SAP PRESS

Narrowing
Down Data
Selection o
MultiProvi
with Query
Pruning

#56 CORRECTLY POPULATING THE PROCESS KEYS FOR LOGISTICS EXTRACTORS

Tip 38

FLEXIBLE CALCULATIONS

Combining the Results of Two Queries Using the Application Process Designer

TIP 14 NING RCHY ASED LITY TING

ENABLING THE USE OF SAP BUSINESSOBJECTS EXPLORER IN AN SAP NETWEAVER BW ON HANA ENVIRONMENT

100 Things You Should Know About

SAP NetWeaver® BW

TIP 99 RESTORING A BEx QUERY FROM VERSION 7.x TO 3.x

#7 Converting Standard InfoCubes to SAP HANA Optimized InfoCubes

USING THE ANALYSIS PROCESS DESIGNER AS A DATA MODELING OBJECT

TRIGGERING A REPORT TO CREAT

DYNAMIC MEASURES FOR BEX KEY

#62 FIGURES IN OLAP UNIVERSES

- 100 little-known time-saving tips and tricks
- Step-by-step instructions and guiding screenshots
- Practical, expert advice for anyone working in SAP NetWeaver BW

Buntic Georgian
Andrew Joo

Galileo Press

Buntic Georgian and Andrew Joo

100 Things You Should Know About SAP NetWeaver® BW

Galileo Press

Bonn • Boston

Dear Reader,

What is 100? Depending on how you look at it, it can be a very small or large number. $100 doesn't get you far these days, but 100 years of life is still an impressive milestone. And when you've worked on a lot of 100 Things books, you realize that 100 takes on a different meaning depending on what type of book your authors are writing.

The life that this book has taken on is a large one, indeed partly due to the sheer size of the topic. With the vastness of SAP NetWeaver BW, it was a task in and of itself to come up not with 100 tips, but with the *most* helpful 100 tips. After countless revisions and reorganizations of the tips in this book, however, Buntic and Andrew have narrowed down over 200 ideas to be presented to you here.

As always, we appreciate your business and welcome your feedback. Your comments and suggestions are the most useful tools to help us improve our books for you, the reader. We encourage you to visit our website at *www.sap-press.com* and share your feedback about *100 Things You Should Know About SAP NetWeaver BW*.

Thank you for purchasing a book from SAP PRESS!

Laura Korslund
Editor, SAP PRESS

Galileo Press
Boston, MA

laura.korslund@galileo-press.com
www.sap-press.com

Imprint

This e-book is a publication many contributed to, specifically:

Editor Laura Korslund
Acquisitions Editor Kelly Grace Weaver
Copyeditor Julie McNamee
Cover Design Graham Geary
Production E-Book Graham Geary
Typesetting E-Book SatzPro, Krefeld (Germany)

We hope that you liked this e-book. Please share your feedback with us and read the Service Pages to find out how to contact us.

The Library of Congress has cataloged the printed edition as follows:
Joo, Andrew.
100 things you should know about SAP NetWeaver BW / Andrew Joo and
Buntic Georgian. -- 1st edition.
pages cm
Includes indexes.
ISBN 978-1-59229-447-3 -- ISBN 1-59229-447-2 -- ISBN 978-1-59229-645-3 --
ISBN 978-1-59229-646-0
1. SAP NetWeaver BW. I. Georgian, Buntic. II. Title. III. Title: One hundred things
you should know about SAP NetWeaver BW. IV. Title: Hundred things you should
know about SAP NetWeaver BW.
TK5105.8885.S24J66 2013
658.4'038028553--dc23
2013024828

ISBN 978-1-59229-447-3 (print)
ISBN 978-1-59229-645-3 (e-book)
ISBN 978-1-59229-646-0 (print and e-book)

Contents

Acknowledgments

We'd like to express our gratitude to the people who have been instrumental in the successful completion of this project.

First we'd like to thank Phani Kunchala and Ken Wong for helping us to develop the various scenarios described in the book and for his constant support on the actual development of the tips.

We would also like to thank Jai Selvarajan, Brett Leeladharan, Kumar Varadarajan, Adeel Hashmi, Anil Kandala, Rao Vedula, Shlomi Weiss, and Adir Oren and Theo Kazakopoulos for their contributing tip ideas and content.

We are grateful to Praveen Kola and Deepu Sasidharan for contributing their technical expertise to help us brainstorm topics, review content, and support the development of the book.

The guidance and support received from all these individuals who contributed was vital for the success of the project.

And last but not least, we would like to thank our family for all the support, understanding, and encouragement they provided throughout the process.

Introduction

This book is part of SAP PRESS's small series of reference books that's based on 100 tips and tricks for various SAP software components and platforms that come directly from industry experts. It's designed to make reading and understanding SAP topics more interesting and accessible for your day-to-day work. The tips in this book have been carefully selected to help you gain access to the expert SAP NetWeaver BW knowledge that otherwise might take you years to accumulate. Each part of this book covers a particular task area of SAP NetWeaver BW, but each tip is standalone, so you can look through the book to find the topic you want without having to read an entire chapter.

SAP NetWeaver Business Warehouse is one of SAP's key business intelligence platforms. It is applicable across the entire SAP portfolio, and working with it requires a vast array of knowledge. While the purpose of this book is not to give you a complete overview of the platform, you will benefit from reading the tips in this book to supplement your knowledge or learning to find special insight into the system.

How to Read This Book

This book is organized into five parts, with each part covering a major task within the SAP NetWeaver BW system.

- **Part 1, SAP NetWeaver BW Data Modeling**: This part of the book will provide insights into some of the most critical aspects of data modeling, beginning with the concept of data warehousing and working with the architecture of SAP NetWeaver BW.
- **Part 2, SAP NetWeaver BW Reporting and Analysis**: Here you'll find many tips and tricks that will help you to implement, execute, and optimize reporting and analysis functions with SAP Business Explorer (SAP BEx) tools.
- **Part 3, SAP NetWeaver BW Data Flow**: This section will cover key topics in the data flow, specifically the extraction, transformation, and loading (ETL) process.
- **Part 4, SAP NetWeaver BW Administration and Development**: This section will cover some key topics in the administration of the SAP NetWeaver BW system, including system performance optimization and tuning.

- **Part 5, Integration**: This section will cover tips and tricks on SAP NetWeaver BW integration areas such as SAP Data Services, SAP BusinessObjects Business Intelligence tools, and SAP HANA.

Who This Book is For

Each tip in this book aims to replicate a scenario in which a skilled SAP expert is by your side, demonstrating how to best and most efficiently accomplish a task. It assumes a basic knowledge of functionality in SAP NetWeaver BW on your part. This book should be used as a companion and supplement for consultants and users who are working with this platform.

Part 1

SAP NetWeaver BW Data Modeling

Things You'll Learn in this Section

Effective data modeling is your essential foundation for building a successful SAP NetWeaver Business Warehouse (BW) system. In this part of the book, we'll provide tips and tricks that relate to some of the most critical aspects of data modeling, beginning with the concept of data warehousing and the architecture of SAP NetWeaver BW.

You'll benefit from step-by-step instructions on a wide range of topics on extraction, transformation, and loading (ETL) such as creating mockup data, flattening a hierarchy for reporting flexibility, and finding the lineage of an InfoObject in a DataSource. This part will also address some core modeling topics such as converting a key figure model to an account model and modeling planning scenarios using direct update DSOs. We'll also provide you with insight into modeling capabilities with SAP NetWeaver BW version 7.3.

Tip 1

Creating Mock Data for an InfoCube for Testing

You can use an SAP provided, standard utility program to create mockup data for InfoCubes that you can use for testing purposes.

Lack of quality test data in a development environment of the source systems is a persistent problem in most SAP NetWeaver Business Warehouse (SAP NetWeaver BW) platforms. Even if some data is available in the source systems, to load data into an InfoCube, the end-to-end extracting, transformation, and loading (ETL) process needs to be built, which puts constraints on the frontend report development as lack of data is in the critical path for the development of the reports.

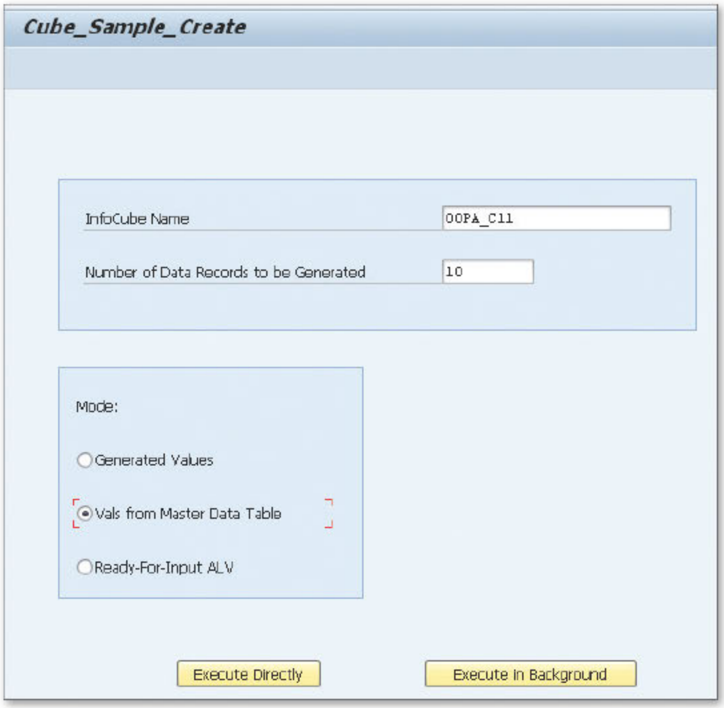
This limitation results in changes with limited testing being moved into the acceptance box and increasing the recycling of changes from development to acceptance systems.

This tip describes a method to create mockup transaction data using master data or to create data even without existing master data. This method can be used to generate as many records as needed and also to edit the mockup data to refine it further. This enables developers to perform quality testing in the development box on both backend as well as frontend development objects without implementing the ETL process.

✓ And Here's How ...

To create mock data, execute Transaction SE38 to invoke the ABAP Editor and follow these steps:

1. Enter program name CUBE_SAMPLE_CREATE and click on CREATE.
2. In the next screen shown in Figure 1, enter the name of the InfoCube for which the mockup data needs to be created, as well as the number of records to be generated.


Cube_Sample_Create

InfoCube Name	0OPA_C11
Number of Data Records to be Generated	10

Mode:
Generated Values [✓ Vals from Master Data Table] Ready-For-Input ALV

Figure 1 Creating Mockup Data

3. Select the MODE of the sample data creation. On execution—depending upon the chosen mode—the system will generate random master data and transaction data values, generate transaction data based on master data, or allow direct entry of transaction data.

