



How-to Guide
SAP NetWeaver 7.0 (2004s)

How To... **Create Generic** **DataSources** **which use the** **Delta Queue**

Version 1.00 – February 2006

Applicable Releases:
SAP NetWeaver 7.0
Business Information Management
Enterprise Data Warehousing

© Copyright 2006 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, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation 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, 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.

SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

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.

1 Business Scenario

You have been tasked to bring purchase requisitions into your data warehouse and you have not been able to identify a standard content dataSource which will accomplish this task for you.

Using the generic datasources available to you in SAP NetWeaver BI, is your preferred solution however you have not been able to identify a date field which accurately captures delta records.

You need to deliver a solution to appropriately capture deltas within your purchasing system for purchase requisitions.

2 Introduction

Within SAP R/3 and mySAP ERP systems there are Business Transaction Events or BTE's. BTE's are essentially events that take place at the time of a business transaction such as a change to a purchase order or for our particular scenario a purchase requisition. Keep in mind there are a great deal of different types of business transaction events in all areas of SAP R/3, each event can indeed enable a delta capture mechanism. Please consult transaction FINF to navigate through the existing BTE's in your SAP R/3 system and educate yourself further.

These events can be leveraged to capture delta changes and write these changes to the delta queue directly within transactional systems. BTE's act as a "delta hook", this enables the use of standard delta queue functionality within your data warehouse as long as your datasource is configured appropriately.

Certain delta processes within datasources support the use of the delta queue whereas others do not. In order to implement this solution a dataSource will have to be configured with a delta process such as ABR, or AIM that inherently support the use of the delta queue as a delta capture mechanism.

3 Step by Step Solution

The task of creating a dataSource which retrieves purchase requisition line item information from a transactional (SAP R/3) system moving this information into a data warehouse (SAP NetWeaver BI) has been assigned to us.

3.1 Find a BTE to use for delta capture

1. Navigate to transaction FINF and identify a business transaction event, BTE which captures changes to the table(s) needed.

The screenshot shows the SAP Business Transaction Events: Publish & Subscribe Interfaces search screen. It features several search criteria sections:

- BTE Search for Selection Attributes:** Includes fields for Attribute Type and Selection Attribute.
- BTE Interval Search:** Includes fields for Business Transaction Event (start and end) and checkboxes for "Only BTEs with Active Products" and "Display Short Texts Only".
- BTE Search for:** Includes fields for ISO code and Application Indicator.
- BTE Search for:** Includes fields for Partner and Product.
- BTE Search for:** Includes a field for Additional cust.prod.

2. Identify the BTE which captures the necessary information.

The screenshot shows the SAP Business Transaction Events: Publish & Subscribe Interfaces results screen. The "Sample function module" button is circled in red. Below the buttons, a table lists the selected BTEs:

P&S BTE	BTE Description
01000710	AP0 Plug-In: Purchase Requisition
01000720	AP0 Plug-In: Confirmation
01000730	AP0 Plug-In: Purchase Order
01000740	AP0 Plug-In: Goods Receipt for Purchase Order
01000811	AP0 Plug-In: Sales Document and Requirements
01000902	AP0 Plug-In: Campaign
02000104	Markdown Planning: Price Activation
0CML0001	CML Account Master Data: Acct Fields Rel. for BaFin
CACS0000	Reserved Events for CA-CS: CACS0000 to CACS9999
CACS1000	IS-IS-CS: User Exits Commission Case Dual Control
CACS1001	IS-IS-CS: Industry Key, Obj. Char. Grp, AttrCheck
CACS1002	IS-IS-CS: Authorization Check in Org. Hierarchy

3. Create a function module to take care of the delta write functionality being built into our DataSource.

A sample function module interface can be found on the screen we saw in the last step (it is circled in red). By clicking on the Sample Function Module button the Function Module Builder transaction will then start up where the Sample Function Module can be copied as can be seen here on the right.

Name and save this function module because it will be used again in step 3.3.

```

Function Builder: Change ZSC_BTE_DELTA_CAPTURE_01000710
Function module ZSC_BTE_DELTA_CAPTURE_01000710 Active
Source code
FUNCTION ZSC_BTE_DELTA_CAPTURE_01000710.
***-----
***"Local interface:
***  TABLES
***      T_XEBAN STRUCTURE EBKN OPTIONAL
***      T_XEBKN STRUCTURE EBKN OPTIONAL
***      T_YEBAN STRUCTURE EBKN OPTIONAL
***      T_YEBKN STRUCTURE EBKN OPTIONAL
***-----
ENDFUNCTION.

```



Note Within Business Transaction Events two types of internal tables are passed for each database table, an X table and a Y table. The Y table contains the original values of the records modified, whereas the X table contains the new records with the modified values.

