Step by Step Guide to Create a Generic Datasource Based on Infoset Query Populated Via External Program

Applies to:

SAP ECC 5.0 and above releases For more information, visit the Business Intelligence homepage.

Summary

This paper gives a detail understanding and use of Infoset Query using External Program for creation of a generic datasource. This document provides a step by step guide to create a generic datasource based on Infoset Query populated via External Program.

Author: Devesh Laad

Company: Deloitte Consulting

Created on: 23 September 2009

Author Bio

Devesh Laad is SAP Certified Netweaver 2004s BI Solution Consultant. He has more than 6 yrs experience in SAP BW and has worked on SAP BW 3.0b, 3.5 and 7.0 versions.

© 2009 SAP AG

Table of Contents

Introduction 3
Comparison - Infoset Query via External Program vs Function Module based Generic Extraction 3
Scenario 3
Step By Step Procedure 4
Related Content 14
Disclaimer and Liability Notice 15

© 2009 SAP AG

Introduction

A Generic datasource is created when no standard business content datasource is available that meets client's reporting requirements. There are three main methods for generic extraction, namely

- Table or View
- Function Module
- Infoset Query

This document is focused on creation of generic datasource using Infoset Query populated via External Program.

Comparison - Infoset Query via External Program vs Function Module based Generic Extraction

- Function Module based generic extraction is pretty much rigid in terms of ABAP coding in different sections. On the other hand, Infoset Query via External Program is much more flexible and easier to code.
- One needs to understand statements like OPEN CURSOR, CLOSE CURSOR, FETCH NEXT, RANGES etc to build a function module based generic extractor. Infoset Query via External program can be created using local structures, internal tables and key statements `*<QUERY_HEAD> and *<QUERY_BODY>`

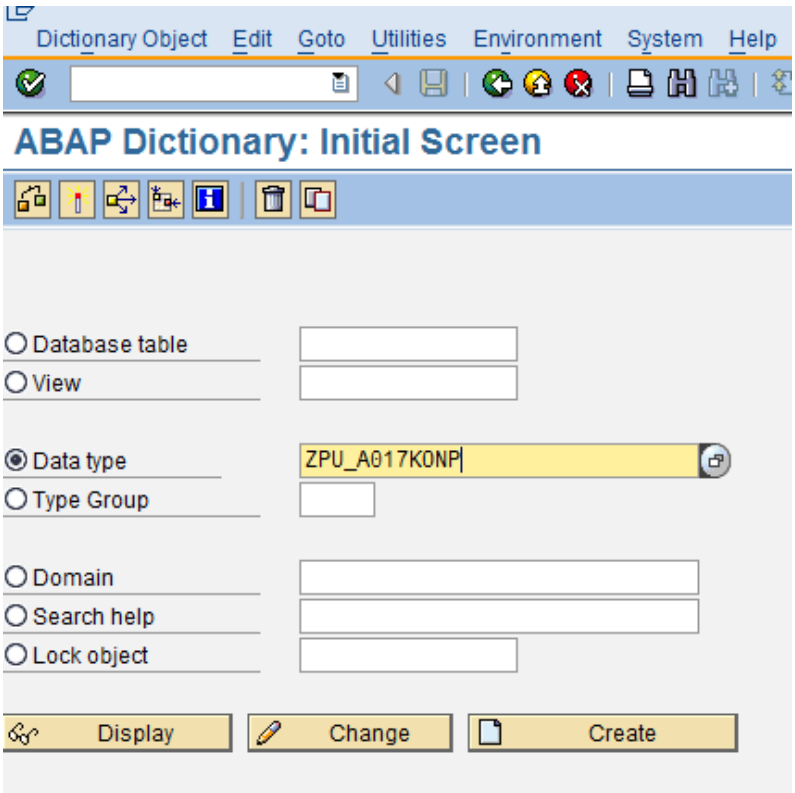
Scenario

To demonstrate steps for creation of generic datasource using Infoset Query via External Program, we would be combining data from tables A017 (Material Info Record (Plant-Specific)) and KONP (Conditions (Item)) based on field KNUMH (Condition Record Number).

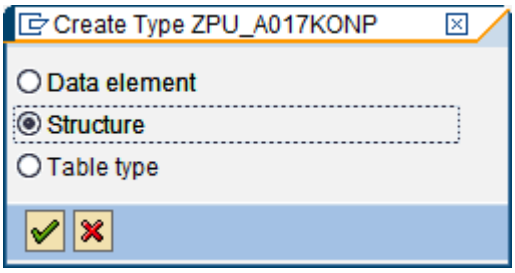
© 2009 SAP AG

Step By Step Procedure

1. Create a structure ZPU_A017KONP via SE11 transcation. Include all fields from table A017. Include fields KBETR KONWA KPEIN KMEIN from table KONP. Activate the structure.
   Goto SE11 transaction. Enter technical name of the structure ZPU_A017KONP in 'Data Type' field. Click Create.