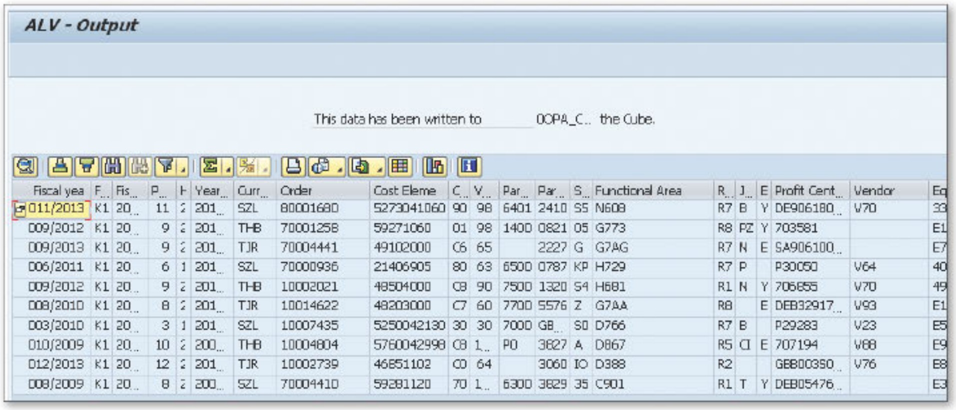
The three modes available for data creation are as follows:

- GENERATED VALUES
  With this mode, you can generate values without referencing any master data. The system generates dummy values for the master data objects and generates the transaction data based on that. This is truly dummy data, and you may use this option if you don't have any master data available in the system.

- **Vals from Master Data Table**

  The next mode generates the sample data using values from the master data. The system automatically uses the master data tables of the InfoObjects included in the InfoCube and generates transaction data based on that. Select this option in Figure 1, and click on Execute Directly.

  The system generates mock data based on the master data and saves the data into the InfoCube. See Figure 2 for the output of the data creation.



ALV - Output

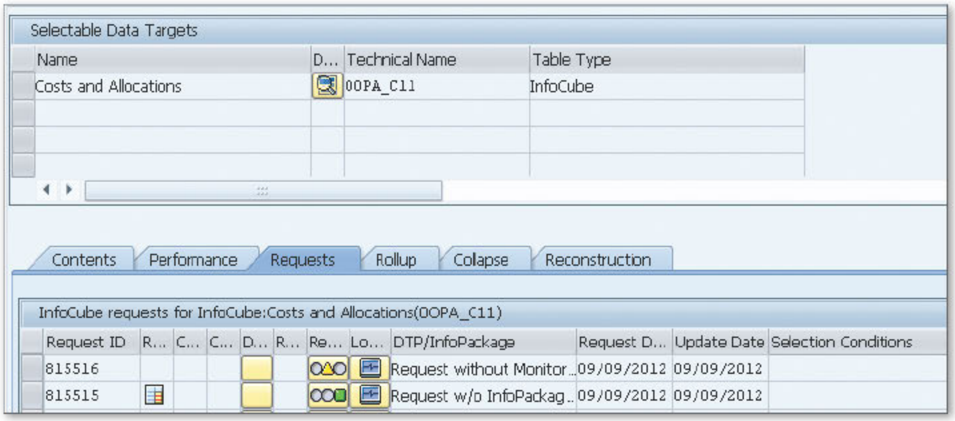
This data has been written to 0OPA_C.. the Cube.

Fiscal yea	F...	Fis...	P...	H	Year...	Curr...	Order	Cost Eleme	C...	V...	Par...	Par...	S...	Functional Area	R...	J...	E	Profit Cent...	Vendor	Eq
011/2013	K1	20...	11	2	201...	SZL	80001680	5273041060	90	98	6401	2410	S5	N608	R7	B	Y	DE906180...	V70	33
009/2012	K1	20...	9	2	201...	THB	70001258	59271060	01	98	1400	0821	05	G773	R8	PZ	Y	703581		E1
009/2013	K1	20...	9	2	201...	TJR	70004441	49102000	C6	65		2227	G	G7AG	R7	N	E	SA906100...		E7
006/2011	K1	20...	6	1	201...	SZL	70000936	21406905	80	63	6500	0787	KP	H729	R7	P		P30050	V64	40
009/2012	K1	20...	9	2	201...	THB	10002021	48504000	C8	90	7500	1320	S4	H681	R1	N	Y	706855	V70	49
008/2010	K1	20...	8	2	201...	TJR	10014622	48203000	C7	60	7700	5576	Z	G7AA	R8		E	DEB32917...	V93	E1
003/2010	K1	20...	3	1	201...	SZL	10007435	5250042130	30	30	7000	GB...	S0	D766	R7	B		P29283	V23	E5
010/2009	K1	20...	10	2	200...	THB	10004804	5760042998	C8	1...	P0	3827	A	D867	R5	CI	E	707194	V88	E9
012/2013	K1	20...	12	2	201...	TJR	10002739	46851102	C0	64		3060	IO	D388	R2			GB800350...	V76	E8
008/2009	K1	20...	8	2	200...	SZL	70004410	59281120	70	1...	6300	3829	35	C901	R1	T	Y	DEB05476...		E3

Figure 2 Mock Data Created Based on Master Data

- **Ready-For-Input ALV**

  The third mode for data creation is to generate data using data entry as shown in Figure 3. This option allows the user to enter data directly on the screen.



Selectable Data Targets

Name	D...	Technical Name	Table Type
Costs and Allocations		0OPA_C11	InfoCube

InfoCube requests for InfoCube:Costs and Allocations(0OPA_C11)

Request ID	R...	C...	C...	D...	R...	Re...	Lo...	DTP/InfoPackage	Request D...	Update Date	Selection Conditions
815516								Request without Monitor...	09/09/2012	09/09/2012	
815515								Request w/o InfoPackag...	09/09/2012	09/09/2012	

Figure 3 Screen for Input for Direct Data Entry

With all these modes, you can execute the data creation in the background and use the standard managed InfoCube functionality to review the status of the data creation tasks and take corrective action as needed. You'll be able to see the requests that are generated for the data creation activity in the INFOCUBE MONITORING screen.

With this process, you can create as much mockup data as you need, and the data will be available in the InfoCubes for testing. This is really helpful for rapid application development, enables you to build SAP Business Explorer or SAP BusinessObjects reports on top of this data, and allows you to test the report functionality.

Tip 2

Using Remodeling Rules to Change the Structure of an InfoCube after Data Load

Instead of deleting or copying an InfoCube, you can create rules to "remodel" the structure of an InfoCube that already contains data.

Changing the structure of an InfoCube with data content is a common requirement, particularly during any project development phase. However, before SAP NetWeaver BW release 7.0, the only solution to the problem was to either delete the data or perform a long series of steps that included making a copy of the InfoCube.

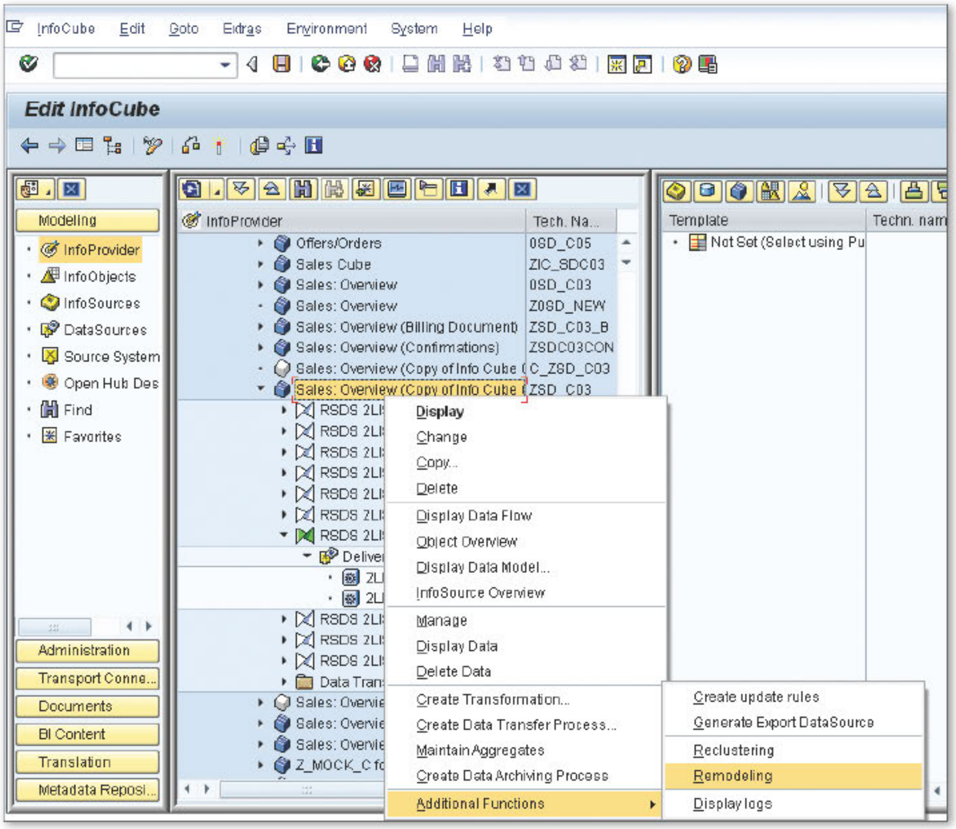
This tip describes a new functionality called *remodeling* (available as of SAP NetWeaver BW 7.0) that you can use to address these types of scenarios. It's efficient and solves the issue of changing the structures of the InfoCube even in production environments without any data loss.

✓ And Here's How ...

