

Extraction of Hierarchy into Flat File from R/3 and Loading in BW System



Applies to:

This article applies to SAP R/3 (any version) and SAP B/W (any version). For more information, visit the [Business Intelligence homepage](#).

Summary

This document discusses the R/3 Tables involved in Extracting Hierarchies (mainly covered Cost Center Group Hierarchy and Cost Element Group Hierarchy), dump the hierarchy in Flat File in required BW structure and Load the same to BW.

Author: Anil Suryavansi

Company: Patni Computer Systems

Created on: 15 September 2009

Author Bio



Anil Suryavansi is presently working with Patni Computers Systems Ltd. He has worked extensively on SAP BIW 3.5, BI 7.0, different versions of R/3 and is mainly responsible for various BW and R/3 developments.

Table of Contents

Introduction	3
Cost Element Group Hierarchy in R/3	3
R/3 Tables Involved	4
Code to Extract the Hierarchy:.....	6
How Hierarchy is shown at BW Reports:.....	12
Related Content.....	13
Related Content.....	13
Disclaimer and Liability Notice.....	14

Introduction

In SAP R/3 for a Group (Ex: Cost Center Group or Cost Element Group), if any of Header node / Sub Node / Base Node is entered, hierarchy for that node appears. The same functionality can be achieved in B/W by loading a Single Hierarchy and with maintaining proper Link for an Organizational Unit. There are many scenarios in which it is required to Extract Hierarchies from R/3 tables, dump Hierarchy in Flat File and Load into B/W. Using same logic and code All the Group Hierarchies present in R/3 can be Extracted.

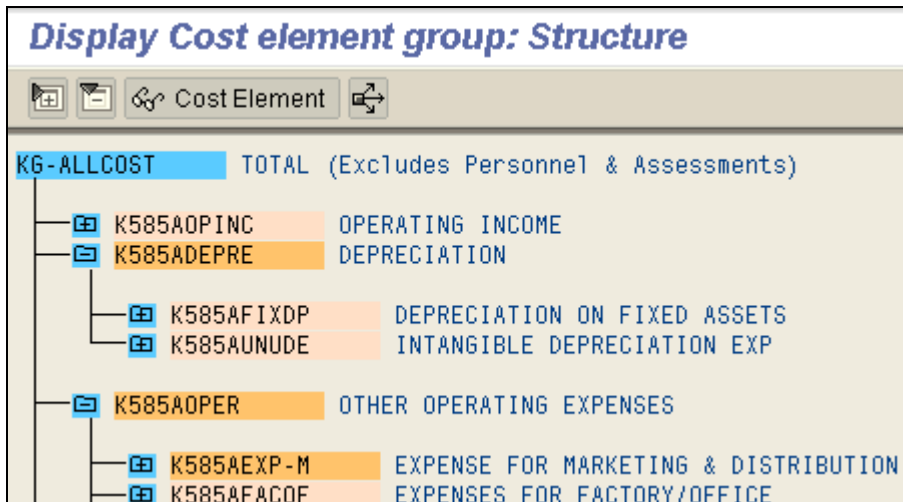
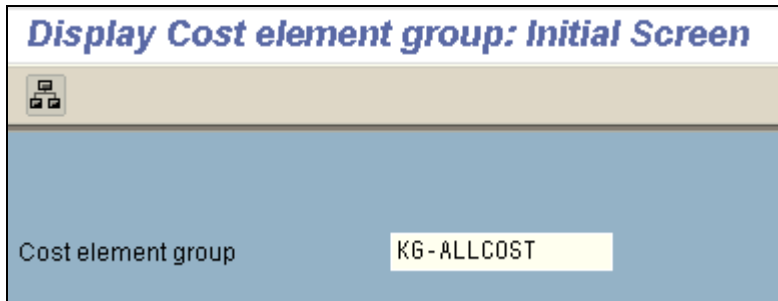
For Example:

- Cost Center Group Hierarchy
- Cost Element Group Hierarchy

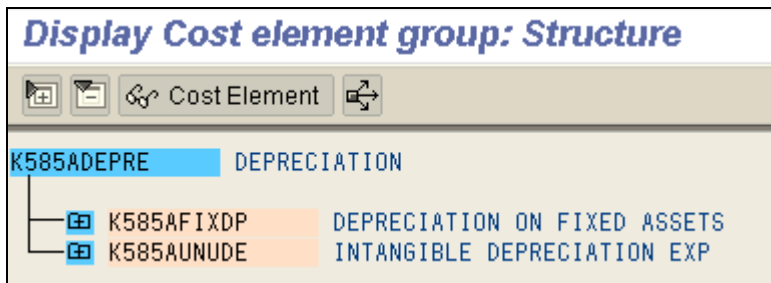
Cost Element Group Hierarchy in R/3

We can check Cost Element Group in R/3 by T-code KAH3.