3.2 Create a View

1. Navigate to transaction SE11 within the SAP R/3 system. Create a new view which is built on top of the EBKN table.

NOTE: Here a view is being created in order to be used within the Generic DataSource. If it is preferable to the user to create a FM or InfoSet for the Generic DataSource, this can be done as well.

Dictionary: Maintain View

Database view: ZEBKN New(Revised)

Short Description: Purchase Requisition

Field name	Short description
<input checked="" type="checkbox"/> MANDT	<input checked="" type="checkbox"/> Client
<input checked="" type="checkbox"/> BNFN	<input checked="" type="checkbox"/> Purchase requisition number
<input checked="" type="checkbox"/> BNFPO	<input checked="" type="checkbox"/> Item number of purchase requisition
<input checked="" type="checkbox"/> ZEBKN	<input checked="" type="checkbox"/> Serial number for PReq account assignment segment
<input checked="" type="checkbox"/> LOEKZ	<input type="checkbox"/> Deletion indicator in purchasing document
<input checked="" type="checkbox"/> ERDAT	<input type="checkbox"/> Date on which the record was created
<input checked="" type="checkbox"/> ERNAM	<input type="checkbox"/> Name of Person who Created the Object
<input checked="" type="checkbox"/> MENGE	<input type="checkbox"/> Purchase requisition quantity
<input checked="" type="checkbox"/> VPROZ	<input type="checkbox"/> Distribution percentage in the case of multiple acct
<input checked="" type="checkbox"/> SAKTO	<input type="checkbox"/> G/L Account Number
<input checked="" type="checkbox"/> GSBER	<input type="checkbox"/> Business Area
<input checked="" type="checkbox"/> KOSTL	<input type="checkbox"/> Cost Center
<input checked="" type="checkbox"/> PROJN	<input type="checkbox"/> Old: Project number : No longer used -> PS_POSNR
<input checked="" type="checkbox"/> VBELN	<input type="checkbox"/> Sales and Distribution Document Number
<input checked="" type="checkbox"/> VBELP	<input type="checkbox"/> Sales Document Item
<input checked="" type="checkbox"/> WETEN	<input type="checkbox"/> Schedule Line
<input checked="" type="checkbox"/> ANLN1	<input type="checkbox"/> Main Asset Number
<input checked="" type="checkbox"/> ANLN2	<input type="checkbox"/> Asset Subnumber
<input checked="" type="checkbox"/> AUFNR	<input type="checkbox"/> Order Number
<input checked="" type="checkbox"/> WEMPF	<input type="checkbox"/> Goods Recipient/Ship-To Party
<input checked="" type="checkbox"/> ABLAD	<input type="checkbox"/> Unloading Point
<input checked="" type="checkbox"/> KOKRS	<input type="checkbox"/> Controlling Area

- Next, run the view and make sure that all of the necessary data required is available from this view.

Data Browser: Table ZEBKN Select Entries 185

Table: ZEBKN
Displayed fields: 21 of 56 Fixed columns: List width 0250

MANDT	BANFN	BNFPO	ZEBKN	LOEKZ	ERDAT	ERNAH	MENGE	VPROZ	SAKTO	SSBER
400	0010000208	00010	01		00.00.0000		100,000	0,0	0000473000	1000
400	0010000384	00010	01		00.00.0000		25,000	0,0	0000415000	
400	0010000385	00010	01		00.00.0000		12,000	0,0	0000415000	
400	0010000418	00010	01		00.00.0000		12,000	0,0	0000415000	
400	0010000563	00010	01		00.00.0000		25,000	0,0	0000415000	
400	0010000799	00010	01		00.00.0000		50,000	0,0	0000410000	9900
400	0010000799	00020	01		00.00.0000		80,000	0,0	0000410000	9900
400	0010000799	00030	01		00.00.0000		100,000	0,0	0000410000	9900
400	0010000744	00010	01		00.00.0000		10,000	0,0	0000400000	9900
400	0010000776	00010	01		00.00.0000		20,000	0,0	0000890000	
400	0010000777	00010	01		00.00.0000		20,000	0,0	0000890000	
400	0010000778	00010	01		00.00.0000		20,000	0,0	0000890000	
400	0010000779	00010	01		00.00.0000		20,000	0,0	0000890000	
400	0010000808	00010	01		00.00.0000		20,000	0,0	0000890000	
400	0010000809	00010	01		00.00.0000		20,000	0,0	0000890000	
400	0010000815	00010	01		00.00.0000		100,000	0,0	0000890000	
400	0010000817	00010	01		00.00.0000		100,000	0,0	0000890000	
400	0010000821	00010	01		00.00.0000		10,000	0,0	0000410000	9900
400	0010000822	00010	01		00.00.0000		10,000	0,0	0000410000	9900
400	0010000876	00010	01		00.00.0000		1,000	0,0	0000890000	
400	0010000877	00010	01		00.00.0000		1,000	0,0	0000400010	

3.3 Create the datasource

- Navigate to transaction RSO2.

