Generic DataSource on CDHDR and CDPOS Tables

 Applies to:
 SAP R/3, SAP ECC 6.0 and SAP BI NetWeaver 2004s. For more information, visit the EDW homepage.

 Summary
 This article describes the way, document change data from SAP ECC CDHDR and CDPOS tables can be staged for various functional areas through Generic DataSource. This data can be used for compliance and audit reports in SAP BI.

 Author: Harpal Sinh Gohil
 Company: Infosys Technologies Ltd.
 Created on: 22 September 2010

 Author Bio
 Harpal Sinh Gohil is currently working with Infosys Technologies Ltd. He is a Business Intelligence consultant with 4 years of experience in SAP BW and BI Space.
# Table of Contents

- Brief Functional Introduction; Why Required? ................................................................. 3
- CDHDR and CDPOS Tables in ECC: Where to Get Data From? ........................................... 3
- Extract through Function Module: How to Get Data to BI? .................................................. 5
- Related Content ......................................................................................................................... 8
- Disclaimer and Liability Notice ............................................................................................... 9
Brief Functional Introduction; Why Required?

In day-to-day operational reporting, there are business scenarios, where you need to track changes to documents posted in SAP ECC. Even for audit or compliance purpose, reports are required to track changes. Some of the business scenarios, where change tracking reports are required are as below.

1. Delivery Manager to track changes in production plan date. This will help him manage his delivery dates. Same report can be used to monitor changes in planned delivery dates to a customer.
2. Track Changes in Sales order or Invoice prices based on date
3. To find out the changes made in purchase requisition, and its effect on supply from Vendor.
4. To check changes in due date for Delivery in Account Receivables module. A reschedule report can be created, which shows the changes in due date and what is cost of extending due date for customer payment.
5. To track changes in Forecast or plan data submission for internal audit, and implementing compliance for planning procedure. And, there can be many other business scenarios, where you need to stage change documents from ECC and show it in formatted BI Report.

CDHDR and CDPOS Tables in ECC: Where to Get Data From?

CDHDR is a header table to store change documents, while CDPOS stores data at item level. Generally it is required to check property in setup for transactions and background programs to store changes in CDHDR and CDPOS tables. You can store changes for FI and Sales Order module, but you can skip storing changes for purchase requisition. The structure of both the tables is small, since they store huge number of entries. Every change is assigned a change ID number, which will be a key. It also has special identifier to determine, which module is referred in change document.

CDHDR is transparent table, but CDPOS is a cluster table. So, you cannot join CDHDR and CDPOS table directly. Generally, old change documents from these tables are always archived for better performance of inserting new data. And, it is not preferable in ECC to create a report based on these tables, as it take long times for execution.

The snapshots below shows structure of both the tables. Common fields in both tables are OBJECTCLAS, OBJECTID and CHANGENR.

OBJECTCLAS field stores overall class under which the change falls. For e.g. for accounting document change, the class is ‘BELEG’, Sales order is ‘VERKBELEG’, purchase order is ‘EINKBELEG’. TCODE in CDHDR table refers to transaction, through which user has made changes. Similarly, TABNAME and FNAME field in CDPOS gives table name and field name which got changed for the change document.

OBJECTID or TABKEY is an important field which contains changed Document Number concatenated with other fields like item number, client etc.

For e.g. in case of change in accounting document number, the OBJECT ID will be comprised of corresponding Accounting Document Number, Company Code, Fiscal Year and item number.
### Generic DataSource on CDHDR and CDPOS Tables