You can create and schedule remodeling rules to change the structure of the InfoCube. In this tip, we'll illustrate an example in which you want to delete a key figure from an InfoCube. Follow these steps:

1. Execute Transaction RSA1 in the SAP NetWeaver BW system.

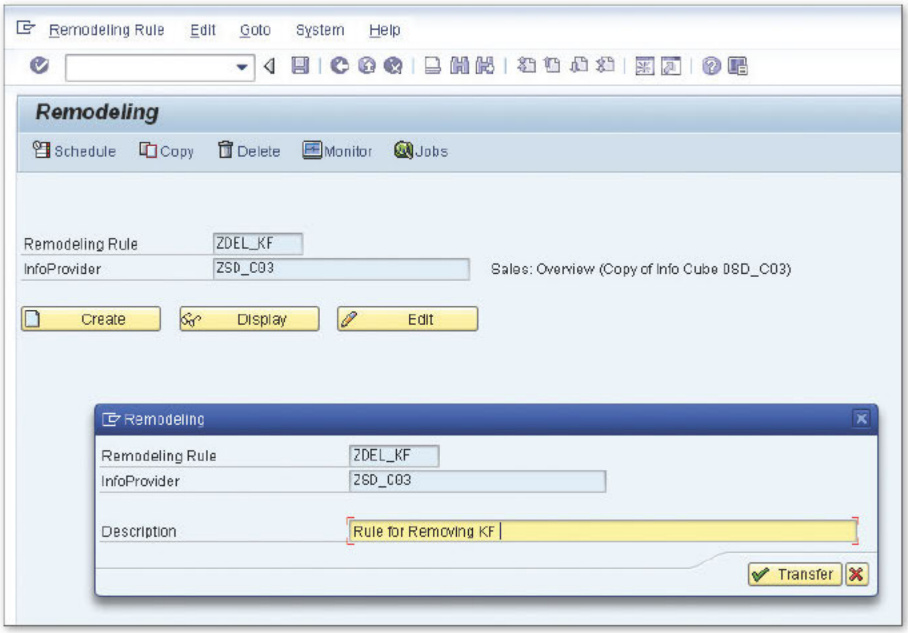
2. Hover your mouse over the InfoCube you want to change from the list of Info-Providers, right-click, and select ADDITIONAL FUNCTIONS and then REMODELING (see Figure 1).



InfoProvider	Tech. Na...
Offers/Orders	0SD_C05
Sales Cube	ZIC_SDC03
Sales: Overview	0SD_C03
Sales: Overview	Z0SD_NEW
Sales: Overview (Billing Document)	ZSD_C03_B
Sales: Overview (Confirmations)	ZSDC03CON
Sales: Overview (Copy of Info Cube (	C_ZSD_C03
Sales: Overview (Copy of Info Cube (	ZSD_C03
RSDS 2LI	
RSDS 2LI	
RSDS 2LI	
RSDS 2LI	
RSDS 2LI	
RSDS 2LI	
RSDS 2LI	
Deliver	
2LI	
2LI	
RSDS 2LI	
RSDS 2LI	
RSDS 2LI	
Data Tran	
Sales: Overvie	
Sales: Overvie	
Sales: Overvie	
Z_MOCK_C fo	

Figure 1 Choose to Remodel an Existing InfoCube

3. On the resulting screen shown in Figure 2, right-click on the InfoCube to create a new remodeling rule. In this screen, specify a technical name for the remodeling rule. In this example, we'll use the technical name "ZDEL_KF" for the new rule. You may also use this screen to display or edit an existing remodeling rule.



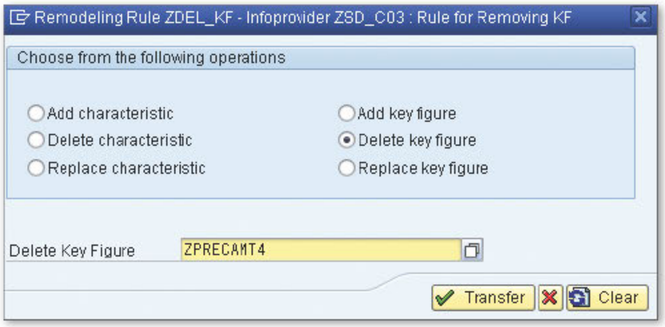
Remodeling Rule ZDEL_KF
InfoProvider ZSD_C03 Sales: Overview (Copy of Info Cube 0SD_C03)

Remodeling

Remodeling Rule ZDEL_KF
InfoProvider ZSD_C03
Description Rule for Removing KF

Figure 2 Creating a New Remodeling Rule for the InfoCube

4. Provide a short description for the rule (e.g., in Figure 2, we've entered "Rule for Removing KF"), and click on TRANSFER to continue.
5. Click on the toolbar button on the left to add a new rule.
6. On the screen that appears (see Figure 3), select the DELETE KEY FIGURE radio button, select the name of the key figure to delete, and click TRANSFER. A new operation is created in the rule. Note that a remodeling rule can have more than one operation.



Remodeling Rule ZDEL_KF - Infoprovider ZSD_C03 : Rule for Removing KF

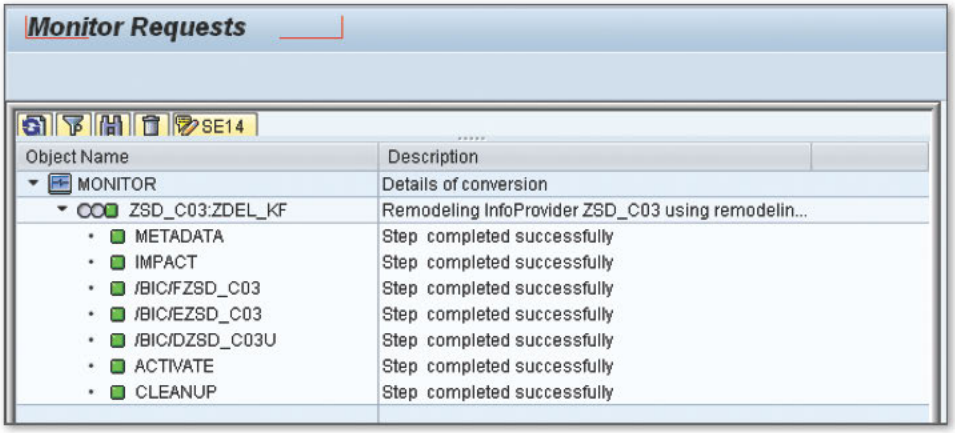
Choose from the following operations

○ Add characteristic	○ Add key figure
○ Delete characteristic	● Delete key figure
○ Replace characteristic	○ Replace key figure

Delete Key Figure ZPRECAMT4

Figure 3 Create a New Rule

7. Click on the SAVE icon to save the rule.
8. Click on the SCHEDULE button to schedule the rule to be executed.
9. On the resulting screen, click on START IMMEDIATELY to execute the rule job. Alternately, you can select START LATER to schedule the job if there is a huge of volume of data in the InfoCube.
10. Click on the MONITOR button as shown in Figure 4 to view the progress of the job; here, you can see the successful completion of the various steps in the remodeling process.



Object Name	Description
MONITOR	Details of conversion
ZSD_C03:ZDEL_KF	Remodeling InfoProvider ZSD_C03 using remodelin...
METADATA	Step completed successfully
IMPACT	Step completed successfully
/BIC/FZSD_C03	Step completed successfully
/BIC/EZSD_C03	Step completed successfully
/BIC/DZSD_C03U	Step completed successfully
ACTIVATE	Step completed successfully
CLEANUP	Step completed successfully

Figure 4 Job Progress: Remodeling Successful

Tip 3

Flattening a Hierarchy for Increased Flexibility in Reporting

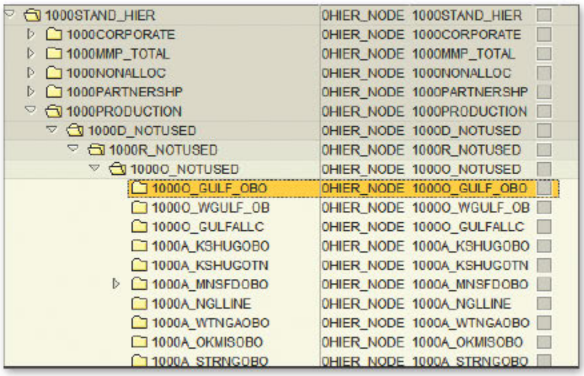
You can flatten a standard hierarchy by converting it to a characteristic-attribute relationship so end users can benefit from increased flexibility in performing BI analysis.

A hierarchy is a useful way of displaying master data in SAP NetWeaver BW because it allows you to filter on any level of the hierarchy and enables users to drill down or drill up on the data. However, the strict structure of the hierarchy imposes some limitations on report formatting. Individual hierarchy nodes cannot appear in separate columns. You can't reorder or swap the individual nodes of a hierarchy in a report. Further, hierarchies cannot be easily used in InfoSpokes either as filter criteria or in the outbound extract.

This tip describes a method to overcome this problem by flattening the hierarchy and including the nodes relevant to a particular master data object within its navigational attributes. Flattening the hierarchy gives a higher level of flexibility to the business users to filter on these navigational attributes within SAP NetWeaver BW reporting and also places the navigational attributes in separate columns. This dynamically reorders the appearance of the individual columns and allows the user to easily swap the columns to slice and dice data. You can also use navigational attributes as filters in an InfoSpoke or in the outbound extract.

✓ And Here's How ...

Here, we'll show you how to map and convert SAP ERP data to a characteristic-attribute relationship based on InfoObject 0COSTCENTER. The SAP ERP cost center master data has a hierarchy called 1000STAND_HIER with the structure shown in Figure 1.



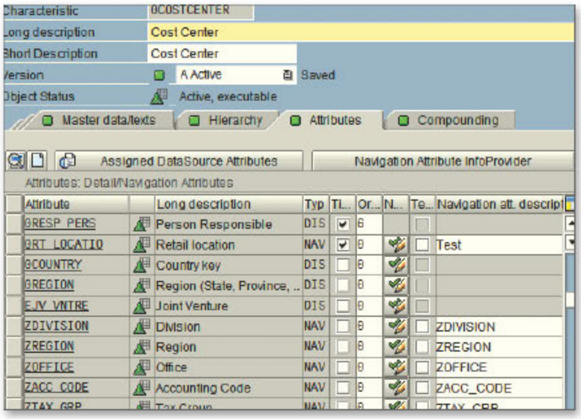
1000STAND_HIER	0HIER_NODE	1000STAND_HIER
1000CORPORATE	0HIER_NODE	1000CORPORATE
1000MMP_TOTAL	0HIER_NODE	1000MMP_TOTAL
1000NONALLOC	0HIER_NODE	1000NONALLOC
1000PARTNERSHP	0HIER_NODE	1000PARTNERSHP
1000PRODUCTION	0HIER_NODE	1000PRODUCTION
1000D_NOTUSED	0HIER_NODE	1000D_NOTUSED
1000R_NOTUSED	0HIER_NODE	1000R_NOTUSED
1000O_NOTUSED	0HIER_NODE	1000O_NOTUSED
1000O_GULF_OBO	0HIER_NODE	1000O_GULF_OBO
1000O_WGULF_OB	0HIER_NODE	1000O_WGULF_OB
1000O_GULFALLC	0HIER_NODE	1000O_GULFALLC
1000A_KSHUGOBO	0HIER_NODE	1000A_KSHUGOBO
1000A_KSHUGOTN	0HIER_NODE	1000A_KSHUGOTN
1000A_MNSFDOBO	0HIER_NODE	1000A_MNSFDOBO
1000A_NGLLINE	0HIER_NODE	1000A_NGLLINE
1000A_WTNGAOBO	0HIER_NODE	1000A_WTNGAOBO
1000A_OKMISOBO	0HIER_NODE	1000A_OKMISOBO
1000A_STRNGOBO	0HIER_NODE	1000A_STRNGOBO

Figure 1 Cost Center Hierarchy in SAP ERP

We'll show you how to flatten the hierarchy with an ABAP program that's designed to read the cost center hierarchy and update the cost center master data object.

