-UNAME to t_roosource-TSTPNM.
  move sy-datum to t_roosource-TSTPDAT.
  move sy-uzeit to t_roosource-TSTPTIM.
  move '1' to t_roosource-tfmethods.
  append t_roosource.

* Update of table roosourcet
  move p_dsorce to t_roosourcet-OLTPSOURCE.
  move 'A' to t_roosourcet-objvers.
  move 'EN' to t_roosourcet-langu.
  move w_text to : t_roosourcet-txtsh, t_roosourcet-txtmd, t_roosourcet-txtlg.
  append t_roosourcet.

* Update of table roosfield
  move p_dsorce to t_roosfield-OLTPSOURCE.
  move 'A' to t_roosfield-objvers.
  move 'OLTPSOURCE' to t_roosfield-field.
  move 'Y' to t_roosfield-notexrel.
  append t_roosfield.

* Update of table roohiecom
  move p_dsorce to t_roohiecom-OLTPSOURCE.
  move 'A' to t_roohiecom-objvers.
  move p_iobj to : t_roohiecom-fieldnm, t_roohiecom-fieldcmp.
  move '1' to t_roohiecom-posit.
  move w_dtelnm to t_roohiecom-dtelnm.
  append t_roohiecom.
  move p_dsorce to t_roohiecom-OLTPSOURCE.
  move 'A' to t_roohiecom-objvers.
  move '0ORGUNIT' to : t_roohiecom-fieldnm, t_roohiecom-fieldcmp.
  move '1' to t_roohiecom-posit.
  move 'ORGEH' to t_roohiecom-dtelnm.
  append t_roohiecom.

* Update of table roohiecat
  move p_dsorce to t_roohiecat-OLTPSOURCE.
  move 'A' to t_roohiecat-objvers.
  move 'HR01' to t_roohiecat-typehc.
  move w_domain to t_roohiecat-domain.
  move w_dtelnm to t_roohiecat-dtelnm.
  move 'X' to t_roohiecat-hietabfl.
  move 'Z_BW_EMPL_HIERARCHY_CATALOG' to t_roohiecat-FHCATALOG.
  move 'ROSHIERSEL' to t_roohiecat-STRUC_HDR.
  move 'ROSHIETEXT' to t_roohiecat-STRUC_HDRT.
  move 'ROSHIENODE' to t_roohiecat-STRUC_NODE.
  move 'ROSFOLDERT' to t_roohiecat-STRUC_NODT.
  move 'ROSHIEINTV' to t_roohiecat-STRUC_INTV.
  append t_roohiecat.

* Update of table roosseg
  move p_dsorce to t_roosseg-OLTPSOURCE.
  move 'A' to t_roosseg-objvers.
  move '0001' to t_roosseg-SEGID.
  move 'ROSHIERSEL' to t_roosseg-SEGSTRUCT.
  append t_roosseg.
move '0002' to t_roosseg-SEGID.
move 'ROSHIETEXT' to t_roosseg-SEGSTRUCT.
append t_roosseg.

move '0003' to t_roosseg-SEGID.
move 'ROSHIENODE' to t_roosseg-SEGSTRUCT.
append t_roosseg.

move '0004' to t_roosseg-SEGID.
move 'ROSFOLDERT' to t_roosseg-SEGSTRUCT.
append t_roosseg.

* Update of table rohiebas
move p_iobj to t_rohiebas-chabasnm.
move 'HR01' to t_rohiebas-hclass.
move w_domain to t_rohiebas-DOMANM.
move w_dtelnm to t_rohiebas-ROLLNM.
move 'Z_BW_EMPL_HIERARCHY_CATALOG' to t_rohiebas-FHCATALOG.
move 'Z_BW_EMPL_HIERARCHY_TRANSFER' to t_rohiebas-FHTRANSFER.
append t_rohiebas.

check p_update eq 'X'.
insert rodcha from table t_rodcha.
insert rodchabas from table t_rodchabas.
insert roosource from table t_roosource.
INSERT roosourcet from table t_roosourcet.
insert roosfield from table t_roosfield.
insert ROOHIECOM from table T_ROOHIECOM.
insert roohiecat from table t_roohiecat.
insert roosseg from table t_roosseg.
insert rohiebas from table t_rohiebas.
Copyright

© Copyright 2007 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, System i, System i5, System p, System p5, System x, System z, System z9, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, POWER5+, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies (“SAP Group”) for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided “as is” without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines/strings (“Code”) included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.