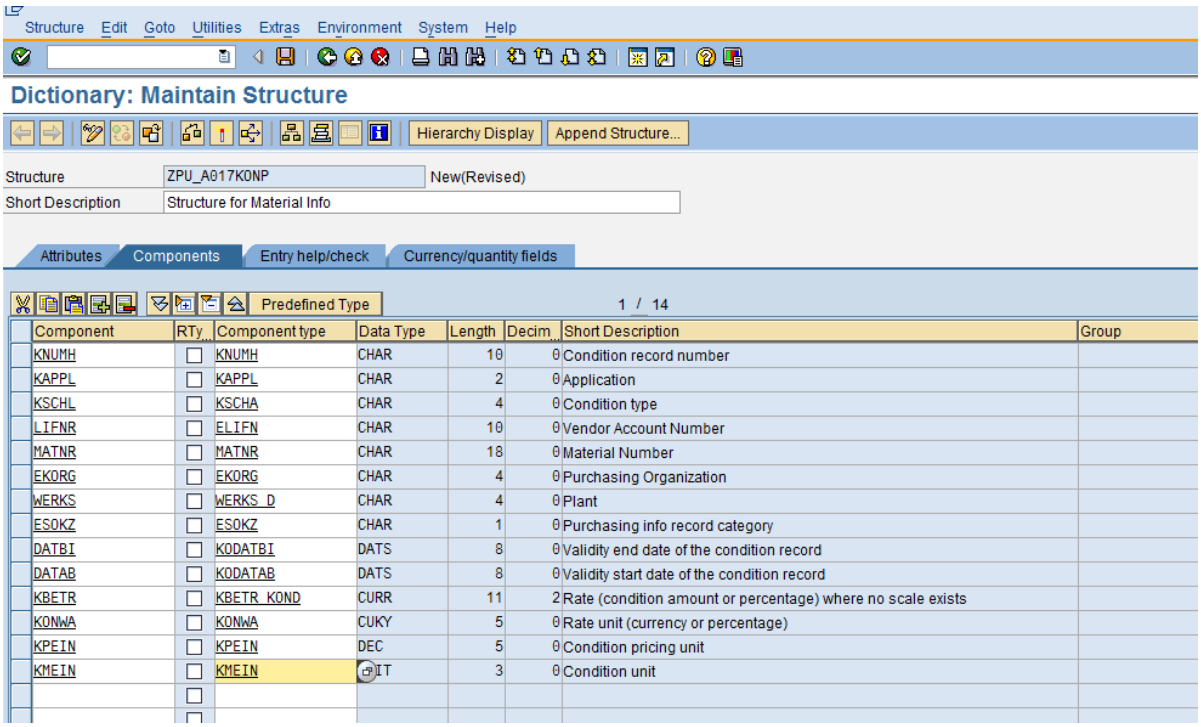


Select 'Structure' and enter.



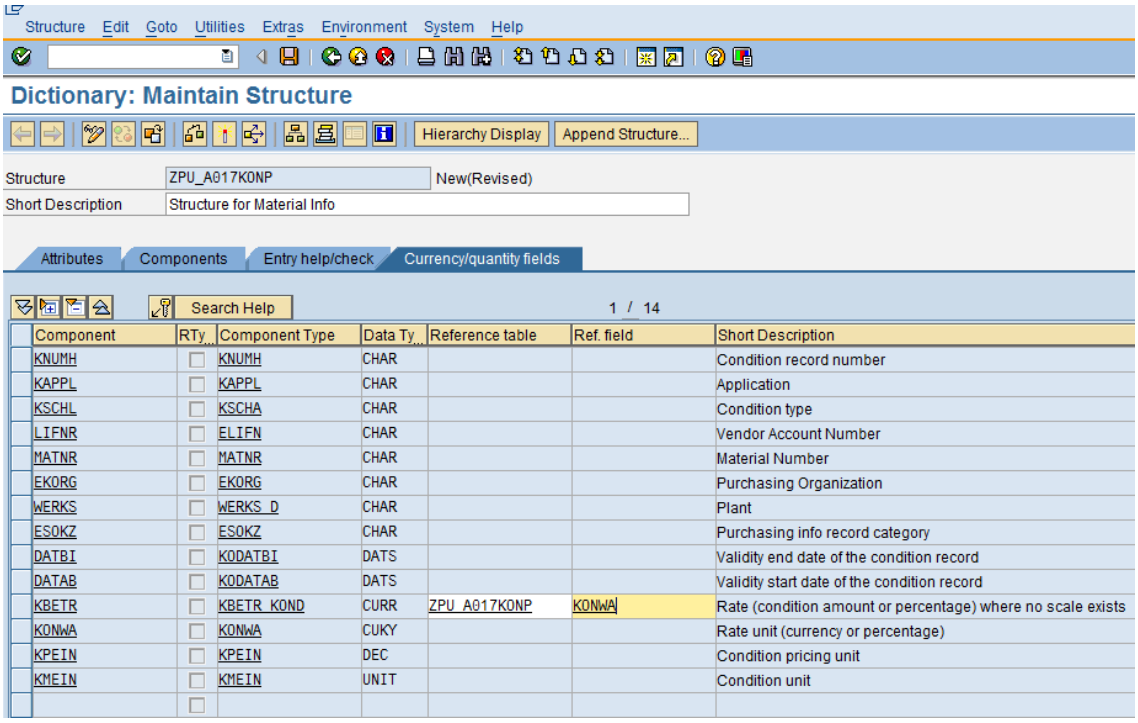
Enter description for structure. Include required fields from A017 and KONP table.

© 2009 SAP AG



Component	RTy	Component type	Data Type	Length	Decim	Short Description	Group
KNUMH		KNUMH	CHAR	10	0	Condition record number	
KAPPL		KAPPL	CHAR	2	0	Application	
KSCHL		KSCHA	CHAR	4	0	Condition type	
LIFNR		ELIFN	CHAR	10	0	Vendor Account Number	
MATNR		MATNR	CHAR	18	0	Material Number	
EKORG		EKORG	CHAR	4	0	Purchasing Organization	
WERKS		WERKS_D	CHAR	4	0	Plant	
ESOKZ		ESOKZ	CHAR	1	0	Purchasing info record category	
DATBI		KODATBI	DATS	8	0	Validity end date of the condition record	
DATAB		KODATAB	DATS	8	0	Validity start date of the condition record	
KBETR		KBETR_KOND	CURR	11	2	Rate (condition amount or percentage) where no scale exists	
KONWA		KONWA	CUKY	5	0	Rate unit (currency or percentage)	
KPEIN		KPEIN	DEC	5	0	Condition pricing unit	
KMEIN		KMEIN	IT	3	0	Condition unit	