Follow these steps to flatten the hierarchy:

1. In SAP NetWeaver BW, add additional attributes to 0COSTCENTER as shown in Figure 2. The new attributes are DIVISION, REGION, OFFICE, ACCOUNTING CODE, TAX GROUP, REGULATORY FIELD, PROPERTY CASE, and WELL NUMBER. The flattening program will flatten the hierarchy and save the values in the node to these attributes in 0COSTCENTER.



Characteristic	0COSTCENTER
Long description	Cost Center
Short Description	Cost Center
Version	A Active · Saved
Object Status	Active, executable

Master data/texts
Hierarchy
Attributes
Compounding
Assigned DataSource Attributes
Navigation Attribute InfoProvider
Attributes: Detail/Navigation Attributes

Attribute	Long description	Typ	Ti...	Or...	N...	Te...	Navigation att. descript
0RESP_PERS	Person Responsible	DIS	✓	6			
0RT_LOCATIO	Retail location	NAV	✓	0	✓	☐	Test
0COUNTRY	Country key	DIS	☐	0	✓		
0REGION	Region (State, Province, ..	DIS	☐	0	✓		
EJV_VNTRE	Joint Venture	DIS	☐	0	✓		
ZDIVISION	Division	NAV	☐	0	✓	☐	ZDIVISION
ZREGION	Region	NAV	☐	0	✓	☐	ZREGION
ZOFFICE	Office	NAV	☐	0	✓	☐	ZOFFICE
ZACC_CODE	Accounting Code	NAV	☐	0	✓	☐	ZACC_CODE
ZTAX_GRP	Tax Group	NAV	☐	0	✓	☐	ZTAX_GRP

Figure 2 Adding New Navigational Attributes to the Cost Center InfoObject to Hold the Flattened Nodes

2. Create custom program for flattening the cost center hierarchy with the following code. This particular code has been written for the cost center object, but can be generically used for any SAP ERP hierarchy by modifying the code as necessary.

```
report  z_flatten_cc_hier.
tables:   /bi0/pcostcenter.

* Cost Center Internal Table
data: ic_data like standard table of /bi0/pcostcenter.
data: wa_ic_data like /bi0/pcostcenter.
data: up_data like standard table of /bi0/pcostcenter.

* Hierarchy declarations
data: it_costcenter like hashed table of /bi0/hcostcenter with
unique key hieid objvers nodeid iobjnm nodename .
data: do_costcenter like hashed table of /bi0/hcostcenter with
unique key hieid objvers nodeid iobjnm nodename.
data: wa_costcenter like /bi0/hcostcenter,
      do_wa_costcenter like /bi0/hcostcenter.

data: i/bi0/pcostcenter like /bi0/pcostcenter occurs 0 with header
line.

data: begin of p/bi0/pcostcenter occurs 100.
        include structure /bi0/pcostcenter.
data:  index type i,
end of p/bi0/pcostcenter.

data: xref type ref to cx_sy_dynamic_osql_semantics.

* Copy the cost center attributes structure into memory
try.
    select * from /bi0/pcostcenter                     "client
specified
                    appending table i/bi0/pcostcenter
    where co_area = '1000'
    and   objvers = 'A'.

  catch cx_sy_dynamic_osql_semantics into xref.
```

```
    if xref->kernel_errid = 'SAPSQL_ESCAPE_WITH_POOLTABLE'.
      message i412(mo).
      exit.
    else.
      raise exception xref.
    endif.
endtry.

ic_data[] = i/bi0/pcostcenter[].

* Copy the cost center hierarchy structure into memory
Select   hieid objvers nodeid iobjnm nodename tlevel link parentid
childid            nextid intervl
   into table it_costcenter from /bi0/hcostcenter.

do_costcenter[] = it_costcenter[].

* Search for parent nodes for Div, Region, Office, Acc.code,
Taxgrp,
* Reg.fld, Prod.case, Well No. Costcenter
* and update the P table with the new rows.

data: doloop(2).
sort ic_data by costcenter.
refresh up_data.

loop at ic_data into wa_ic_data.

* read Hierarchy table
  read table it_costcenter into wa_costcenter
                         with key objvers    = 'A'
                                  iobjnm     = '0COSTCENTER'
                                  nodename+4 = wa_ic_data-cost-
center.
  doloop = wa_costcenter-tlevel - 3.
  check doloop gt 0.
  do_wa_costcenter-parentid = wa_costcenter-parentid.
  do doloop times.
```

```
      read table do_costcenter into do_wa_costcenter
                     with key objvers    = 'A'
                          nodeid = do_wa_costcenter-parentid.

      check sy-subrc = 0.

      case do_wa_costcenter-tlevel.
        when '10'.
          wa_ic_data-/bic/zwell_no =  do_wa_costcenter-nodename.
        when '9'.
          wa_ic_data-/bic/zprop_cse = do_wa_costcenter-nodename.
        when '8'.
          wa_ic_data-/bic/zreg_fld = do_wa_costcenter-nodename.
        when '7'.
          wa_ic_data-/bic/ztax_grp = do_wa_costcenter-nodename.
        when '6'.
          wa_ic_data-/bic/zacc_code = do_wa_costcenter-nodename.
        when '5'.
          wa_ic_data-/bic/zoffice = do_wa_costcenter-nodename.
        when '4'.
          wa_ic_data-/bic/zregion = do_wa_costcenter-nodename.
        when '3'.
          wa_ic_data-/bic/zdivision = do_wa_costcenter-nodename.
      endcase.
      do_wa_costcenter-tlevel = do_wa_costcenter-tlevel - 1.
    enddo.

      append wa_ic_data to up_data.

  *    Update dbtable
       Update /bi0/pcostcenter from wa_ic_data.
  endloop.
```

3. Save the node values to the relevant navigational attributes of the 0COSTCENTER master data.
4. Before running the ABAP program, you must do the following:
   - Schedule the program as a background job to ensure that the navigational attributes and hierarchies remain in sync.

  - After the nightly hierarchy refresh from SAP ERP, use process chains to trigger the apply hierarchy/attribute change run in SAP NetWeaver BW. After running the ABAP program nightly, execute the apply hierarchy/attribute change run in SAP NetWeaver BW to ensure that the SIDs for 0COSTCENTER are generated and that the new attributes are available for reporting

The program logic will work as follows:

- The hierarchies in SAP NetWeaver BW are organized in a standard hierarchy table structure, and the hierarchy table names start with BI0/H. In this case, for 0COSTCENTER, the hierarchy table will be BI0/HCOSTCENTER.
- The program will start from the bottom or leaf nodes of the hierarchy structure and traverse its way up the hierarchy until it reaches the top level. There is no dependency on the number of levels in the hierarchy or attributes corresponding to that hierarchy node level. The program updates the InfoObject master data table /BI0/PCOSTCENTER with the node level information from the hierarchy table /BI0/HCOSTCENTER.

Tip 4

Implementing a Custom Conversion Routine for an InfoObject

For greater flexibility when working with logic, you can create custom conversion routines for InfoObjects to handle issues that are related to standard data manipulation.

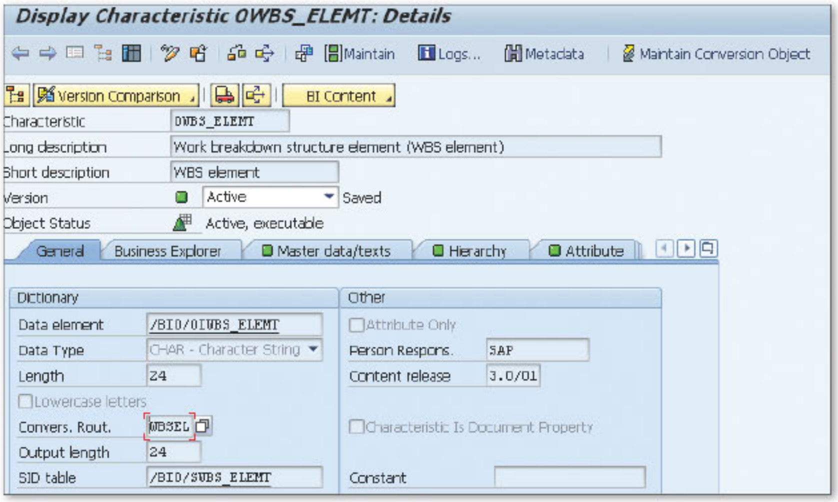
Conversion routines (also known as conversion exits) are used in SAP NetWeaver BW so that the characteristic values (key) of an InfoObject can be displayed or used in a different format than how they are stored in the database. Sometimes, specific project scenarios require an SAP NetWeaver BW developer to use a custom conversion routine for an InfoObject instead of an SAP-delivered routine to implement a different logic to output InfoObject field values.

This tip illustrates how to implement a custom conversion routine for an InfoObject in SAP NetWeaver BW 7.x and shows how the conversion exit can facilitate master data loading. The custom conversion exits provide a developer more flexibility on implementing specific logic that cannot be accomplished in a standard delivered routine.

✓ And Here's How ...

To implement a custom conversion routine for an InfoObject, access Transaction RSD1 (Display 0WBS_ELEMT InfoObject) and follow these steps:

1. On the General tab in the resulting screen, double-click on the conversion routine WBSEL to see the function module behind the routine (see Figure 1).



Display Characteristic 0WBS_ELEMT: Details

Characteristic	0WBS_ELEMT
Long description	Work breakdown structure element (WBS element)
Short description	WBS element
Version	Active Saved
Object Status	Active, executable

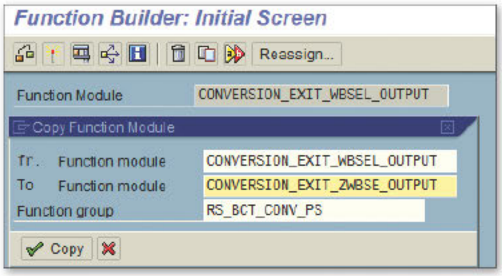
Dictionary		Other	
Data element	/BI0/OIWBS_ELEMT	☐ Attribute Only	
Data Type	CHAR - Character String	Person Respons.	SAP
Length	24	Content release	3.0/01
☐ Lowercase letters			
Convers. Rout.	WBSEL	☐ Characteristic Is Document Property	
Output length	24		
SID table	/BI0/SWBS_ELEMT	Constant	

Figure 1 Function Module Behind the Routine

You'll see the following two routines delivered by SAP from the Transaction SE37 Function Builder: Initial Screen:

- CONVERSION_EXIT_WBSEL_INPUT
- CONVERSION_EXIT_WBSEL_OUTPUT

2. For this example, copy the function module CONVERSION_EXIT_WBSEL_OUTPUT to a custom function module so that you can change the existing logic.[1] This can be done using the copy function in the Transaction SE37 Function Builder: Initial Screen (see Figure 2).