Fill in the appropriate textual information and the name of the database view just created. Hit the save button.

NOTE: The DataSource must begin with a 'Z_'.

Create DataSource für Transactn data: Z_BTE_DELTA

Generic Delta

DataSource: Z_BTE_DELTA
 Applic. Component: MM
 Obj. status: New

Extraction from View
 Extraction from Query
 Extraction by FM

Texts

Short description: Purchase Requisition
 Medium description: Purchase Requisition AA Table
 Long description: Purchase Requisition Account Assignment Table

Extraction from DB View

ViewTable: ZEBKN
 ExtractStruct:

Extraction from SAP Query

InfoSet:

Extraction by Function Module

Function Module:
 Extract.Struct.:

2. Choose the fields which will be used for selection criteria and then save the DataSource.

Field Name	Short text	Selection	Hide field	Inversion	Field only
BANFN	Purchase requisition number	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
BNFPO	Item number of purchase requisition	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ZEBKN	Serial number for PReq account assign...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LOEKZ	Deletion indicator in purchasing docume...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ERDAT	Date on which the record was created	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ERNAM	Name of Person who Created the Object	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
VPROZ	Distribution percentage in the case of m...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SAKTO	G/L Account Number	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
GSBER	Business Area	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
KOSTL	Cost Center	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3. Now, it is necessary to modify the delta process type of the newly created dataSource to one that uses the delta queue. To do this there is a utility program created (see appendix), Z_CHANGE_DELTA_PROCESS.
 - a. This program needs to be installed, to do this navigate to transaction se38 and type in the program name.
 - b. Hit the create button.
 - c. Copy and paste the code from the appendix into the ABAP Editor and activate it.

```

*&-----*
*& Report  Z_CHANGE_DELTA_PROCESS
*&-----*
*&
*&
*&-----*

report z_change_delta_process

*P_DATAS   DataSource
*P_DELTAP  Delta Process for DataSource
parameters:
  p_datas type roosource-oltpsource,
  p_deltap type roosource-delta.

tables:
  roosource.

data:
  ls_roosource type roosource.

select single * from roosource into ls_roosource
  where oltpsource = p_datas
  and objvers = 'A'.

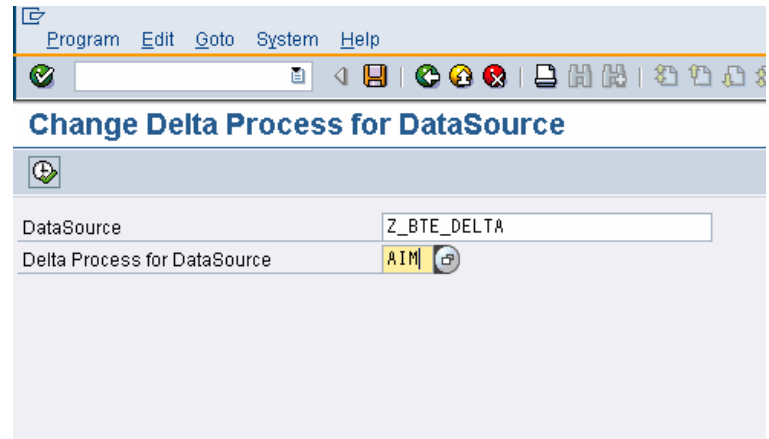
if sy-subrc eq 0.
  ls_roosource-delta = p_deltap.
  update roosource from ls_roosource.
  message 'The DataSource has been updated successfully.' type 'I'.
else.
  message 'The DataSource entered is not valid, try again.' type 'E'.
endif.

```

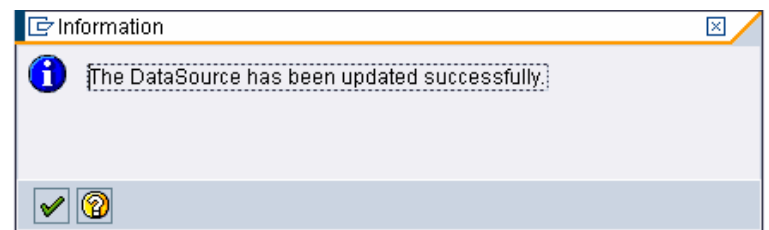
4. To modify the delta process type, run this program. You will see two parameters that you need to fill in to run the program.

- a. The DataSource name
- b. The Delta Process type

The Delta Process type you enter needs to support the delta queue. For this example we will use 'AIM', After Images using the Delta Queue.