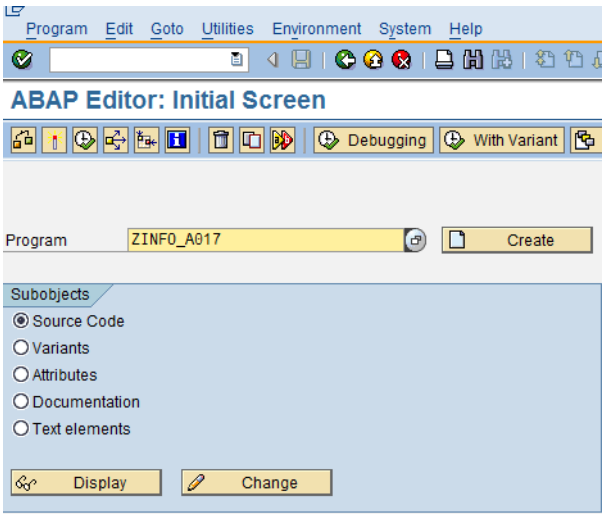
In ‘Currency/Quantity fields’ tab, enter the reference table ZPU_A017KONP (Structure name) and reference field KONWA for currency field KBETR. Save and Activate the structure. Ignore warnings (if any).



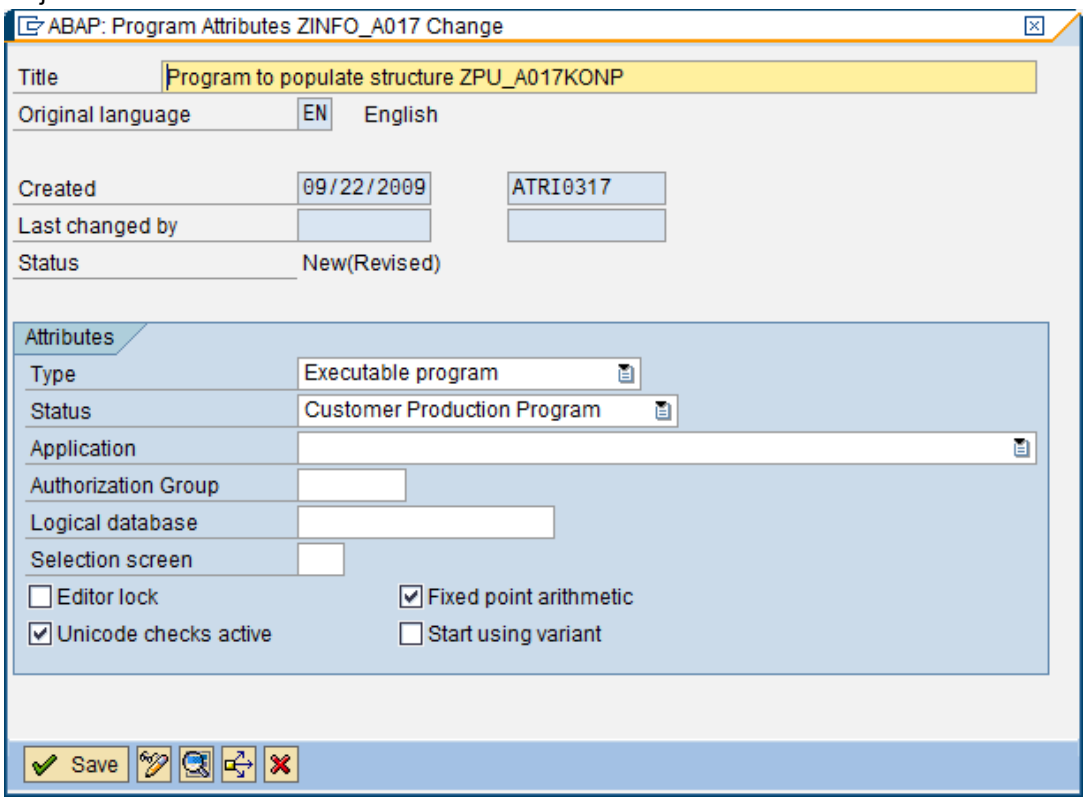
Component	RTy	Component Type	Data Ty	Reference table	Ref. field	Short Description
KNUMH		KNUMH	CHAR			Condition record number
KAPPL		KAPPL	CHAR			Application
KSCHL		KSCHA	CHAR			Condition type
LIFNR		ELIFN	CHAR			Vendor Account Number
MATNR		MATNR	CHAR			Material Number
EKORG		EKORG	CHAR			Purchasing Organization
WERKS		WERKS_D	CHAR			Plant
ESOKZ		ESOKZ	CHAR			Purchasing info record category
DATBI		KODATBI	DATS			Validity end date of the condition record
DATAB		KODATAB	DATS			Validity start date of the condition record
KBETR		KBETR_KOND	CURR	ZPU_A017KONP	KONWA	Rate (condition amount or percentage) where no scale exists
KONWA		KONWA	CUKY			Rate unit (currency or percentage)
KPEIN		KPEIN	DEC			Condition pricing unit
KMEIN		KMEIN	UNIT			Condition unit

2. Create external program ZINFO_A017 to populate structure ZPU_A017KONP via SE38 transaction. Goto SE38 transaction. Create program ZINFO_A017. Click ‘Create’.

© 2009 SAP AG



Enter required fields as per screen below and press enter. Select appropriate package or 'Local Object'.