Function Builder: Initial Screen

Function Module CONVERSION_EXIT_WBSEL_OUTPUT

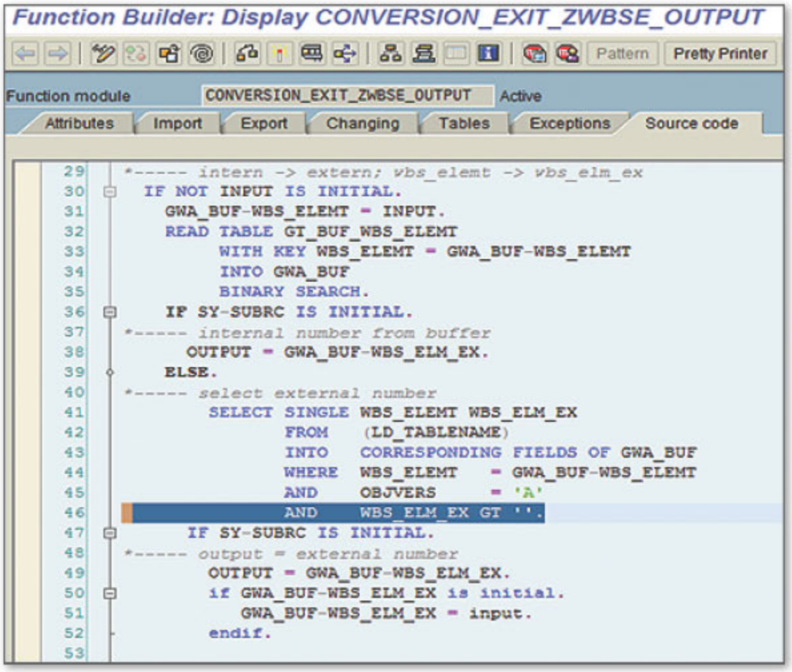
Copy Function Module

Fr. Function module	CONVERSION_EXIT_WBSEL_OUTPUT
To Function module	CONVERSION_EXIT_ZWBSE_OUTPUT
Function group	RS_BCT_CONV_PS

Figure 2 Copy Function Module on the Transaction SE37 Screen

1 "WBSEL" has been replaced with "ZWBSE". It is important that you follow a specific naming convention when you copy the function module to a custom function module. The function module name must always start with "CONVERSION_EXIT" followed by a five-character name for the conversion routine (in this case, "_ZWBSE"), followed by "_INPUT" or "_OUTPUT" as the suffix based on whether it is an input or output conversion exit.

3. Click the Copy icon in this screen, and then enter the name of the new custom function module in the To Function Module field.
4. Alternatively, if you are not copying an existing standard routine, then you can use the same Transaction SE37 Function Builder: Initial Screen to create a new custom function module by clicking on the Create button in the previous Function Builder: Initial Screen screen.
5. After you enter the name of the custom function module in Figure 2, click the Copy button on that screen to open the Function builder screen (see Figure 3).
6. Go to the Source Code tab and implement the required logic that will correctly output the InfoObject value. In Figure 3, a change of logic in the `Where` clause of the `Select` statement has been implemented.



Function module CONVERSION_EXIT_ZWBSE_OUTPUT Active

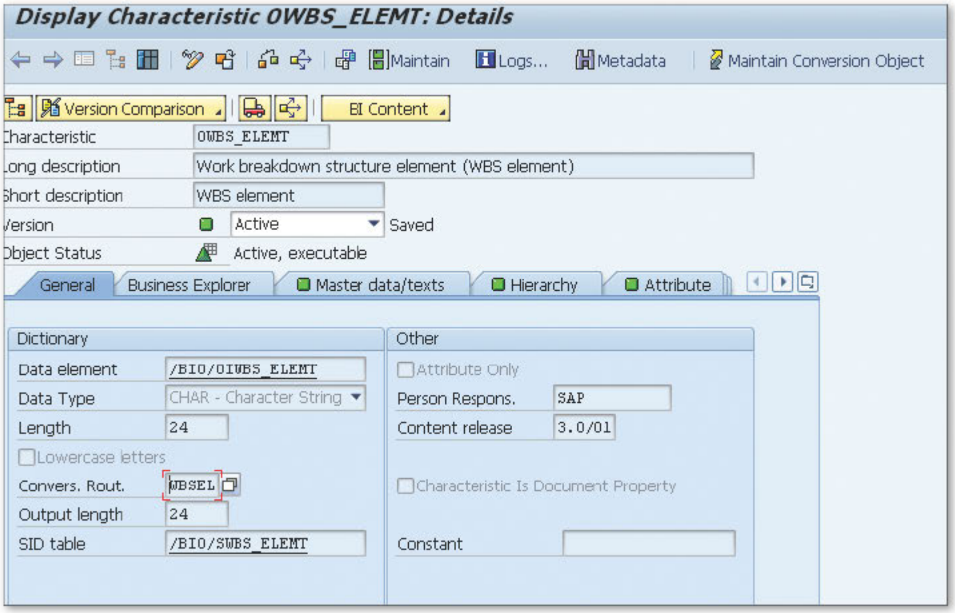
```
*----- intern -> extern; wbs_elemt -> wbs_elm_ex
  IF NOT INPUT IS INITIAL.
    GWA_BUF-WBS_ELEMT = INPUT.
    READ TABLE GT_BUF_WBS_ELEMT
         WITH KEY WBS_ELEMT = GWA_BUF-WBS_ELEMT
         INTO GWA_BUF
         BINARY SEARCH.
    IF SY-SUBRC IS INITIAL.
*----- internal number from buffer
      OUTPUT = GWA_BUF-WBS_ELM_EX.
    ELSE.
*----- select external number
        SELECT SINGLE WBS_ELEMT WBS_ELM_EX
               FROM   (LD_TABLENAME)
               INTO   CORRESPONDING FIELDS OF GWA_BUF
               WHERE  WBS_ELEMT    = GWA_BUF-WBS_ELEMT
               AND    OBJVERS      = 'A'
               AND    WBS_ELM_EX GT ''.
      IF SY-SUBRC IS INITIAL.
*----- output = external number
        OUTPUT = GWA_BUF-WBS_ELM_EX.
        if GWA_BUF-WBS_ELM_EX is initial.
           GWA_BUF-WBS_ELM_EX = input.
        endif.
```

Figure 3 Source Code Tab of Transaction SE37 Function Builder Screen

7. Test the function module to make sure the logic works correctly.

The next step is to assign the new custom conversion exit to the InfoObject (in this case, 0WBS_ELEMT). You can assign the new custom function module to the InfoObject from the InfoObject maintenance transaction (Transaction RSD1) by using the following steps (see Figure 4):

1. Maintain or edit the 0WBS_ELEMT InfoObject in Transaction RSD1.
2. From the GENERAL tab, remove the existing conversion routine WBSEL and add the new conversion routine ZWBSE.
3. Activate the InfoObject. Note that it's important to use the same five-character string that comes after "CONVERSION_EXIT_" from the custom function module name created in step 3 as the CONVERS. ROUT (conversion routine) name.



Characteristic	0WBS_ELEMT
Long description	Work breakdown structure element (WBS element)
Short description	WBS element
Version	Active Saved
Object Status	Active, executable

Data element	/BI0/OIWBS_ELEMT	Attribute Only	☐
Data Type	CHAR - Character String	Person Respons.	SAP
Length	24	Content release	3.0/01
Lowercase letters	☐		
Convers. Rout.	WBSEL	Characteristic Is Document Property	☐
Output length	24		
SID table	/BI0/SWBS_ELEMT	Constant	

Figure 4 General Tab of InfoObject Maintenance RSD1 Screen

With activation of the InfoObject 0WBS_ELEMT, any data display involving this InfoObject will use the custom routine to convert the internal database-stored format value to the desired external format. In this example, we use the conversion exit type OUTPUT that is raised when the InfoObject is displayed in a report. Similarly, you can code your own custom conversion routine for conversion exit type INPUT that is invoked during the data load. If you have the need to store the value of a particular InfoObject in a custom format, then you may use the conversion exit type INPUT to code that logic.

Tip 5

Finding the Lineage of an InfoObject in the Data Source

You can determine the lineage of an InfoObject to its source SAP ECC field to provide more options for data analysis.

Very often, we need more visibility to the mapping between SAP ECC source fields and SAP NetWeaver BW InfoObjects. Knowing the source SAP ECC fields enables the developer to further analyze the data based on the data element and domain of the source SAP ECC field. This is important for the new developers to make modeling decisions based on the lineage of the InfoObjects. And, this mapping is not readily available because the lineage of the InfoObjects becomes lost in the ETL process.

This tip describes a simple method to find out the lineage of an InfoObject.

✓ And Here's How ...

To determine an InfoObject's origins, follow these steps:

1. Go to Transaction SE11, provide the table name "RSOSFIELDMAP", and click the DISPLAY button.
2. Go to Transaction SE16N and press `Enter`.
3. On the resulting screen, enter the table name "RSOSFIELDMAP" (see Figure 1). The screen displays the selection criteria for the table. Two commonly used selection criteria are the DATASOURCE or the INFOOBJECT name.
4. In this case, we'll use a DataSource. Enter the name of the DataSource, "2LIS_05_QE1", which is mapped back to the InfoObject.