#### Cluster Table
- **Field:** CDHDR
- **Key:** Activate
- **Short Description:** Change document items

#### Attributes
- **Delivery and Maintenance**
- **Fields**
- **Entity help check**
- **Currency/Quantity Fields**

#### Table Details
- **Field:** MAINT
  - **Key Init:** CLINT
  - **Data Type:** CHAR
  - **Length:** 3
  - **Description:** Client
- **Field:** OBJECTCLAS
  - **Key Init:** COOJECT
  - **Data Type:** CHAR
  - **Length:** 15
  - **Description:** Object class
- **Field:** OBJECTID
  - **Key Init:** COOJECTY
  - **Data Type:** CHAR
  - **Length:** 90
  - **Description:** Object value
- **Field:** CHANGE
  - **Key Init:** COOCHANG
  - **Data Type:** CHAR
  - **Length:** 16
  - **Description:** Document change number
- **Field:** TSWNAME
  - **Key Init:** COOTSWNAME
  - **Data Type:** CHAR
  - **Length:** 30
  - **Description:** Table name
- **Field:** CHANGC
  - **Key Init:** CCOCHANGC
  - **Data Type:** CHAR
  - **Length:** 70
  - **Description:** Changed table record key
- **Field:** CHANGN
  - **Key Init:** COOCCHANGN
  - **Data Type:** CHAR
  - **Length:** 30
  - **Description:** Field name
- **Field:** CHANGD
  - **Key Init:** COOCHANGD
  - **Data Type:** CHAR
  - **Length:** 1
  - **Description:** Change type (U, I, E, D)
- **Field:** TEXT
  - **Key Init:** COOCTEXT
  - **Data Type:** CHAR
  - **Length:** 1
  - **Description:** Flag: V-Text change
- **Field:** UNIT_NEW
  - **Key Init:** COOCUNIT
  - **Data Type:** UNIT
  - **Length:** 3
  - **Description:** Change documents, unlinked
- **Field:** CURR_NEW
  - **Key Init:** COOCURRY
  - **Data Type:** CURR
  - **Length:** 5
  - **Description:** Change documents, referenced currency
- **Field:** VALUE_NEW
  - **Key Init:** COOCOLON
  - **Data Type:** CHAR
  - **Length:** 254
  - **Description:** New contents of changed field
- **Field:** VALUE_OLD
  - **Key Init:** COOCLOD
  - **Data Type:** CHAR
  - **Length:** 254
  - **Description:** Old contents of changed field

### CDPOS Table
- **Field:** MANT
  - **Key Init:** CLINT
  - **Data Type:** CHAR
  - **Length:** 3
  - **Description:** Client
- **Field:** OBJECTCLAS
  - **Key Init:** COOJECT
  - **Data Type:** CHAR
  - **Length:** 15
  - **Description:** Object class
- **Field:** OBJECTID
  - **Key Init:** COOJECTY
  - **Data Type:** CHAR
  - **Length:** 90
  - **Description:** Object value
- **Field:** CHANGER
  - **Key Init:** COOCHANGER
  - **Data Type:** CHAR
  - **Length:** 18
  - **Description:** Document change number
- **Field:** USERNAME
  - **Key Init:** COOUSERNAME
  - **Data Type:** CHAR
  - **Length:** 12
  - **Description:** User name of the person responsible for change document
- **Field:** DATE
  - **Key Init:** COODATE
  - **Data Type:** DATS
  - **Length:** 8
  - **Description:** Creation date of the change document
- **Field:** TIME
  - **Key Init:** COOTIME
  - **Data Type:** TOYS
  - **Length:** 6
  - **Description:** Time change
- **Field:** SOURCE
  - **Key Init:** COOSOURCE
  - **Data Type:** CHAR
  - **Length:** 30
  - **Description:** Transaction in which a change was made
- **Field:** PLANDNUM
  - **Key Init:** COOPLAN
  - **Data Type:** CHAR
  - **Length:** 12
  - **Description:** Planned change number
- **Field:** ACT_CHANGE
  - **Key Init:** COOACTCHANGE
  - **Data Type:** CHAR
  - **Length:** 18
  - **Description:** Change number of the document created by this change
- **Field:** PLANNED
  - **Key Init:** COOPLAN
  - **Data Type:** CHAR
  - **Length:** 1
  - **Description:** Flag that changes were generated from planned changes
- **Field:** CHANGND
  - **Key Init:** COOCHANGND
  - **Data Type:** CHAR
  - **Length:** 1
  - **Description:** Aplication object change type (U, I, E, D)
- **Field:** LANGU
  - **Key Init:** COOALANGU
  - **Data Type:** LANG
  - **Length:** 1
  - **Description:** Language key
- **Field:** VERSION
  - **Key Init:** COOVERSION
  - **Data Type:** CHAR
  - **Length:** 3
  - **Description:** Byte field
Extract through Function Module: How to Get Data to BI?