For an example, we consider a group KG-ALLCOST:



Now if we check for K585ADEPRE:



Now if we check for K585AFIXDP:

Display Cost element group: Structure	
Cost Element	
K585AFIXDP	DEPRECIATION ON FIXED ASSETS
— 731010	ORDINARY DEPRECIATIONS ON INTANGIBLE FIX
— 731020	ORDINARY DEPRECIATION ON TANGIBLE FIXED A
— 731126	Depr.normal Other Factory/office equipmen
— 732020	EXT. DEPR. ON TANGIBLE FIXED ASSET
— 740010	LOSS ON DISPOSAL OF INTANGIBLE FIXED ASS
— 740020	LOSS ON DISPOSAL OF TANGIBLE FIXED ASSET

By above example it is possible to check hierarchy of KG-ALLCOST, K585ADEPRE and K585AFIXDP, as all these entries are maintained in SETHEADER table.

R/3 Tables Involved

Mainly 3 tables are involved to extract hierarchy:

- SETHEADER

The value of field Set class represents that the hierarchy is related to which field, as value 0102 is for Cost Element Group, 0101 is for Cost Center Group.

SCla	Description
0000	Set
0001	Dynamic set
0002	Temporary set
0003	Report set
0101	Cost center group
0102	Cost element group
0103	Order group
0104	Statistical key figure group
0105	Activity type group
0106	Profit center group
0107	Business process group
0108	Cost object group
0109	Account group
0110	WBS element group
0201	Set (capacity req. planning)
0202	Set (capacity req. planning)
0301	EC-CS cons. unit set
0302	EC-CS FS item set
0303	EC-CS FS subitem set

19 Entries found

If we check Set ID KG-ALLCOST (Higher Node) entry is available in this table.

Data Browser: Table SETHEADER Select Entries 1

Displayed fields: 17 of 21 Fixed columns: 4 List width 0250

Client	Set class	Org. unit	Set ID	Set type	Dynamic set	Authorization	M. unique	Repres. value
300	0102	WIKR	KG-ALLCOST	S			X	

- SETNODE

If we check table SETNODE with Set ID KG-ALLCOST, there are 12 Subset Ids available.

Data Browser: Table SETNODE Select Entries 12

Displayed fields: 9 of 9 Fixed columns: 5 List width 0250

Client	Set class	Org. unit	Set ID	Set line	Subset class	Subset subclass	Subset ID	Set line
300	0102	WIKR	KG-ALLCOST	0000000001	0102	WIKR	K585AOPINC	1
300	0102	WIKR	KG-ALLCOST	0000000004	0102	WIKR	K585ADEPRE	2
300	0102	WIKR	KG-ALLCOST	0000000005	0102	WIKR	K585AOPER	3
300	0102	WIKR	KG-ALLCOST	0000000006	0102	WIKR	K585AINVIC	4
300	0102	WIKR	KG-ALLCOST	0000000007	0102	WIKR	K585AINTPL	5
300	0102	WIKR	KG-ALLCOST	0000000008	0102	WIKR	K585AINVDR	6
300	0102	WIKR	KG-ALLCOST	0000000009	0102	WIKR	K585AFIXDP	7

If we now take any Subset ID ex. K585ADEPRE and check this as Set ID, there are two Subset IDs for this.

Data Browser: Table SETNODE Select Entries 2

Displayed fields: 9 of 9 Fixed columns: 5 List width 0250

Client	Set class	Org. unit	Set ID	Set line	Subset class	Subset subclass	Subset ID	Set line
300	0102	WIKR	K585ADEPRE	0000000010	0102	WIKR	K585AFIXDP	1
300	0102	WIKR	K585ADEPRE	0000000011	0102	WIKR	K585AUNUDE	2

Now if we take a Subset ID K585AFIXDP and check this as Set ID, there are no entries available in SETNODE table as it's a leaf node.

- SETLEAF

If we take K585AFIXDP and check this as Set ID in table SETLEAF, it contains values of Cost Elements assigned to the node.