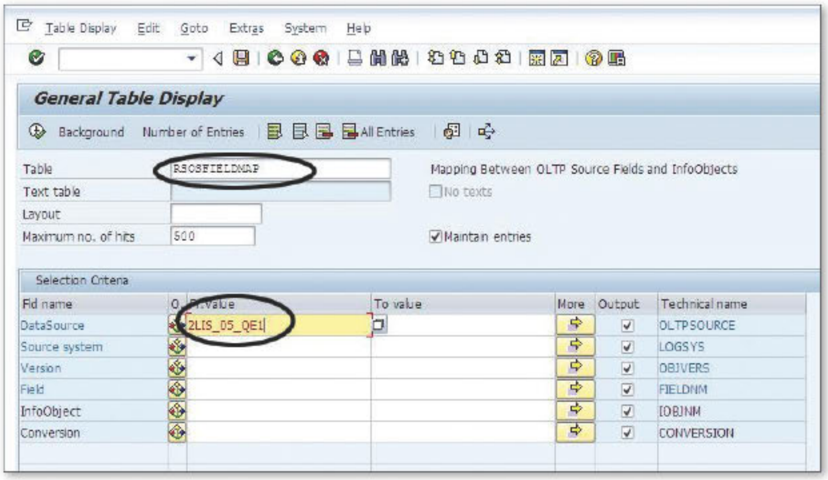


Table Display Edit Goto Extras System Help
General Table Display
Background Number of Entries All Entries

Table	RSOSFIELDMAP	Mapping Between OLTP Source Fields and InfoObjects
Text table		☐ No texts
Layout		
Maximum no. of hits	500	☑ Maintain entries

Selection Criteria

Fld name	O.	Fr.Value	To value	More	Output	Technical name
DataSource		2LIS_05_QE1			✓	OLTPSOURCE
Source system					✓	LOGSYS
Version					✓	OBJVERS
Field					✓	FIELDNM
InfoObject					✓	IOBJNM
Conversion					✓	CONVERSION

Figure 1 Transaction SE16N Screen for General Table Display

5. Click the EXECUTE button. The resulting screen in Figure 2 displays the mapping of all the DataSource fields and the corresponding InfoObject. For instance, you can see the standard INFOOBJECT 0MATERIAL and the corresponding DATASOURCE field MATNR.

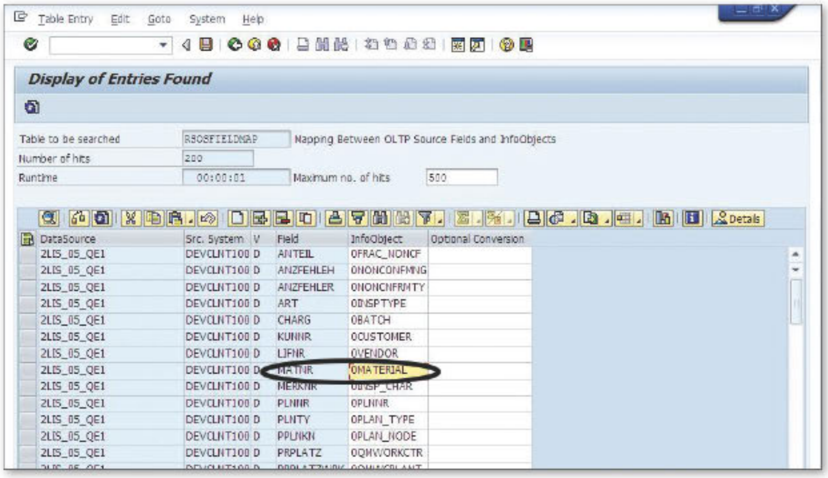


Table to be searched	RSOSFIELDMAP	Mapping Between OLTP Source Fields and InfoObjects		
Number of hits	200			
Runtime	00:00:01	Maximum no. of hits	500	

DataSource	Src. System	V	Field	InfoObject	Optional Conversion
2LIS_05_QE1	DEVCLNT100	D	ANTEIL	0FRAC_NONCF	
2LIS_05_QE1	DEVCLNT100	D	ANZFEHLEH	0NONCONFMNG	
2LIS_05_QE1	DEVCLNT100	D	ANZFEHLER	0NONCNFRMTY	
2LIS_05_QE1	DEVCLNT100	D	ART	0INSPTYPE	
2LIS_05_QE1	DEVCLNT100	D	CHARG	0BATCH	
2LIS_05_QE1	DEVCLNT100	D	KUNNR	0CUSTOMER	
2LIS_05_QE1	DEVCLNT100	D	LIFNR	0VENDOR	
2LIS_05_QE1	DEVCLNT100	D	MATNR	0MATERIAL	
2LIS_05_QE1	DEVCLNT100	D	MERKNR	0INSP_CHAR	
2LIS_05_QE1	DEVCLNT100	D	PLNNR	0PLNNR	
2LIS_05_QE1	DEVCLNT100	D	PLNTY	0PLAN_TYPE	
2LIS_05_QE1	DEVCLNT100	D	PPLNKN	0PLAN_NODE	
2LIS_05_QE1	DEVCLNT100	D	PRPLATZ	0QMWORKCTR	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	

Figure 2 Display of Entries for Transaction SE16N for Table RSOSFIELDMAP

This method can be used to get information on the lineage of the InfoObjects. This information can be used to further analyze the data element and the data on the source SAP ECC side and will help make data modeling decisions based on that.

Tip 6

Looking Up a DataStore Object with a New Rule Type in SAP NetWeaver BW 7.3

You can use new rule types in the SAP NetWeaver BW 7.3 transformations to look up DSO data without coding in START/END routines.

In a typical layered scalable architecture modeling approach, business rules transformations are implemented in the level 2 DSO layer. This requires lookups on other DSOs to transform the data. Currently, the method used for implementing lookups on DSOs is to code the lookup logic in the `START/END` routines.

This tip shows you how to use a new rule type within SAP NetWeaver BW 7.3 to look up a DSO without ABAP coding in the transformation. The benefit of this new rule type is that the logic for looking up a DSO can be accomplished without coding ABAP, which reduces the coding time and complexity of the models.

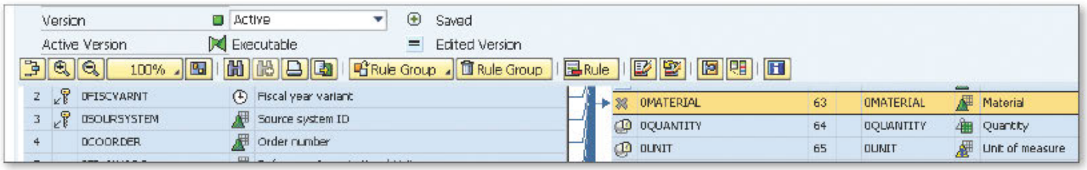
✓ And Here's How ...

The key prerequisite to implementing this functionality is that the source structure should contain the primary keys of the lookup DSO.

In the example in this tip, the InfoObject *Material* is derived from a DSO with three primary keys (Record No, Fiscal Year Variant, and Source System ID). The source structure contains these primary key fields, and the target structure contains the InfoObject Material.

Follow these steps:

1. In the SAP NetWeaver BW Administrator Workbench, open up the transformation where the lookup logic needs to be implemented, and locate the InfoObject 0MATERIAL in the target structure as shown in Figure 1.



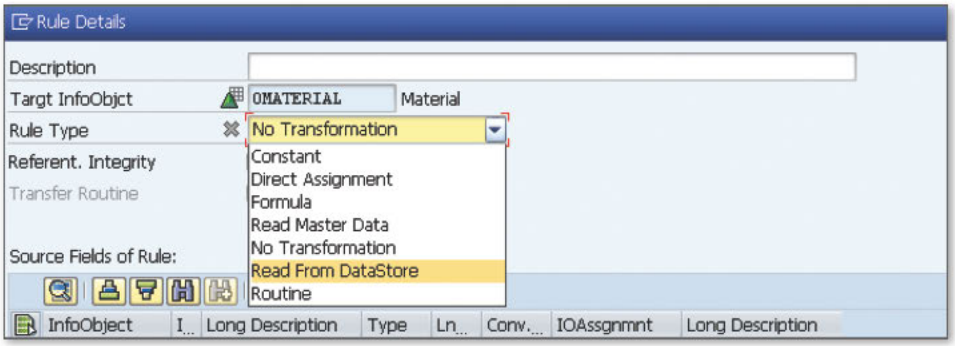
Version	Active
Active Version	Executable
Saved	Edited Version

2	0FISCVARNT	Fiscal year variant	
3	0SOURSYSTEM	Source system ID	
4	0COORDER	Order number	

0MATERIAL	63	0MATERIAL	Material
0QUANTITY	64	0QUANTITY	Quantity
0UNIT	65	0UNIT	Unit of measure

Figure 1 Transformation Screen

2. Right-click on 0MATERIAL from the target structure and choose Rule Details in change mode.
3. The Rule Type dropdown box shows various rule types that are available for use. Select Read from DataStore (see Figure 2). This is a new rule type that can be used for looking up a DSO.



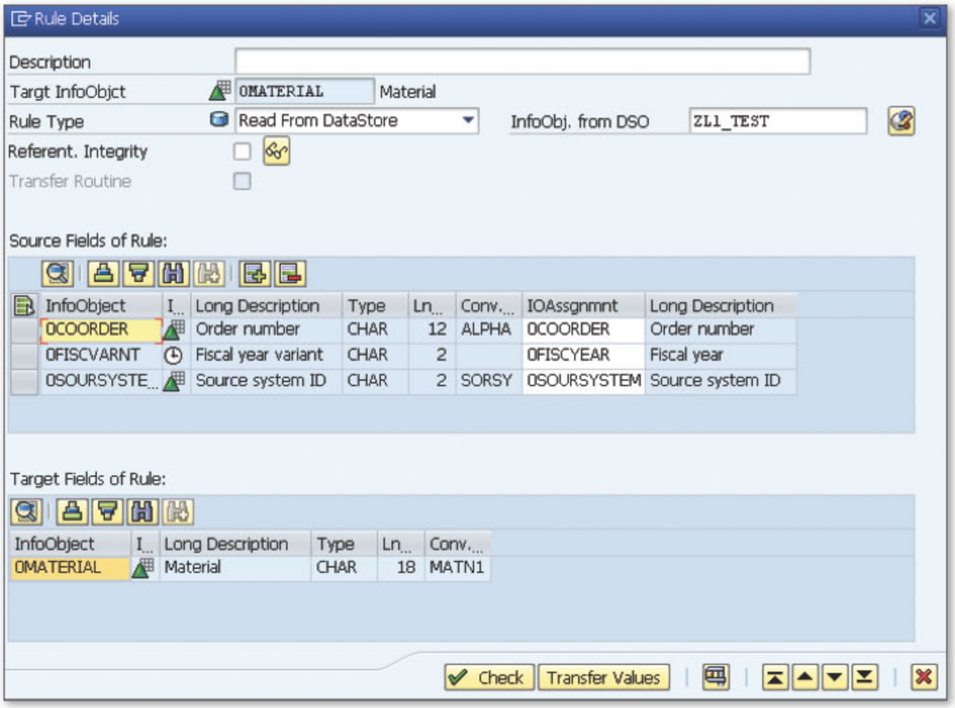
Rule Details

Description	
Targt InfoObjct	0MATERIAL Material
Rule Type	No Transformation
Referent. Integrity	
Transfer Routine	
Source Fields of Rule:	

InfoObject	I...	Long Description	Type	Ln...	Conv....	IOAssgnmnt	Long Description

Figure 2 Rule Details Displaying Various Rule Types

4. The resulting screen shows an option to select the lookup DSO. Using the dropdown box, select the DSO for lookup.
5. Press Enter, and the system shows the primary key objects of the lookup DSO. Map the lookup DSO primary key to the respective source structure fields. The mapping can be done by entering the source structure fields in the various columns in Figure 3.