There is no standard DataSource available based on CDHDR and CDPOS tables. So, the only way change control data can be fetched from ECC system to BI is creating a generic DataSource.

But, CDPOS table is cluster table; so, it is not possible to create DataSource based on table view. The only option is to create function module based generic DataSource and join the table in function module code.

So, let’s get started with step-by-step process.

1. **Create an Extract Structure.**
   - As per the requirement, identify all the fields, which are required to be staged and extracted. Let us take a simple example of tracking change in Request Delivery date of sales Order document. Go to transaction SE11 and create a test structure as below.
     - VBAK-VBELN “Sales Document
     - VBAK-POSNR “Item number
     - VBAK-ZREQUDELOLD “Old Request Delivery Date
     - VBAK-ZREQUEDELNEW “New Request Delivery Date
     - CDHDR-UPDATE “Change Date

2. **Create a function module**
   - First, identify the field you need to track for change and the module by which it can be changed.
     - To continue with the example of tracking change in Requested delivery date for delivery document, a new function module can be created as below. The tables referred here is VBAK (Sales Order Header Data)
       - a) Copy and modify RSAX_BIW_GET_DATA_SIMPLE as per your data load requirements. Change name and assign it to your Function Group.
       - b) Go to the ‘Tables’ tab and associate E_T_DATA with your structure (which you created for extraction)
       - c) Define all the tables, datafields, internal table and work area as required.
         - TYPES:
         - DATA:
       - d) Then Select the range of date for which you want to track the change document as below.
         - RANGES: L_R_CDATE FOR ZTEST_EXTRACT-CDATE, “CDATE – CHANGE DATE
         - LOOP AT I_T_SELECT INTO L_S_SELECT WHERE FIELDNM = ‘CDATE’. “I_T_SELECT IS AN INTERNAL TABLE OF TYPE G_T_SELECT
         - MOVE-CORRESPONDING L_S_SELECT TO L_R_CDATE.”L_R_CDATE IS DEFINED TO CONTAIN RANGE OF DATES
         - APPEND L_R_CDATE.
         - ENDLOOP.
       - e) Select the entire change documents from table CDHDR with range of above date.
         - OPEN CURSOR WITH HOLD G_CURSOR FOR “CURSOR IS OPENED TO SELECT DATA TILL MAXIMUM DATAPACKAGE SIZE IS REACHED.
         - SELECT OBJECTCLAS OBJECTID CHANGENR UDATE UTIME FROM CDHDR “YOU CAN SELECT OTHER FIELDS AS REQUIRED
         - WHERE OBJECTCLAS EQ ‘VERKBELEG’ AND UDATE IN L_R_CDATE. “VERKBELEG IS OBJECTCLAS FOR SALES/DELIVERY DOCUMENTS
         - FETCH NEXT CURSOR G_CURSOR “NEED TO DEFINE G_CURSOR IN DECLARATION INTO TABLE I_CDHDR “INTERNAL TABLE TO STORE CDHDR DATA
PACKAGE SIZE L_MAXSIZE.

f) If you want to track latest change for a date only, then delete all the duplicate entries after descending sorting based on change time.

    SORT I_CDHDR BY UDATE UTIME.