© 2009 SAP AG

Type the following custom code to populate the structure ZPU_A017KONP. Activate the program.

data:

ZPU_A017KONP type ZPU_A017KONP,

it_data type standard table of ZPU_A017KONP.

```
field-symbols: <struc> type ZPU_A017KONP.
        TYPES : Begin of fs1,
        KAPPL type A017-KAPPL,
        KSCHL type A017-KSCHL,
        LIFNR type A017-LIFNR,
        MATNR type A017-MATNR,
        EKORG type A017-EKORG,
        WERKS type A017-WERKS,
        ESOKZ type A017-ESOKZ,
        DATBI type A017-DATBI,
        DATAB type A017-DATAB,
        KNUMH TYPE A017-KNUMH,
        end of fs1.
types : begin of fs2,
        KBETR TYPE KONP-KBETR,
        KONWA TYPE KONP-KONWA,
*        PEINH TYPE KONP-PEINH,
        KNUMH TYPE A017-KNUMH,
        KPEIN TYPE KONP-KPEIN,
        KMEIN TYPE KONP-KMEIN,
        end of fs2.
types : begin of fs3,
        ESOKZ type A017-ESOKZ,
        MATNR type A017-MATNR,
        WERKS type A017-WERKS,
        EKORG type A017-EKORG,
        LIFNR type A017-LIFNR,
        KSCHL type A017-KSCHL,
        DATBI type A017-DATBI,
        DATAB type A017-DATAB,
        KNUMH TYPE A017-KNUMH,
        KBETR TYPE KONP-KBETR,
        KONWA TYPE KONP-KONWA,
        KPEIN TYPE KONP-KPEIN,
        KMEIN TYPE KONP-KMEIN,
        end of fs3.
DATA : it1 type table of fs1,
       wa1 like line of it1,
       it2 type table of fs2,
       wa2 like line of it2,
       it3 type table of fs3,
       wa3 like line of it_data.
*---------------------------------------------------------------------*
*   selection screen statements
*---------------------------------------------------------------------*
*   (define your selection-screen here)
* !! the following comment MUST NOT BE CHANGED !!
*<QUERY_HEAD>
*---------------------------------------------------------------------*
*   read data into IT_DATA
```


© 2009 SAP AG

```
*-----------------------------------------------------------------*
*  (select your data here into internal table IT_DATA)
*-------------------------------------------------------------*
*   output of the data
*   (this section can be left unchanged)
*-------------------------------------------------------------*
select kappl kschl lifnr matnr ekorg werks esokz datbi datab
knumh
from A017 into table it1 where kappl = 'M'.
select KBETR KONWA KNUMH KPEIN KMEIN from KONP into table it2 for all
entries in it1
where knumh = it1-knumh.
loop at it1 into wa1.
clear : wa3, wa2.
move-corresponding wa1 to wa3.
read table it2 into wa2 with key knumh = wa1-knumh.
 if sy-subrc eq 0.
 move-corresponding wa2 to wa3.
 endif.
append wa3 to it_data.
endloop.

loop at it_data assigning <struc>.
  move-corresponding <struc> to ZPU_A017KONP.
* !! the following comment MUST NOT BE CHANGED !!
*<QUERY_BODY>
endloop.
```

Important Points while writing custom program for Infoset Query.

a) *<QUERY_HEAD> and *<QUERY_BODY> are key statements (even though they are commented).
b) Make all necessary declarations of structures, work areas, internal tables, field symbols before statement *<QUERY_HEAD>.
c) Do all necessary selects and logic for populating internal table between statements *<QUERY_HEAD> and *<QUERY_BODY>.
d) The last statement after *<QUERY_BODY> should be endloop (for populating the final structure).
e) Create a workarea with the same name as final structure (ZPU_A017KONP in this case).

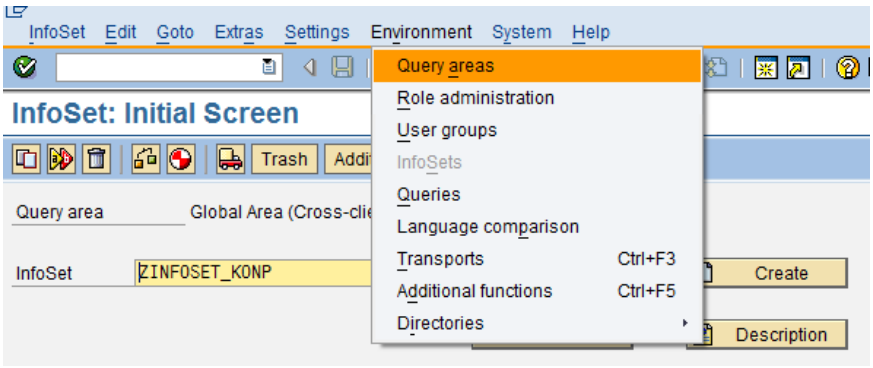
The structure of this model report and the sequence of its parts are regular. The main structure of a model report is given below.