Rule Details

Description			
Targt InfoObjct	0MATERIAL Material		
Rule Type	Read From DataStore	InfoObj. from DSO	ZL1_TEST
Referent. Integrity	☐		
Transfer Routine	☐		

Source Fields of Rule:

InfoObject	I...	Long Description	Type	Ln...	Conv....	IOAssgnmnt	Long Description
0COORDER		Order number	CHAR	12	ALPHA	0COORDER	Order number
0FISCVARNT		Fiscal year variant	CHAR	2		0FISCYEAR	Fiscal year
0SOURSYSTE...		Source system ID	CHAR	2	SORSY	0SOURSYSTEM	Source system ID

Target Fields of Rule:

InfoObject	I...	Long Description	Type	Ln...	Conv....
0MATERIAL		Material	CHAR	18	MATN1

Figure 3 Rule Details Screen to Provide the Mapping of DSO InfoObjects to the Source Structure Fields

6. Click on the Transfer Values button, and the resulting screen will provide options for error handling.
7. You can now configure what the system will do if no values were found in the DataStore. Your options are to provide an error message or constant.

Now, the InfoObject 0MATERIAL be derived in the transformations by doing a lookup on the DSO. As long as the source structure fields match with the primary fields of the DSO, the transformation will automatically look up the corresponding record in the DSO for the InfoObject value. This eliminates any need for additional coding to look up an InfoObject from a DSO.

Tip 7

Combining the Results of Two Queries Using the Application Process Designer

You can use the Application Process Designer to combine results from two or more queries into one result set for further transformation when you're modeling data.

In some modeling scenarios, you may need to combine results from two or more queries into one dataset so that further transformations can be done on the resulting dataset. This scenario is mainly applicable when you want to leverage the query modeling capabilities such as restricted key figures or variable exits to model the data.

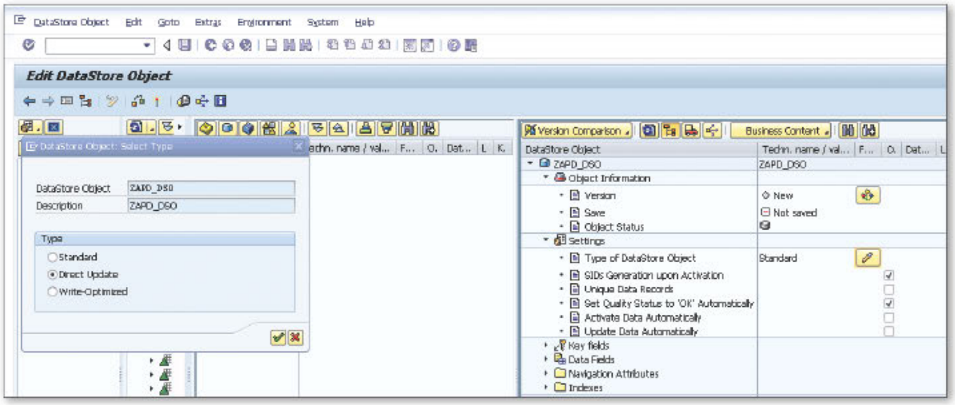
You can accomplish this by storing the query results to DSOs and then applying further transformations on the DSO; however, this is a cumbersome process and not easy to model or maintain.

This tip describes how you can use the Application Process Designer (APD) to combine the results of queries in a seamless manner and also implement transformation logic on the result set.

✓ And Here's How ...

To describe this tip, we'll use a scenario where we join the results of two BEx queries into a DSO. So in this case, the two queries will be the source, and we'll describe the process for combing the results of the two queries into the target DSO.

1. Create two BEx queries using the BEx Query Designer: TEST_QUERY_1 and TEST_QUERY_2 on InfoCube 0SD_C03.
2. Go to Transaction RSA1 and create a direct update DSO. During the initial creation of the DSO, the system will prompt you to select the DSO type, so select DIRECT UPDATE as shown in Figure 1. Make sure to include all of the key fields and data fields into the DSO per the design of the source BEx queries.



DataStore Object: ZAPD_DSO
Description: ZAPD_DSO
Type: Standard [✓ Direct Update] Write-Optimized

DataStore Object	Techn. name / val...
ZAPD_DSO	ZAPD_DSO
Object Information	
Version	New
Save	Not saved
Object Status	
Settings	
Type of DataStore Object	Standard
SIDs Generation upon Activation	✓
Unique Data Records	☐
Set Quality Status to 'OK' Automatically	✓
Activate Data Automatically	☐
Update Data Automatically	☐
Key fields	
Data Fields	
Navigation Attributes	
Indexes	

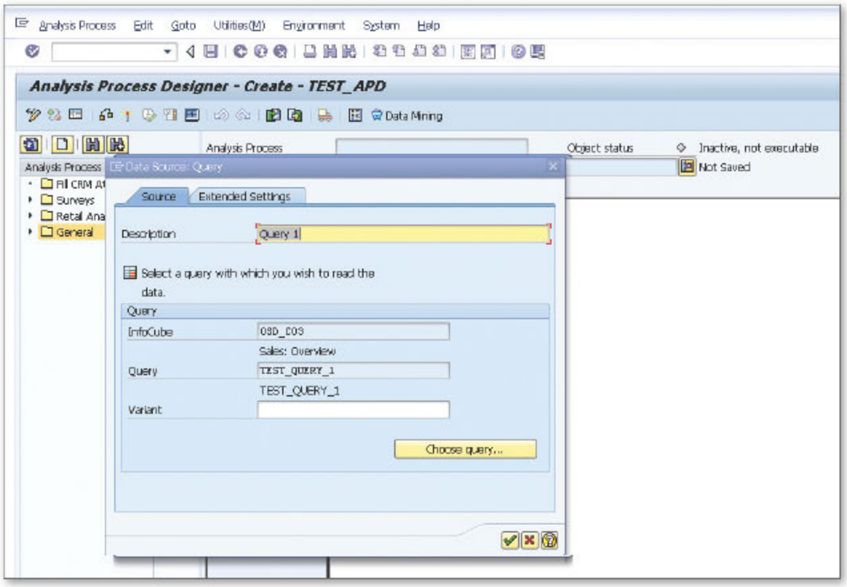
Figure 1 Direct Access DSO in Transaction RSA1

3. Now that you have the two source BEx queries and the target DSO created, use the APD to combine the two queries into the DSO:
   - Open the APD using Transaction RSANWB
   - Right-click GENERAL in the ANALYSIS PROCESS pane, and choose CREATE ANALYSIS PROCESS.

In the CREATE ANALYSIS PROCESS screen, there are three main options. With the DATASOURCES option, you can select the source of the APD. In this case, the sources are the BEx queries that we created earlier. With the TRANSFORMATIONS option, you can define transformations for the APD. With the DATA TARGETS option, you can define the target of the APD, which in this case is the DSO that we created earlier.

Follow these steps to combine the two queries using APD:

1. Under the DATASOURCES option, click on the QUERY icon to open the DATA SOURCE: QUERY screen (see Figure 2). In this screen, select QUERY as the DataSource, and input the name of the first query. You have to specify the INFOCUBE name first, that is, "0SD_C03".



Analysis Process Designer - Create - TEST_APD

Analysis Process

Object status: Inactive, not executable

Not Saved

Data Source: Query

Description: Query 1

Select a query with which you wish to read the data.

Query

InfoCube: 0SD_C03

Sales: Overview

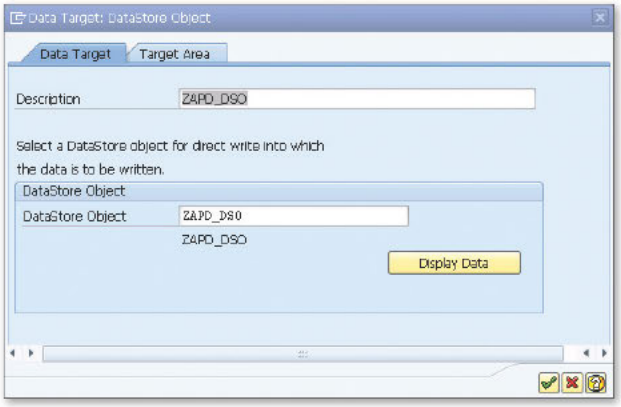
Query: TEST_QUERY_1

TEST_QUERY_1

Variant:

Figure 2 Selecting the First Query

2. Using the same DataSources option, select the second query as DataSource, and give the name of the second query.
3. From the Data Targets option, click on the icon for DSO. Enter the name of the DSO created in an earlier step as the data target (see Figure 3).



Data Target: DataStore Object

Description: ZAPD_DSO

Select a DataStore object for direct write into which the data is to be written.

DataStore Object

DataStore Object: ZAPD_DSO

ZAPD_DSO

Figure 3 Specifying a Target for the APD

4. Merge the two queries by selecting Union operator from the Transformations option as shown in Figure 4. Use your mouse to click on the query, and then you can extend the arrow line to the Union operator.

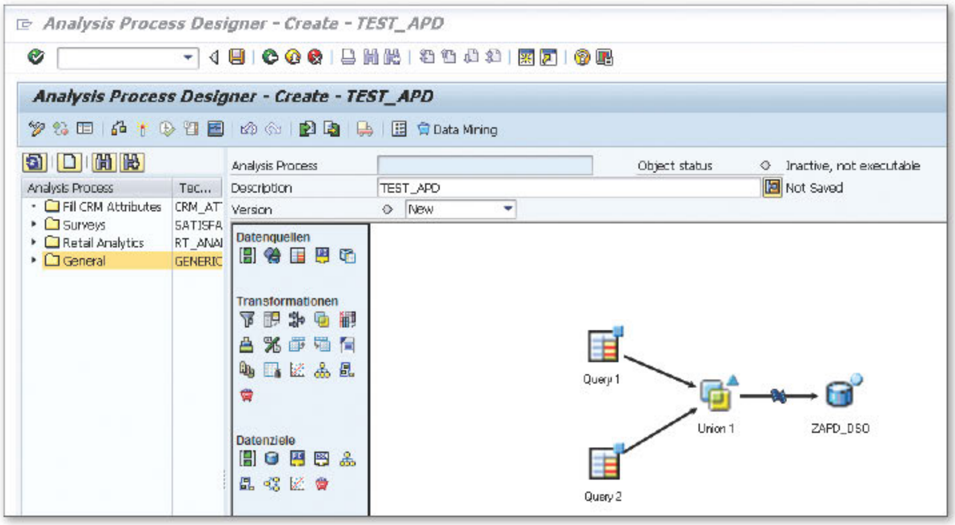


Figure 4 Union of Two Queries

With these steps, you have implemented a basic UNION operation to combine two queries into a target DSO. Similarly, you can use any of the TRANSFORMATION options to transform the data.