Data Browser: Table SETLEAF Select Entries 6

Displayed fields: 10 of 10 Fixed columns: 5 List width 0250

Client	Set class	Org. unit	Set ID	Set line	Sign	option	From value	To value	Set line
300	0102	WIKR	K585AFIXDP	0000000033	I	EQ	0000731010	0000731010	1
300	0102	WIKR	K585AFIXDP	0000000034	I	EQ	0000731020	0000731020	2
300	0102	WIKR	K585AFIXDP	0000000035	I	EQ	0000731126	0000731126	3
300	0102	WIKR	K585AFIXDP	0000000036	I	EQ	0000732020	0000732020	4
300	0102	WIKR	K585AFIXDP	0000000037	I	EQ	0000740010	0000740010	5
300	0102	WIKR	K585AFIXDP	0000000038	I	EQ	0000740020	0000740020	6

Code to Extract the Hierarchy:

```

*****
*TYPES
*****
TYPES :BEGIN OF T_FL_SETHEADER,
        SETNAME TYPE SETHEADER-SETNAME,
END OF T_FL_SETHEADER.

TYPES: BEGIN OF T_FL_SETLEAF,
        SETNAME TYPE SETLEAF-SETNAME,
        LINEID TYPE SETLEAF-LINEID,
        VALOPTION TYPE SETLEAF-VALOPTION,
        VALFROM TYPE SETLEAF-VALFROM,
        VALTO TYPE SETLEAF-VALTO,
        LTEXT TYPE CSKU-LTEXT,
END OF T_FL_SETLEAF.

TYPES: BEGIN OF T_FL_CSKU,
        KSTAR TYPE CSKU-KSTAR,
        LTEXT TYPE CSKU-LTEXT,
END OF T_FL_CSKU.

TYPES: BEGIN OF T_FL_CSKT,
        KOSTL TYPE CSKT-KOSTL,
        LTEXT TYPE CSKT-LTEXT,
        DATBI TYPE CSKT-DATBI,
END OF T_FL_CSKT.

TYPES: BEGIN OF T_FL_HIER,
        MID(8) TYPE N,
        IOBJ(30),
        NNAME(32),
        PID(8) TYPE N,
        DESC(60),
END OF T_FL_HIER.

TYPES: BEGIN OF T_FL_SETNODE,
        SETNAME TYPE SETNODE-SETNAME,
        SUBSETNAME TYPE SETNODE-SUBSETNAME,
END OF T_FL_SETNODE.

TYPES: BEGIN OF T_FL_FINAL,
        OUT(500),
END OF T_FL_FINAL.

```

```

*****
*WORK AREAS
*****
DATA : VFL_CSKU TYPE T_FL_CSKU,
       VFL_CSKT TYPE T_FL_CSKT,
       VFL_SETLEAF TYPE T_FL_SETLEAF,
       VFL_SETHEADER TYPE T_FL_SETHEADER,
       VFL_HIER TYPE T_FL_HIER,
       VFL_HIER1 TYPE T_FL_HIER,
       VFL_SETNODE TYPE T_FL_SETNODE,
       VFL_FINAL TYPE T_FL_FINAL.

*****
*INTERNAL TABLES
*****
DATA : VIT_CSKU TYPE STANDARD TABLE OF T_FL_CSKU,
       VIT_CSKT TYPE STANDARD TABLE OF T_FL_CSKT,
       VIT_SETLEAF TYPE STANDARD TABLE OF T_FL_SETLEAF,
       VIT_SETLEAF1 TYPE STANDARD TABLE OF T_FL_SETLEAF,
       VIT_SETHEADER TYPE STANDARD TABLE OF T_FL_SETHEADER,
       VIT_SETHEADER1 TYPE STANDARD TABLE OF T_FL_SETHEADER,
       VIT_HIER TYPE STANDARD TABLE OF T_FL_HIER,
       VIT_HIER1 TYPE STANDARD TABLE OF T_FL_HIER,
       VIT_HIER2 TYPE STANDARD TABLE OF T_FL_HIER,
       VIT_SETNODE TYPE STANDARD TABLE OF T_FL_SETNODE,
       VIT_FINAL TYPE STANDARD TABLE OF T_FL_FINAL.

*****Variables*****
DATA : NID(8) TYPE N,
       PID(8) TYPE N,
       OUT(500),
       L_SETNAME LIKE SETHEADER-SETNAME,
       L_FLAG(2) TYPE N,
       L_LINK(1),
       L_KOSTL LIKE CSKT-KOSTL.

*****
*selection screen inputs
*****
SELECTION-SCREEN BEGIN OF BLOCK SS01 WITH FRAME TITLE TEXT-002.
PARAMETERS : P_IOBJ(9) OBLIGATORY.      "Name of BW InfoObject
SELECT-OPTIONS: S_SUBCLS FOR SETHEADER-SUBCLASS OBLIGATORY,
                S_SETCLS FOR SETHEADER-SETCLASS OBLIGATORY.
PARAMETER: C LZ AS CHECKBOX.          "Check to delete leading 0
SELECTION-SCREEN END OF BLOCK SS01.