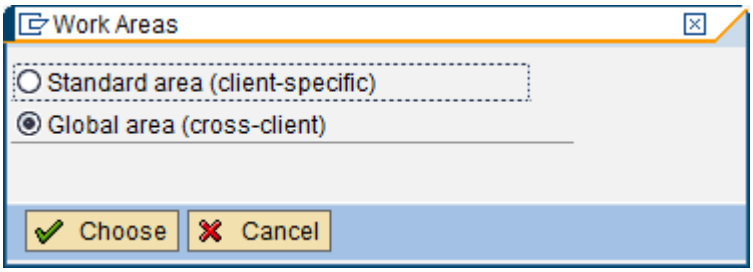
© 2009 SAP AG

Report xxxxxxx.	
Tables tab.	Definition of Dictionary structure used to set up the InfoSet. This structure must contain the records you want to evaluate.
DATA :…	
DATA: BEGIN OF itab OCCURS xxx. INCLUDE STRUCTURE tab. DATA: END of itab.	Definition of an internal table <itab> with structure <tab> which provides the records you want to evaluate.
* <Query_head>	This comment must always appear after your data declarations.
* Code to define the table itab, if such a table is used. * Beginning of a loop to retrieve each record and place it in the structure tab (SELECT, DO, LOOP, ...) * Code for formatting data (if necessary)	
* <Query_body>	This comment must always be the last 'statement' in the loop. The data must be available in structure tab.
* End of data retrieval loop for individual records (ENDSELECT, ENDDO, ENDLOOP;...)	

3. Create Infoset ZINFOSET_KONP in Global Area using SQ02 transaction using structure ZPU_A017KONP and program ZINFO_A017.

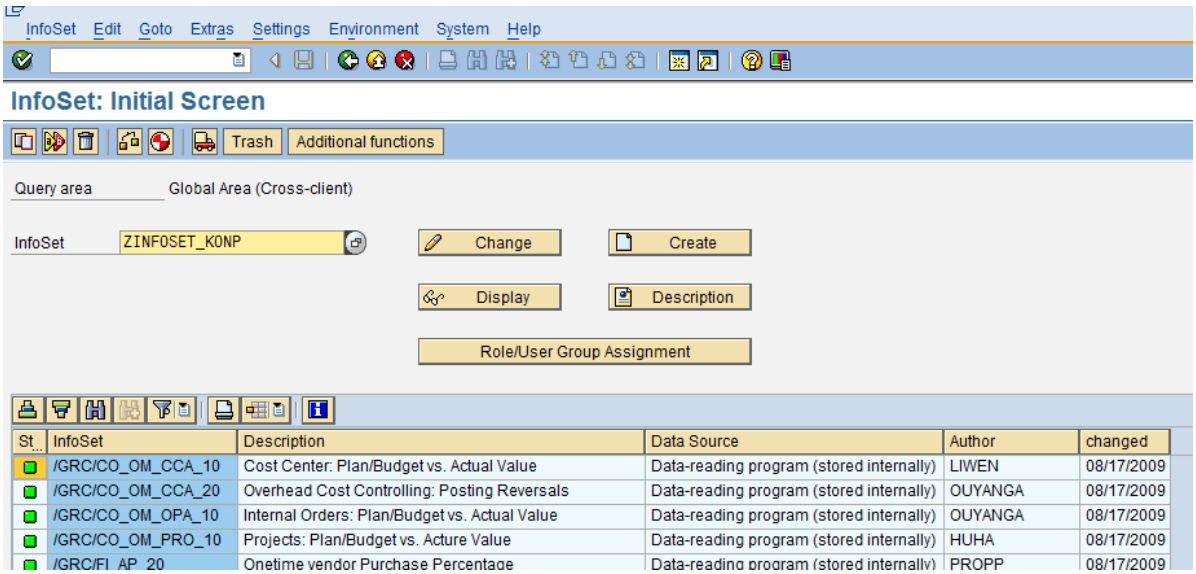
   Goto transaction SQ02. Make sure you are creating Infoset in 'Global Area'.



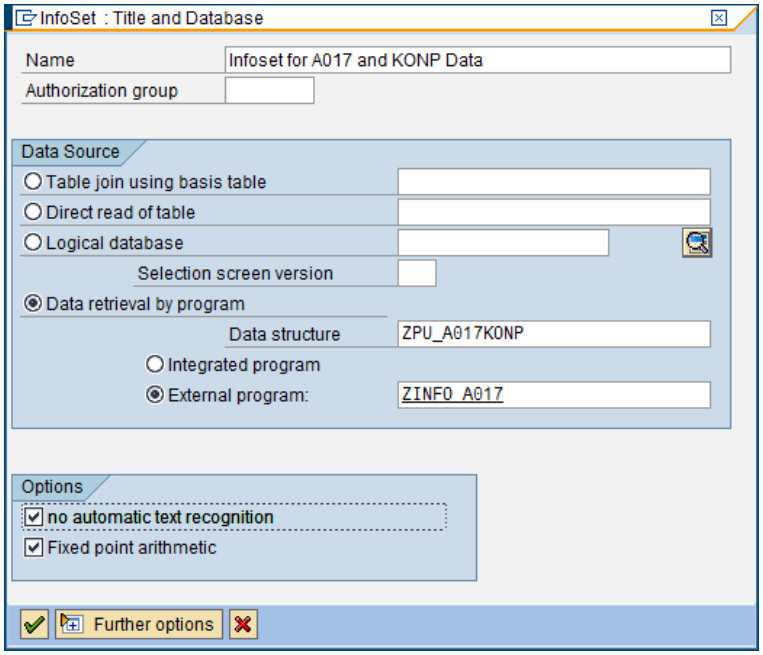


   Enter technical name of Infoset as ZINFOSET_KONP. Click Create.