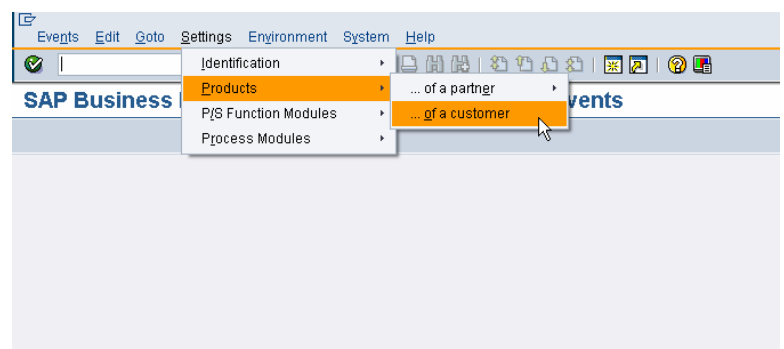


5. Once the necessary parameters have been entered execute the program. This will update the data source appropriately.



3.4 Create Product and Link BTE & Function Module

1. Navigate to transaction FIBF and find the menu path to create a new Product.

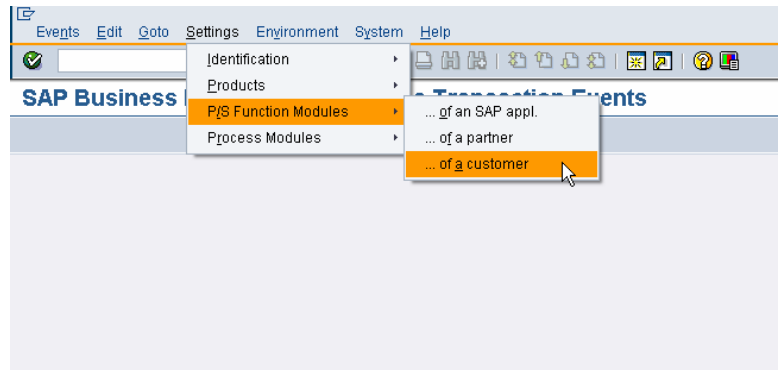


- From here a customer product is created, this is necessary in order to invoke a function module at the time the business transaction event occurs. It is a good idea to maintain some sort of naming convention for both the products created as well as their associated Function Modules.

Product	Text	RFC destination	Active
OPTARCH			<input checked="" type="checkbox"/>
Z_DELTA	Delta Hook		<input checked="" type="checkbox"/>

- A link needs to be created between our BTE, the Function Module created and the Product, Z_DELTA created in the last step.

To do this find the P/S Function Module of a customer option using the following path Settings->P/S Function Modules-> ... of a customer.

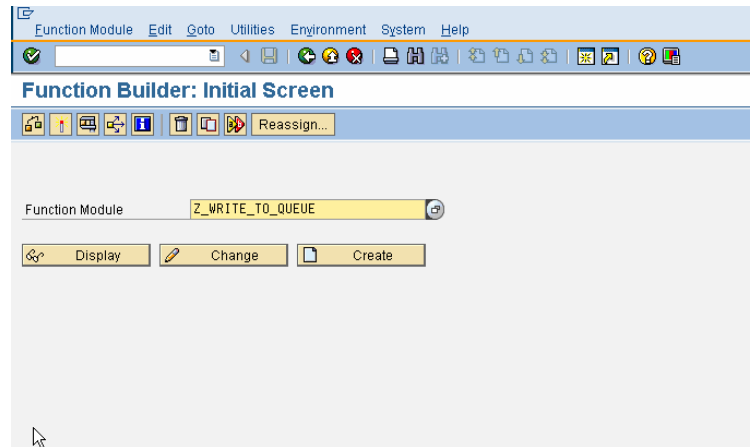


- Create a new entry specifying the BTE number, the product created as well as the function module shell we created in step 3.1. Save this setting.

Event	Product	Ctrl	Appl.	Function Module
01000710	Z_DELTA			ZSC_BTE_DELTA_CAPTURE_01000710