    DELETE ADJACENT DULICATES FROM I_CDHDR WITH KEY UDATE.

g) Select data from CDPOS Table for all the change documents selected from CDHDR table

SELECT OBJECTCLAS OBJECTID CHANGENR TABNAME TABKEY FNAME CHNGIND VALUE_OLD VALUE_NEW FROM CDPOS INTO TABLE I_CDPOS "INTERNAL TABLE DEFINED TO STORE CDPOS DATA FOR ALL ENTRIES IN I_CDHDR WHERE OBJECTCLAS EQ C_OBJECTCLAS

AND OBJECTID EQ I_CDHDR-OBJECTID" Here we are comparing OBJECTCLAS, OBJECTID and Change Document Number field

AND changenr EQ i_cdhdr-changenr

AND tabname EQ “constant”

AND fname EQ “constant”

AND chngind EQ “U”.

h) Now segregate data from CDPOS internal table Tabkey and derive all the required fields to get to the base table fields and create a final structure to be exported.

For e.g. As we are tracking change for sales document requested delivery date,

    LOOP AT I_CDPOS INTO WA_CDPOS.

    WA_DATAPACKAGE-VBELN = WA_CDPOS-TABKEY+0(10).

    "Assign Sales Document Number

    WA_DATAPACKAGE-POSNR = WA_CDPOS-TABKEY+10(2).

    "Assign Sales Document Item

    WA_DATAPACKAGE-ZREQUDELOLD = WA_CDPOS-VALUE_OLD.

    WA_DATAPACKAGE-ZREQUDELNEW = WA_CDPOS-VALUE_NEW.

    "ZREQUDELNEW and ZREQUDELOLD are new fields created to store old and new request delivery date.

    READ TABLE I_CDHDR into WA_CDHDR WITH KEY OBJECTCLAS

    OBJECTID = WA_CDPOS-OBJECTCLAS WITH KEY CHANGENR = WA_CDPOS-CHANGENR

    OBJECTID = WA_CDPOS-OBJECTID

    WA_DATAPACKAGE-UDATE = WA_CDHDR-UDATE.

    "Assign Change Date to Data package

    MOVE WA_DATAPACKAGE to E_T_DATA.

    "E_T_DATA is standard internal table to export data from function module. This is also structure of DataSource

ENDLOOP
With this, our function module on CDHDR and CDPOS is ready. But, we need to take care of following important points.

**Important Points:**

a. It is good to define all the constants in declaration sections, so that program can be reused with minor changes.

b. Though cursor method is recommended for selection statement in function module, it must be used in all Generic DataSource function module where data package size can be more than 10000 records.

c. Since CDHDR and CDPOS tables are very huge tables, always select on key field of tables, i.e. OBJECTCLAS, OBJECTID and CHANGENR. Also, OBJECTCLAS should be clear before you start to build any function module on CHDDR/CDPOS tables.

d. As far as possible, keep selection/delta for one day only.

e. Select only the fields which are required in function module logic or to pass to DataSource structure.

*Code mentioned here is just suggestive pseudo code and should not be used as it is.*

3. **Create a Generic DataSource**

   It is always practical to create a DataSource which supports delta mechanism, when fetching data from CDHDR and CDPOS. There are two ways, delta can be implemented for this DataSource.

   a. Create a Generic Delta using fields UDATE CDHDR table. With daily load, all the documents changed for last day can be extracted.

      But, to implement this, an entry needs to be maintained in table which stores the last timestamp of data transfer and get data after that. Above function module also needs to be changed accordingly. More details about this approach can be accessed at [Using Timestamps in Generic Delta Extraction by Function Module](#).

   b. Create a pseudo delta by passing a date range to a DataSource while extraction.

      In above example of function module, we have used this approach. The advantage with this approach is that, you can have flexible data load as and when required, and there is no need of doing initial full load. You can divide the loads in date range.

Select any of the above approach, go to transaction RSO2 and create a new DataSource assigning extract structure and function module as created in step 1 and 2.

Just replicate the DataSource in BI system and create an infopackage to fetch data. Data can then be modeled as per the reporting requirement.
Related Content

Functional Module Based Generic DataSource Part I

Using Timestamps in Generic Delta Extraction by Function Module
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.