© 2009 SAP AG

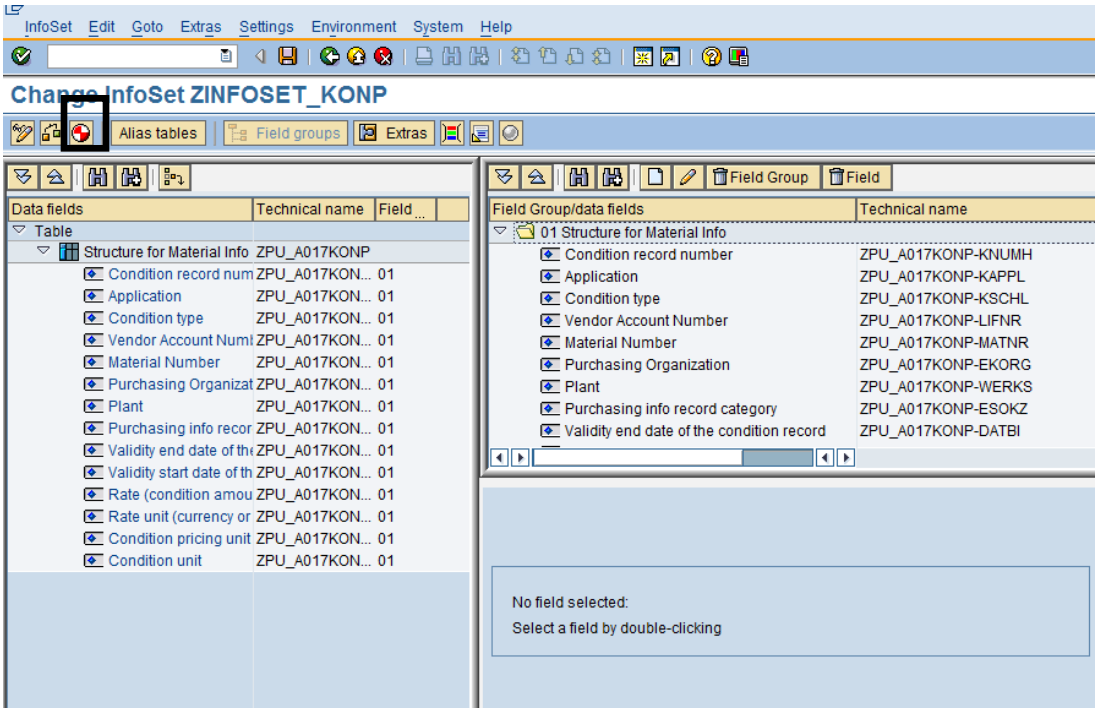


Enter description for Infoset. Select option 'Data retrieval by program'. Enter structure for Infoset (ZPU_A017KONP) and External Program (ZINFO_A017). Check 'No automatic text recognition' (to ignore text fields). Press enter.



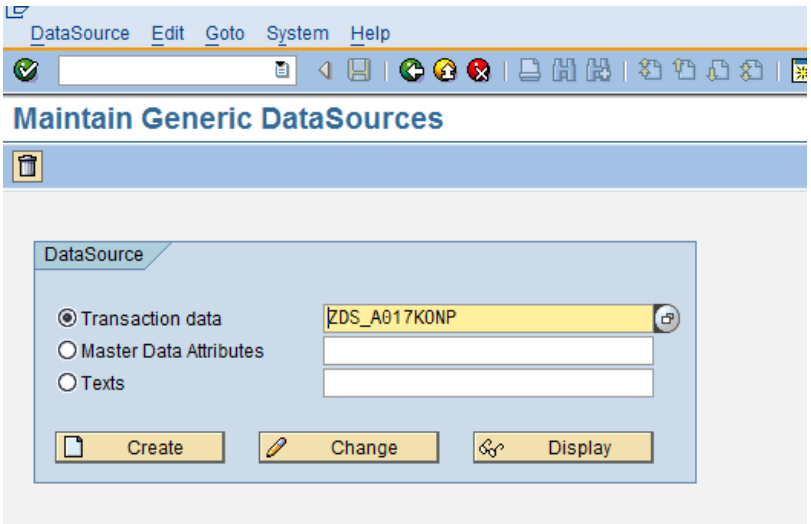
© 2009 SAP AG

4. Save and Generate Infoset.



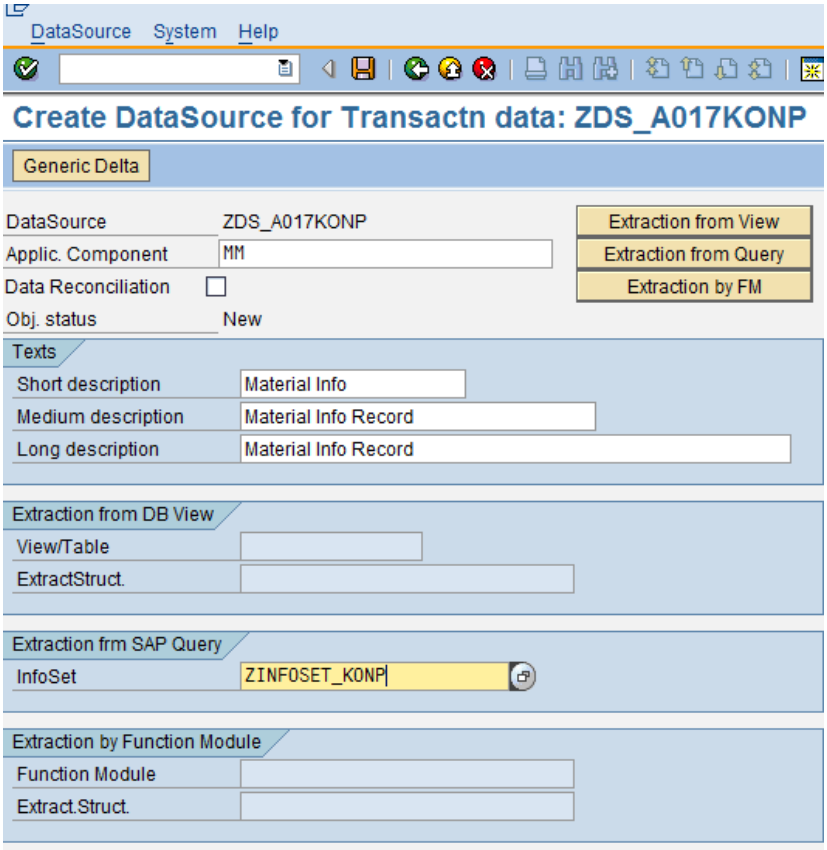
5. Create generic datasource ZDS_A017KONP using RSO2 transaction.