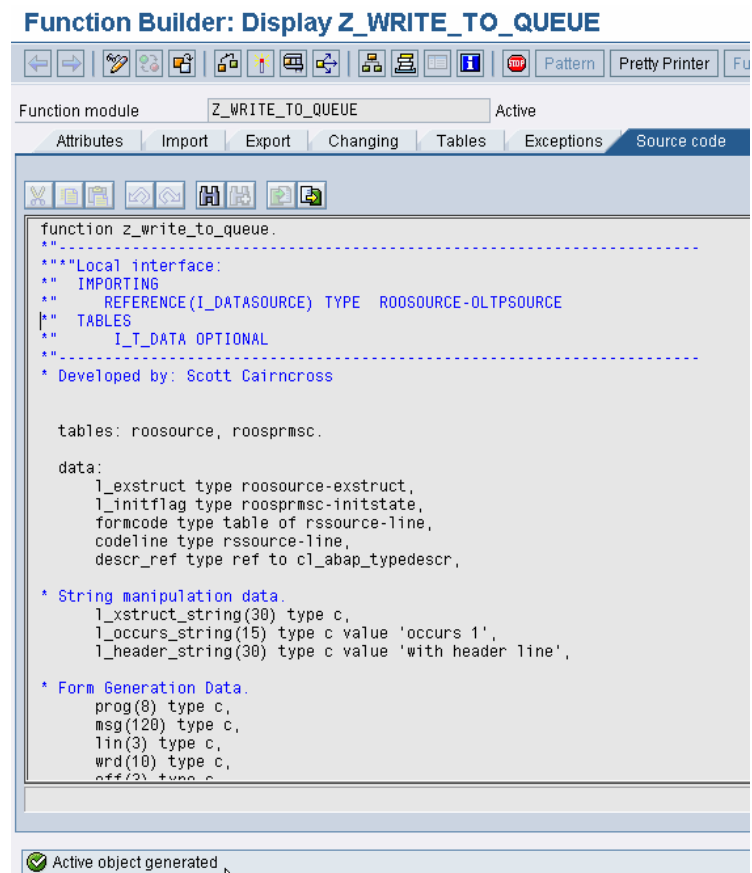
3.5 Finish Delta Hook

1. Navigate to transaction se37 create a new Function module and give it the name Z_WRITE_TO_QUEUE.



2. Copy and paste the code from the Appendix 4.2 into the Function Module Editor, then save and activate the Function Module.

NOTE: This function module will be used later to write to the delta queue. The standard extraction user exit is also called within this Function Module; all enhancements created via the standard mechanism are supported.



3. Now navigate to the other Function Module we created in step 3.1 and fill it with the necessary code to write to the delta queue.
 - a. Navigate to transaction se37
 - b. Place the name of the function module in the input field and hit change.
 - c. Insert the code from the Appendix 4.3.
 - d. Insert the name of your dataSource created
 - i. `l_datasource = 'Z_BTE_DELTA'`.
 - e. The last step is to specify the **X** table passed from the BTE interface within the call to the Function Module `Z_WRITE_TO_QUEUE`, in our example we want to capture Purchase Requisition Account Assignment changes.

Function Builder: Change ZSC_BTE_DELTA_CAPTURE_01000710

```

function zsc_bte_delta_capture_01000710.
***Local interface:
** TABLES
**   T_XEBAN STRUCTURE EBAN OPTIONAL
**   T_XEBKN STRUCTURE EBKN OPTIONAL
**   T_YEBAN STRUCTURE EBAN OPTIONAL
**   T_YEBKN STRUCTURE EBKN OPTIONAL
***
data:
  l_datasource type roosource-oltpsource.

  l_datasource = 'Z_BTE_DELTA'
  if lines( t_xebkn[] ) > 0.

    call function 'Z_WRITE_TO_QUEUE'
      exporting
        l_datasource = l_datasource
        tables
          i_t_data    = t_xebkn.

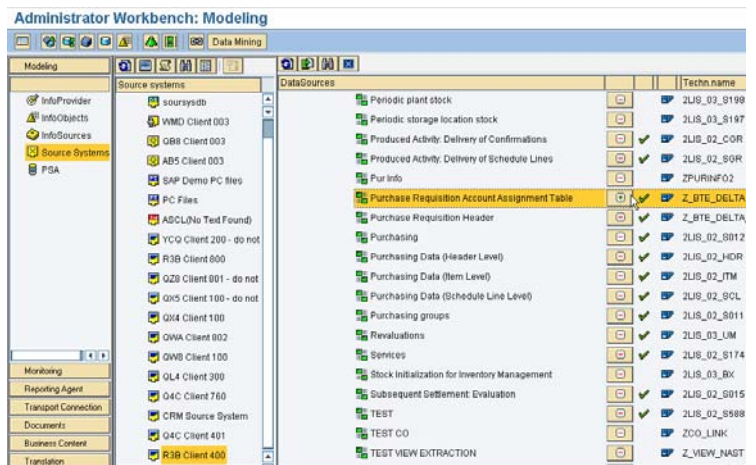
  endif.
endfunction.


```

Now save and activate the Function Module.

3.6 Test New Functionality

1. Login to the system, once in the system perform the following steps
 - a. Navigate to transaction RSA1
 - b. Under the Source System tab replicate the dataSources of the system in which you implemented this How-To.
 - c. Find the dataSource you created in the DataSource Overview Pane.