```

```

SELECT SETNAME LINEID VALOPTION VALFROM VALTO INTO TABLE VIT_SETLEAF
FROM SETLEAF WHERE SUBCLASS IN S_SUBCLS.

SELECT SETNAME SUBSETNAME INTO TABLE VIT_SETNODE FROM SETNODE
WHERE SETCLASS IN S_SETCLS AND SUBCLASS IN S_SUBCLS.

SELECT KSTAR LTEXT INTO TABLE VIT_CSKU
FROM CSKU WHERE KTOPL IN S_SUBCLS AND SPRAS = 'EN'.

SELECT SETNAME INTO TABLE VIT_SETHEADER
FROM SETHEADER
WHERE FIELDNAME = 'KSTAR'
AND TABNAME = 'CCSS'
AND SUBCLASS IN S_SUBCLS.

SORT VIT_CSKU BY KSTAR.

LOOP AT VIT_SETLEAF INTO VFL_SETLEAF.
IF VFL_SETLEAF-VALOPTION = 'EQ'.
  READ TABLE VIT_CSKU INTO VFL_CSKU WITH KEY
      KSTAR = VFL_SETLEAF-VALFROM BINARY SEARCH.
  IF SY-SUBRC = 0.
    VFL_SETLEAF-LTEXT = VFL_CSKU-LTEXT.
    IF VFL_SETLEAF-LTEXT IS INITIAL.
      SELECT SINGLE KTEXT FROM CSKU INTO VFL_SETLEAF-LTEXT
      WHERE SPRAS = 'E' AND KTOPL IN S_SUBCLS
      AND KSTAR = VFL_SETLEAF-VALFROM.
    ENDIF.
    APPEND VFL_SETLEAF TO VIT_SETLEAF1.
  ENDIF.
  CLEAR VFL_CSKU.
ELSEIF VFL_SETLEAF-VALOPTION = 'BT'.
  LOOP AT VIT_CSKU INTO VFL_CSKU WHERE
    KSTAR GE VFL_SETLEAF-VALFROM AND KSTAR LE VFL_SETLEAF-VALTO.
    VFL_SETLEAF-VALFROM = VFL_CSKU-KSTAR.
    VFL_SETLEAF-VALTO = VFL_CSKU-KSTAR.
    VFL_SETLEAF-LTEXT = VFL_CSKU-LTEXT.
    IF VFL_SETLEAF-LTEXT IS INITIAL.
      SELECT SINGLE KTEXT FROM CSKU INTO VFL_SETLEAF-LTEXT
      WHERE SPRAS = 'E' AND KTOPL IN S_SUBCLS
      AND KSTAR = VFL_SETLEAF-VALFROM.
    ENDIF.
    APPEND VFL_SETLEAF TO VIT_SETLEAF1.
  CLEAR VFL_CSKU.
ENDIF.
ENDLOOP.
CLEAR VIT_SETLEAF.