   Goto transaction RSO2. Enter technical name of datasource as ZDS_A017KONP. Click Create.

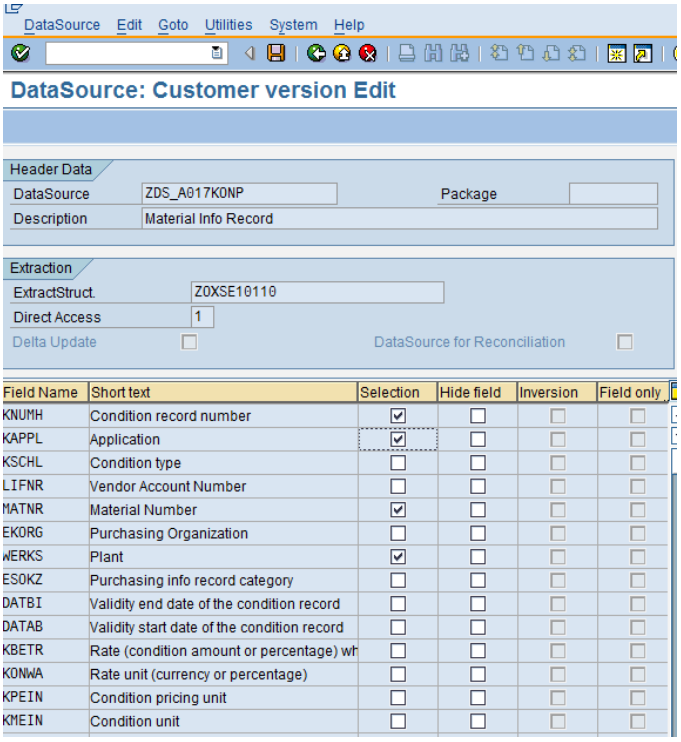


   Select 'Extraction from Query' and fill the required description. Click Save.

© 2009 SAP AG

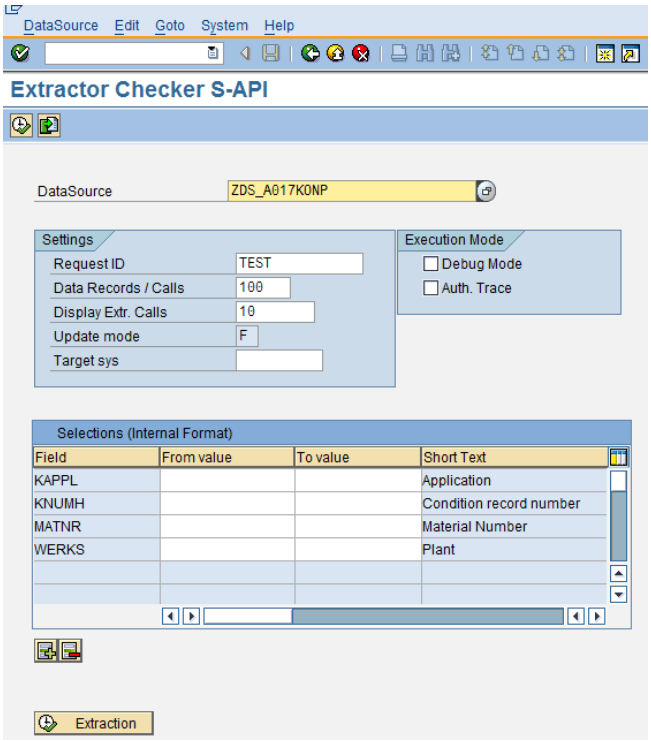


Select 'Selection Fields' and Save.

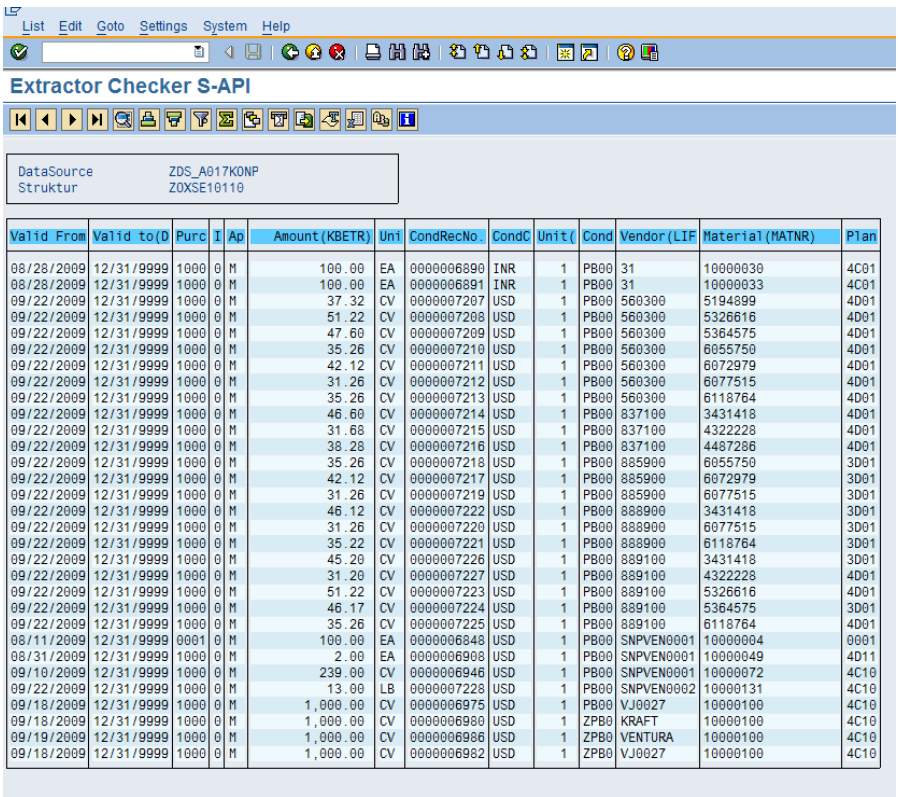


© 2009 SAP AG

6. Test datasource extraction via RSA3 transaction.



Verify the results.