2. Navigate to the InfoSources tab and create an InfoSource that matches the Extraction Structure of your data source. Save and activate.
 - a. Go back to the Source System tab and find your DataSource again
 - b. Click on the red  sign and assign the infoSource created to your DataSource.

InfoSource Z_BTE_IS Change

InfoSource: Z_BTE_IS | InfoSource for Purchase Requisition Acco

Communication_Struct... | Status: Active

Transfer

InfoObject	Descript.	S...	R...	Check object	Type	Length	Deci...	Unit	File
0BBP_REQ_ID	Purchase Requist...				CHAR10		0		BB1
0BBP_RQITEM	Item No. in PurRe...				NU...	10	0		BB1
ZEBKN	Serial number for ...			ZEBKN	CHAR2		0		/B
0CREATEDON	Created on				DATS8		0		CRI
0CREATEDBY	Created by				CHAR12		0		CRI
0SP_PRCNT	Distribtn percenta...				FLTP	16	16		SP...
0GL_ACCOUNT	G/L Account			0GL_AC	CHAR10		0		GL...
0CHRT_ACCTS	Chart of accounts			0CHRT_	CHAR4		0		CHI
0BUS_AREA	Business area			0BUS_A	CHAR4		0		BU...
0COSTCENTER	Cost Center			0COSTC	CHAR10		0		CO...
0CO_AREA	Controlling area			0CO_AR	CHAR4		0		CO...
0DOC_NUMBER	Sales document			0DOC_	CHAR10		0		DO...
0DOC_ITEM	BW: Document Lin...				NU...	6	0		DO...
0NUM_SCHED	Displ.sched.lines				DEC	17	3		NU...

3. Now that the infoSource is created and the DataSource linked, create an infoPackage for the delta initialization. For the purposes of this How-To we will load into the PSA only. Now execute the delta init.

Scheduler (Maintain InfoPackage)

Process Chain Maint.

InfoPackage: BTE Delta Initialization(ZPAK_BK9DN5M1SC3RRMG0HUKQPUSW)

InfoSource: InfoSource for Purchase Requisition Acco(Z_BTE_IS)

DataSource: Purchase Requisition Account Assignment Table(Z_BTE_DELTA)

Data Class: Transaction Dat

Source System: R3B Client 400(ID3CLNT400)

Last Changed By: CAIRNSX | Date: 12/01/2005 | Time: 20:58:18

Data Selection | Processing | Data Targets | Update | Schedule

Consistency check for char. values in the transfer rules

Update Data...

PSA and then into Data Targets (Package by Package)
 PSA and Data Targets in Parallel (Package by Package)
 Only PSA Update Subsequently in Data Targets
 Data Targets Only

4. Verify the data within the PSA

Maintenance of PSA data request REQU_D74N5VFJDVEM6VMX7CFFYKX

Data records to be edited

Status	DataPacket	Data rec.	Purchase Requisition	Item N...	Serial nu...	Deletion	Created on	Created by	Distr
	1	125	0010004939		10	1			0.
	1	126	0010004940		10	1			0.
	1	127	0010004941		10	1			0.
	1	128	0010004942		10	1			0.
	1	129	0010004959		10	1			0.
	1	130	0010005236		10	1	11/19/1998	MEIER	0.
	1	132	0010005375		10	1			0.
	1	133	0010005376		10	1			0.
	1	134	0010005377		10	1			0.
	1	135	0010005378		10	1			0.
	1	136	0010005379		10	1			0.
	1	137	0010005672		10	1			0.
	1	140	0010007694		10	1	01/22/1999	DISPON...	0.
	1	141	0010007783		10	1			0.
	1	142	0010007784		10	1			0.
	1	143	0010007785		10	1			0.

5. Check the delta queue in your source system via transaction RSA7 and make sure a delta queue entry was created successfully.

Pflege BW Delta Queue

Stat..	DataSource	BW System
○○	Z_BTE_DELTA	WMDCLNT003
○○	SFLIGHT_DS	WMDCLNT003
○○	1_CO_PA400S001	BWINST
○○	1_CO_PAR3B400IDEAHK	QB6CLNT003
○○	1_CO_PAR3B800IDEAHK	QB6CLNT003
○○	1_CO_PA_TEST_ZBGE	BWINST
○○	1_CO_PA_TEST_ZBGE	BWINST

6. To verify that the delta hook built is working we now need to modify an existing purchase requisition and make a change to the Account Assignment information; this will then invoke the delta hook.