```



```

SORT VIT_SETLEAF1 BY SETNAME VALFROM.
IF NOT VIT_SETLEAF1 IS INITIAL.
NID = 00000001.
PID = 00000000.
VFL_HIER-NID = NID.
VFL_HIER-PID = PID.
VFL_HIER-IOBJ = 'OHIER_NODE'.
VFL_HIER-NNAME = 'ALL-BASENODE-VALUES'.
VFL_HIER-DESC = 'All Base Nodes'.
NID = NID + 1.
APPEND VFL_HIER TO VIT_HIER.
CLEAR: VFL_SETLEAF, VFL_HIER.
ENDIF.

LOOP AT VIT_SETLEAF1 INTO VFL_SETLEAF.
  AT NEW SETNAME.
    PID = 00000001.
    VFL_HIER-NID = NID.
    VFL_HIER-PID = PID.
    VFL_HIER-IOBJ = 'OHIER_NODE'.
    VFL_HIER-NNAME = VFL_SETLEAF-SETNAME.
    SELECT SINGLE DESCRIPT FROM SETHEADERT INTO VFL_HIER-DESC
      WHERE SETNAME = VFL_SETLEAF-SETNAME
      AND LANGU = 'E' AND SUBCLASS IN S_SUBCLS
      AND SETCLASS IN S_SETCLS.

    PID = NID.
    NID = NID + 1.
    APPEND VFL_HIER TO VIT_HIER.
    CLEAR: VFL_SETLEAF, VFL_HIER.
  ENDAT.

  VFL_HIER-NID = NID.
  VFL_HIER-PID = PID.
  VFL_HIER-IOBJ = P_IOBJ.
  VFL_HIER-DESC = VFL_SETLEAF-LTEXT.
  IF C_LZ = 'X'.
    SHIFT VFL_SETLEAF-VALFROM LEFT DELETING LEADING '0'.
  ENDIF.
  VFL_HIER-NNAME = VFL_SETLEAF-VALFROM.
  NID = NID + 1.
  APPEND VFL_HIER TO VIT_HIER.
  CLEAR: VFL_SETLEAF, VFL_HIER.
ENDLOOP.

LOOP AT VIT_HIER INTO VFL_HIER.
CONCATENATE VFL_HIER-NID ',' VFL_HIER-IOBJ ',' ' ' VFL_HIER-NNAME ' '
  ',' ' ' VFL_HIER-PID ',' 'E' ',' ' ',' ',' ',' ',' ',' '
  VFL_HIER-DESC ' ' INTO OUT.
VFL_FINAL-OUT = OUT.
APPEND VFL_FINAL TO VIT_FINAL.
CLEAR VFL_FINAL.
CLEAR: VFL_HIER, OUT.
ENDLOOP.

```

```

LOOP AT VIT_SETHEADER INTO VFL_SETHEADER.
  SELECT SINGLE SETNAME INTO VFL_SETHEADER-SETNAME FROM SETNODE
    WHERE SETNAME = VFL_SETHEADER-SETNAME
    AND SUBCLASS IN S_SUBCLS AND SETCLASS IN S_SETCLS.
  IF SY-SUBRC = 0.
    APPEND VFL_SETHEADER TO VIT_SETHEADER1.
  ENDIF.
  CLEAR VFL_SETHEADER.
ENDLOOP.

CLEAR VFL_HIER.
LOOP AT VIT_SETHEADER1 INTO VFL_SETHEADER.
  PID = 00000000.
  VFL_HIER-NID = NID.
  VFL_HIER-IOBJ = 'OHIER NODE'.
  VFL_HIER-NNAME = VFL_SETHEADER-SETNAME.
  SELECT SINGLE DESCRIPT FROM SETHEADERT INTO VFL_HIER-DESC
    WHERE SETNAME = VFL_SETHEADER-SETNAME
    AND LANGU = 'E' AND SUBCLASS IN S_SUBCLS
    AND SETCLASS IN S_SETCLS.

  NID = NID + 1.
  APPEND VFL_HIER TO VIT_HIER1.
  CLEAR: VFL_SETHEADER, VFL_HIER.
ENDLOOP.
SORT VIT_HIER1 BY NNAME.
SORT VIT_HIER BY NNAME.
LOOP AT VIT_HIER1 INTO VFL_HIER.
  READ TABLE VIT_HIER INTO VFL_HIER1 WITH KEY NNAME = VFL_HIER-NNAME
    IOBJ = 'OHIER_NODE' BINARY SEARCH.

  IF SY-SUBRC = 0.
    L_LINK = 'X'.
  ELSEIF L_SETNAME = VFL_HIER-NNAME AND VFL_HIER-IOBJ NE P_IOBJ.
    L_LINK = 'X'.
  ENDIF.
  CONCATENATE VFL_HIER-NID ',' VFL_HIER-IOBJ ',' ' ' VFL_HIER-NNAME ' '
    ',' L_LINK ',' VFL_HIER-PID ',' 'E' ',' ',' ',' ',' ' '
    VFL_HIER-DESC ' ' INTO OUT.
  VFL_FINAL-OUT = OUT.
  APPEND VFL_FINAL TO VIT_FINAL.
  CLEAR VFL_FINAL.
  L_SETNAME = VFL_HIER-NNAME.
  CLEAR: VFL_HIER, OUT, L_LINK.
ENDLOOP.

LOOP AT VIT_SETNODE INTO VFL_SETNODE.
READ TABLE VIT_HIER1 INTO VFL_HIER WITH KEY NNAME = VFL_SETNODE-SETNAME
  BINARY SEARCH.

  IF SY-SUBRC = 0.
    PID = VFL_HIER-NID.
    VFL_HIER1-NID = NID.
    VFL_HIER1-PID = PID.
    VFL_HIER1-IOBJ = 'OHIER_NODE'.
    VFL_HIER1-NNAME = VFL_SETNODE-SUBSETNAME.
    SELECT SINGLE DESCRIPT FROM SETHEADERT INTO VFL_HIER1-DESC
      WHERE SETNAME = VFL_SETNODE-SUBSETNAME
      AND LANGU = 'E' AND SUBCLASS IN S_SUBCLS
      AND SETCLASS IN S_SETCLS.