Valid From	Valid to(D	Purc	I	Ap	Amount(KBETR)	Uni	CondRecNo.	CondC	Unit(	Cond	Vendor(LIF	Material(MATNR)	Plan
08/28/2009	12/31/9999	1000	0	M	100.00	EA	0000006890	INR	1	PB00	31	10000030	4C01
08/28/2009	12/31/9999	1000	0	M	100.00	EA	0000006891	INR	1	PB00	31	10000033	4C01
09/22/2009	12/31/9999	1000	0	M	37.32	CV	0000007207	USD	1	PB00	560300	5194899	4D01
09/22/2009	12/31/9999	1000	0	M	51.22	CV	0000007208	USD	1	PB00	560300	5326616	4D01
09/22/2009	12/31/9999	1000	0	M	47.60	CV	0000007209	USD	1	PB00	560300	5364575	4D01
09/22/2009	12/31/9999	1000	0	M	35.26	CV	0000007210	USD	1	PB00	560300	6055750	4D01
09/22/2009	12/31/9999	1000	0	M	42.12	CV	0000007211	USD	1	PB00	560300	6072979	4D01
09/22/2009	12/31/9999	1000	0	M	31.26	CV	0000007212	USD	1	PB00	560300	6077515	4D01
09/22/2009	12/31/9999	1000	0	M	35.26	CV	0000007213	USD	1	PB00	560300	6118764	4D01
09/22/2009	12/31/9999	1000	0	M	46.60	CV	0000007214	USD	1	PB00	837100	3431418	4D01
09/22/2009	12/31/9999	1000	0	M	31.68	CV	0000007215	USD	1	PB00	837100	4322228	4D01
09/22/2009	12/31/9999	1000	0	M	38.28	CV	0000007216	USD	1	PB00	837100	4487286	4D01
09/22/2009	12/31/9999	1000	0	M	35.26	CV	0000007218	USD	1	PB00	885900	6055750	3D01
09/22/2009	12/31/9999	1000	0	M	42.12	CV	0000007217	USD	1	PB00	885900	6072979	3D01
09/22/2009	12/31/9999	1000	0	M	31.26	CV	0000007219	USD	1	PB00	885900	6077515	3D01
09/22/2009	12/31/9999	1000	0	M	46.12	CV	0000007222	USD	1	PB00	888900	3431418	3D01
09/22/2009	12/31/9999	1000	0	M	31.26	CV	0000007220	USD	1	PB00	888900	6077515	3D01
09/22/2009	12/31/9999	1000	0	M	35.22	CV	0000007221	USD	1	PB00	888900	6118764	3D01
09/22/2009	12/31/9999	1000	0	M	45.20	CV	0000007226	USD	1	PB00	889100	3431418	3D01
09/22/2009	12/31/9999	1000	0	M	31.20	CV	0000007227	USD	1	PB00	889100	4322228	4D01
09/22/2009	12/31/9999	1000	0	M	51.22	CV	0000007223	USD	1	PB00	889100	5326616	4D01
09/22/2009	12/31/9999	1000	0	M	46.17	CV	0000007224	USD	1	PB00	889100	5364575	3D01
09/22/2009	12/31/9999	1000	0	M	35.26	CV	0000007225	USD	1	PB00	889100	6118764	4D01
08/11/2009	12/31/9999	0001	0	M	100.00	EA	0000006848	USD	1	PB00	SNPVEN0001	10000004	0001
08/31/2009	12/31/9999	1000	0	M	2.00	EA	0000006908	USD	1	PB00	SNPVEN0001	10000049	4D11
09/10/2009	12/31/9999	1000	0	M	239.00	CV	0000006946	USD	1	PB00	SNPVEN0001	10000072	4C10
09/22/2009	12/31/9999	1000	0	M	13.00	LB	0000007228	USD	1	PB00	SNPVEN0002	10000131	4C10
09/18/2009	12/31/9999	1000	0	M	1,000.00	CV	0000006975	USD	1	PB00	VJ0027	10000100	4C10
09/18/2009	12/31/9999	1000	0	M	1,000.00	CV	0000006980	USD	1	ZPB0	KRAFT	10000100	4C10
09/19/2009	12/31/9999	1000	0	M	1,000.00	CV	0000006986	USD	1	ZPB0	VENTURA	10000100	4C10
09/18/2009	12/31/9999	1000	0	M	1,000.00	CV	0000006982	USD	1	ZPB0	VJ0027	10000100	4C10

© 2009 SAP AG

Related Content

http://help.sap.com/saphelp_nw70/helpdata/EN/d2/cb45b2455611d189710000e8322d00/content.htm

http://help.sap.com/saphelp_erp2005/helpdata/en/f6/c39138e4a0341fe10000009b38f8cf/content.htm

For more information, visit the Business Intelligence homepage.

© 2009 SAP AG

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.

© 2009 SAP AG