Display Purch. requisition 10004379

Document overview on | Personal setting

Purchase requisition 10004379

Header

Item overview

Item [10] 99-100, Oil

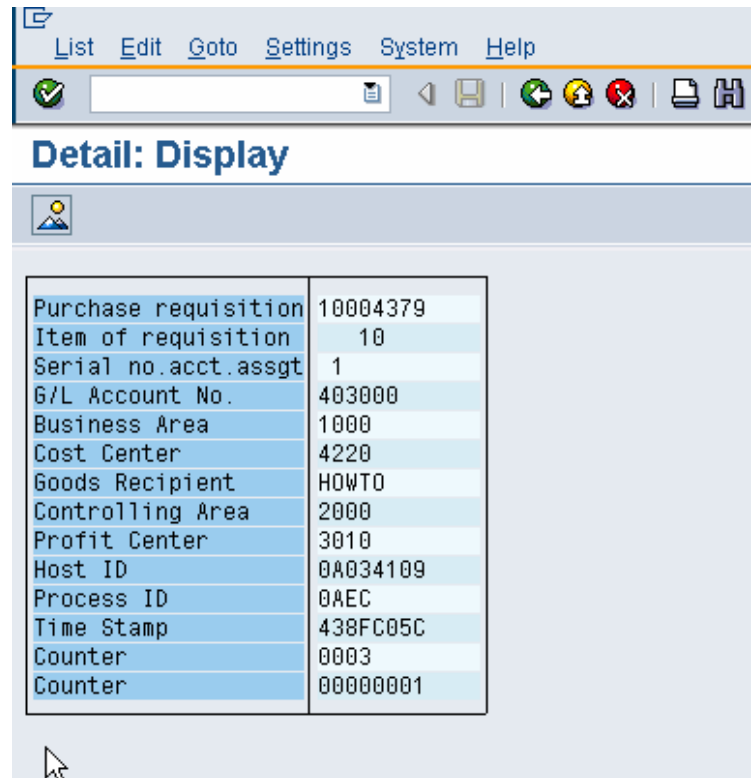
Material data | Quantities/dates | Valuation | **Account assignment** | Source of supply | Status

AccAssCat	Cost center	Distribution	Single account assignm...	CoCode IDE
Unloading Point		Recipient	HOWTO	
G/L Account No.	403000			
Business Area	1000			
CO Area	2000			
Cost Center	4220			
Earmarked Funds		0		

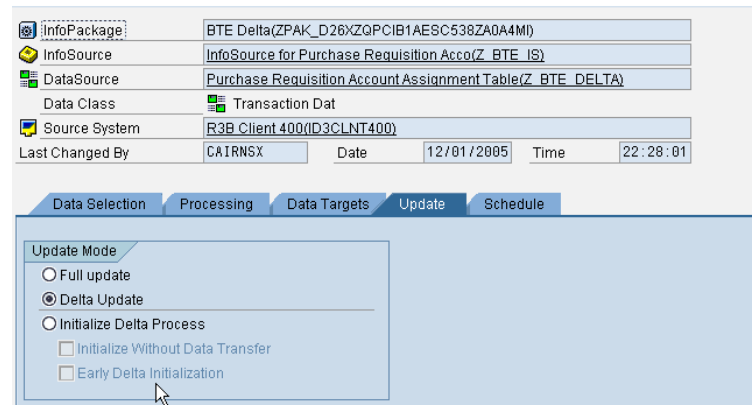
More

✓ Purchase requisition 0010004379 changed

- Verify in the delta queue the successful write of the modified record.



- In the SAP NetWeaver BI system create another infoPackage and set it up for a delta update.



- Kick off the infoPackage and validate the information within the PSA.

Maintenance of PSA data request REQU_9AUJYBBSP18ND10

Data records to be edited

Data rec.	Purchase R	Item No. i	Serial num	Deletion i	Created on	Created by	Distribtn	G/L Accou
1	0010004379	10	1				0.0	00004030

Single record change

BW: Document Line Nu 000000
 No. of schedule line 0000
 Main Asset Number
 Asset Sub-number
 Order Number
 Ship-To Party HOWTO
 Unloading Point
 Controlling area 2000
 Posting to cost cent
 Post To Order
 Post to project

4 Appendix A: Example ABAP Code

4.1 Z_CHANGE_DELTA_PROCESS

```
*&-----*
*& Report  Z_CHANGE_DELTA_PROCESS                *
*&-----*
*&-----*
*&-----*
*&-----*
*&-----*

report  z_change_delta_process

*P_DATAS      DataSource
*P_DELTAP     Delta Process for DataSource
parameters:
  p_datas type roosource-oltpsource,
  p_deltap type roosource-delta.

tables:
  roosource.

data:
  ls_roosource type roosource.

if p_datas(2) ne 'Z_'.
  message 'The DataSource needs to begin with 'Z_'' type 'E'.
endif.

select single * from roosource into ls_roosource
  where oltpsource = p_datas
  and   objvers   = 'A'.

if sy-subrc eq 0.
  ls_roosource-delta = p_deltap.
  update roosource from ls_roosource.
  message 'The DataSource has been updated successfully.' type 'I'.
else.
  message 'The DataSource entered is not valid, try again.' type 'E'.
endif.
```