```

```

      NID = NID + 1.
      APPEND VFL_HIER1 TO VIT_HIER2.
      CLEAR: VFL_SETHEADER, VFL_HIER, VFL_HIER1.
    ENDIF.
  ENDLOOP.
  SORT VIT_HIER2 BY NNAME.
  LOOP AT VIT_HIER2 INTO VFL_HIER.
    READ TABLE VIT_HIER INTO VFL_HIER1 WITH KEY NNAME = VFL_HIER-NNAME
      IOBJ = 'OHIER_NODE' BINARY SEARCH.

    IF SY-SUBRC = 0.
      L_LINK = 'X'.
    ELSE.
      READ TABLE VIT_HIER1 INTO VFL_HIER1 WITH KEY NNAME = VFL_HIER-NNAME
        IOBJ = 'OHIER_NODE' BINARY SEARCH.

      IF SY-SUBRC = 0.
        L_LINK = 'X'.
      ELSEIF L_SETNAME = VFL_HIER-NNAME AND VFL_HIER-IOBJ NE P_IOBJ.
        L_LINK = 'X'.
      ENDIF.
    ENDIF.

  CONCATENATE VFL_HIER-NID ',' VFL_HIER-IOBJ ',' ' ' VFL_HIER-NNAME ' '
    ',' L_LINK ',' VFL_HIER-PID ',' 'E' ',' ' ' ',' ' '
    VFL_HIER-DESC ' ' INTO OUT.
  VFL_FINAL-OUT = OUT.
  APPEND VFL_FINAL TO VIT_FINAL.
  CLEAR VFL_FINAL.
  L_SETNAME = VFL_HIER-NNAME.
  CLEAR: VFL_HIER, OUT, VFL_HIER1, L_LINK.
ENDLOOP.

```

The above code will generate the file, which is compatible to load in BW in Master Hierarchy Object.

How Hierarchy is shown at BW Reports:

Now we can Load this extracted Hierarchy in Cost Element Object and create a Hierarchy Node variable at query level for this hierarchy. In Selection Screen Hierarchy is available to select at different nodes.

- ▼ **TOTAL (Excludes Personnel & Assessments) | KG-ALLCOST**
 - ▶ OPERATING INCOME | K585AOPINC
 - ▶ DEPRECIATION | K585ADEPRE
 - ▶ OTHER OPERATING EXPENSES | K585AOPER
 - ▶ INTEREST PROFIT/LOSS | K585AINTPL
 - ▶ OH ALLOCATION INCOME | K585AEXINC
 - ▶ EXTRAORDINARY EXPENSES | K585AEXEXP
 - ▶ TAXES ON INCOME | K585ATAXIC
 - ▶ TAXES & ASSESSMENTS | K585ATAXES

- ▼ **TOTAL (Excludes Personnel & Assessments) | KG-ALLCOST**
 - ▼ OPERATING INCOME | K585AOPINC
 - Clearing Account for Income from Disposa | 0000631009
 - Profit on Disposal of Tangible Fixed Ass | 0000631020
 - Income from Reduction of General Allowan | 0000632020
 - Income from Release of Provision for Vac | 0000632508

Related Content

[Relation between SETNODE, SETLEAF and SETHEADER](#)

[Structure of a Flat Hierarchy File for Loading via IDoc](#)

[Hierarchy Nodes](#)

[Link Nodes](#)

For more information, visit the [Business Intelligence homepage](#).

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.