4.2 Z_WRITE_TO_QUEUE

```
FUNCTION z_write_to_queue.
*-----*
*""Local interface:
* IMPORTING
*   REFERENCE(I_DATASOURCE) TYPE ROOSOURCE-OLTPSOURCE
* TABLES
*   I_T_DATA OPTIONAL
*-----*
* Developed by: Scott Cairncross

TYPE-POOLS:
  sbiwa.

DATA:
```



```

l_exstruct TYPE roosource-exstruct,
l_initflag TYPE roosprmsc-initstate,
l_subrc     TYPE sy-subrc,
lr_is_data TYPE REF TO data,
lr_es_data TYPE REF TO data,
lr_et_data TYPE REF TO data,
l_t_fields TYPE sbiwa_t_fields,
l_t_select TYPE sbiwa_t_select.

```

FIELD-SYMBOLS:

```

<i_s_data> TYPE ANY,
<e_s_data> TYPE ANY,
<e_t_data> TYPE STANDARD TABLE.

```

* Check to see if Delta initialization has been performed.

```

SELECT SINGLE initstate FROM roosprmsc INTO l_initflag
WHERE oltpsource = i_datasource
AND rlogsys     NE space
AND slogsys     NE space
AND initrnr     NE space.

```

* If initialization has taken place continue
IF sy-subrc EQ 0 AND l_initflag EQ 'X'.

* grab the extraction structure from roosource based on the
* datasource parameter input.

```

SELECT SINGLE exstruct FROM roosource INTO l_exstruct
WHERE oltpsource = i_datasource
AND objvers     = 'A'.

```

CHECK sy-subrc = 0.

```

CREATE DATA lr_is_data LIKE LINE OF i_t_data.
ASSIGN lr_is_data->* TO <i_s_data>.

```

```

CREATE DATA lr_es_data TYPE (l_exstruct).
ASSIGN lr_es_data->* TO <e_s_data>.

```

```

CREATE DATA lr_et_data TYPE STANDARD TABLE OF (l_exstruct).
ASSIGN lr_et_data->* TO <e_t_data>.

```

```

LOOP AT i_t_data ASSIGNING <i_s_data>.
  CLEAR <e_s_data>.
  MOVE-CORRESPONDING <i_s_data> TO <e_s_data>.
  INSERT <e_s_data> INTO TABLE <e_t_data>.
ENDLOOP.

```

```

CALL FUNCTION 'EXIT_SAPLRSAP_001'
  EXPORTING
    i_datasource      = i_datasource
    i_isource         = ''
    i_updmode         = ''
  TABLES
    i_t_select        = l_t_select
    i_t_fields        = l_t_fields
    c_t_data          = <e_t_data>
  EXCEPTIONS
    rsap_customer_exit_error = 1
    OTHERS                = 2.

```

IF sy-subrc <> 0.

```

  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
    WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

```

```

CALL FUNCTION 'RSC1_TRFC_QUEUE_WRITE'
  EXPORTING
    i_isource         = i_datasource

```

```

        i_no_flush    = 'X'
    IMPORTING
        e_subrc       = l_subrc
    TABLES
        i_t_data      = <e_t_data>
    EXCEPTIONS
        name_too_long = 1
        OTHERS        = 2.
    IF sy-subrc <> 0.
        MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
    ENDIF.

ENDIF.

ENDFUNCTION.

```

```

*-----*
*      FORM abs_type_to_rel_type      *
*-----*
*      The purpose of this subroutine is to convert an absolute type *
*      name into a relative type name. *
*-----*
* --> TYPE_NAME *
*-----*
form abs_type_to_rel_type changing type_name.
  data junk(100) type c.

  split type_name at '\TYPE=' into junk type_name.

endform.

```

4.3 ZSC_BTE_DELTA_CAPTURE_01000710

```

function zsc_bte_delta_capture_01000710.
*-----*
*""Local interface:
* TABLES
* T_XEBAN STRUCTURE EBAN OPTIONAL
* T_XEBKN STRUCTURE EBKN OPTIONAL
* T_YEBAN STRUCTURE EBAN OPTIONAL
* T_YEBKN STRUCTURE EBKN OPTIONAL
*-----*
  data:
    l_datasource type roosource-oltpsource.

    l_datasource = 'Z_BTE_DELTA'.

    if lines( t_xebkn[] ) gt 0.

      call function 'Z_WRITE_TO_QUEUE'
        exporting
          i_datasource = l_datasource
        tables
          i_t_data      = t_xebkn.

    endif.
endfunction.

```

www.sdn.sap.com/irj/sdn/howtoguides