Tip 8

Using the Analysis Process Designer as a Data Modeling Object

You can use the APD as a data modeling object to implement complex modeling scenarios in a more efficient manner and easily establish data relationships.

Sometimes, business reporting requirements lead to complex modeling scenarios in SAP NetWeaver BW, and it becomes important to evaluate the different modeling approaches available to determine the most efficient modeling technique. Typically, modeling some of these complex scenarios would require creating multiple layers of DSOs with data persistency and/or writing ABAP code in the transformations to decode the complex relationships within the data. These approaches could be very difficult to implement and support and are also not scalable.

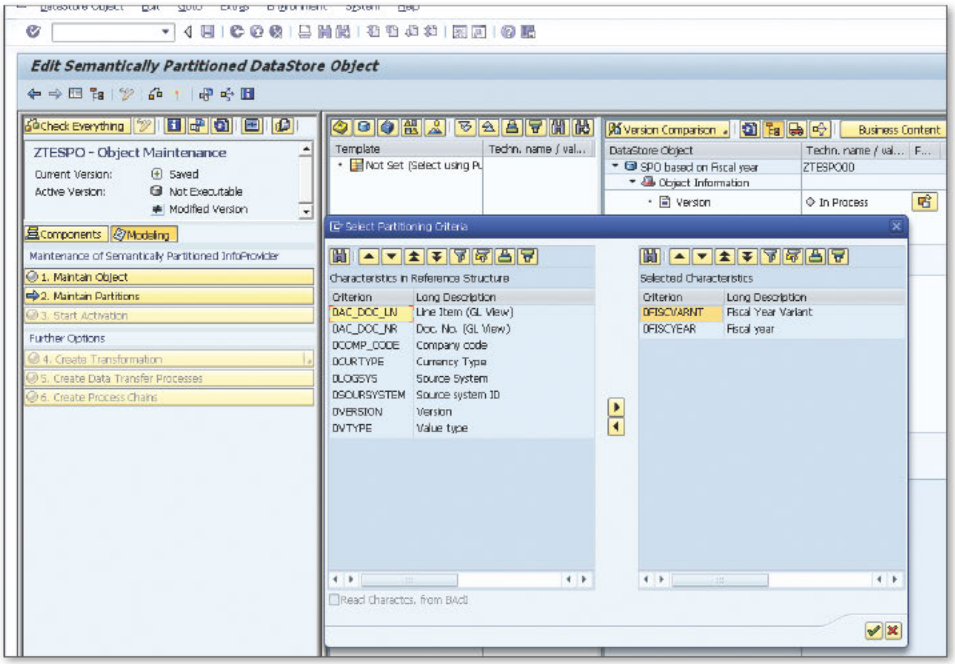
To avoid this issue, you can use the Analysis Process Designer (APD), which is a very efficient data modeling object because it helps to identify the complex relationships between data in a simple way. This tip uses a modeling scenario of converting an account model to a key figure model and describes how the APD can be used as a data modeling object to model these kinds of scenarios.

✓ And Here's How ...

To describe the tip, we'll use an existing InfoCube with an account-based model. The InfoCube shows project budget data, and the budget amount for each fiscal period is stored in a separate record so there will be one record for each fiscal period (month). We will use APD to convert this model to a key figure model where the budget amount for the year is shown on a single record. This tip doesn't cover the basic steps of creating an APD, so we use an existing test demo APD for this purpose.

To use the APD, follow these steps:

1. Go to Transaction RSANWB in the SAP NetWeaver BW system. Right-click on GENERAL in the ANALYSIS PROCESS column, and choose CREATE.
2. Select the fields for which you want to enter filter conditions in the FILTER SELECTION tab, and provide the name of the filter conditions in the FILTER CONDITIONS tab.
3. In the next screen, use drag and drop to join WBS master data and cube data as shown in Figure 1.


Edit Semantically Partitioned DataStore Object

ZTESPO - Object Maintenance

Current Version: Saved
Active Version: Not Executable
Modified Version

Maintenance of Semantically Partitioned InfoProvider
1. Maintain Object
2. Maintain Partitions
3. Start Activation

Further Options
4. Create Transformation
5. Create Data Transfer Processes
6. Create Process Chains

DataStore Object	Techn. name / val...
SPO based on Fiscal year	ZTESPO00
Object Information	
Version	In Process

Select Partitioning Criteria

Characteristics in Reference Structure

Criterion	Long Description
0AC_DOC_LN	Line Item (GL View)
0AC_DOC_NR	Doc. No. (GL View)
0COMP_CODE	Company code
0CURTYPE	Currency Type
0LOGSYS	Source System
0SOURSYSTEM	Source system ID
0VERSION	Version
0VTYPE	Value type

Selected Characteristics

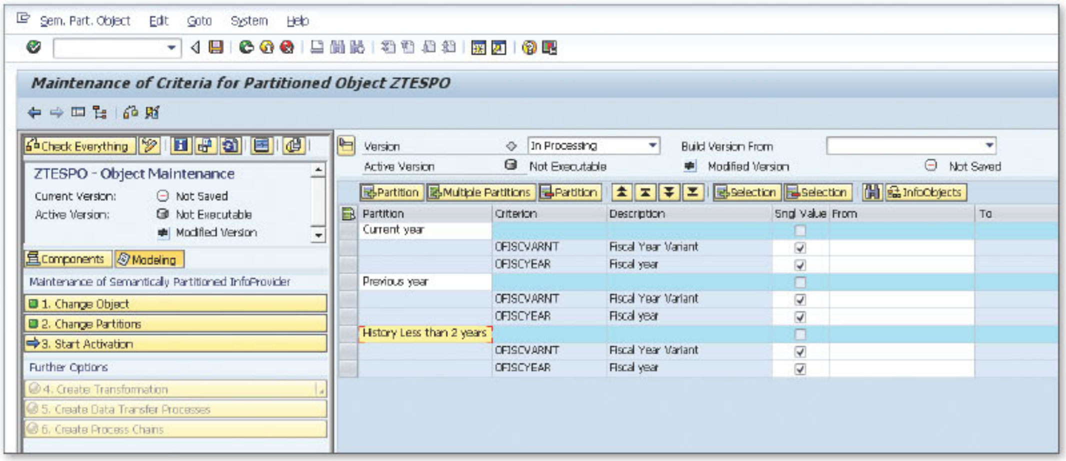
Criterion	Long Description
0FISCVARNT	Fiscal Year Variant
0FISCYEAR	Fiscal year

Read Charactcs. from BAdI

Figure 1 Creation of the Analysis Process Designer

4. Click on the AGGREGATION icon on the screen. Select the fields to be used in the formation of groups and the fields for which the values are to be aggregated.
5. Select the fields that can be used for conversion of the account model to the key figure model in the DETAILS tab of LIST-->DATA RECORD TRANSFORM as shown

in Figure 2. In this case, we will select Fiscal Period and Amount as the fields that will be used to derive the key figure model.



Maintenance of Criteria for Partitioned Object ZTESPO

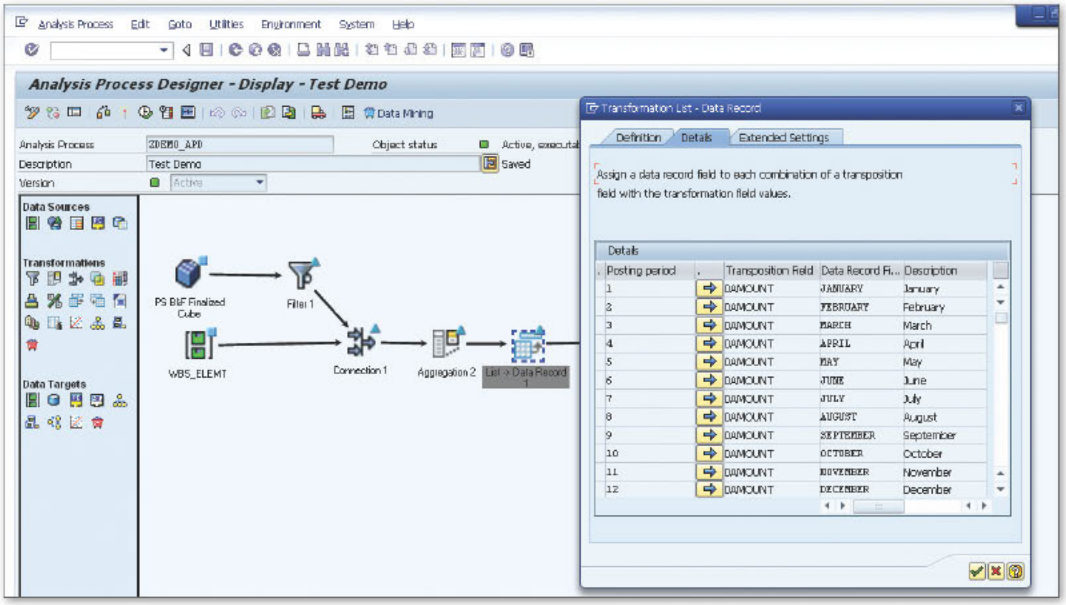
ZTESPO - Object Maintenance
Current Version: Not Saved
Active Version: Not Executable
Modified Version

Version: In Processing
Active Version: Not Executable
Build Version From:
Modified Version
Not Saved

Partition	Criterion	Description	Sngl Value	From	To
Current year			☐		
	0FISCVARNT	Fiscal Year Variant	✓		
	0FISCYEAR	Fiscal year	✓		
Previous year			☐		
	0FISCVARNT	Fiscal Year Variant	✓		
	0FISCYEAR	Fiscal year	✓		
History Less than 2 years			☐		
	0FISCVARNT	Fiscal Year Variant	✓		
	0FISCYEAR	Fiscal year	✓		

Figure 2 Defining the Transformation of the Data Record

6. In the Details tab, assign a Data Record Field to each combination of a Transposition Field with the transformation field values as shown in Figure 3.



Analysis Process Designer - Display - Test Demo

Analysis Process: ZDEMO_APD
Description: Test Demo
Version: Active
Object status: Active, executable
Saved

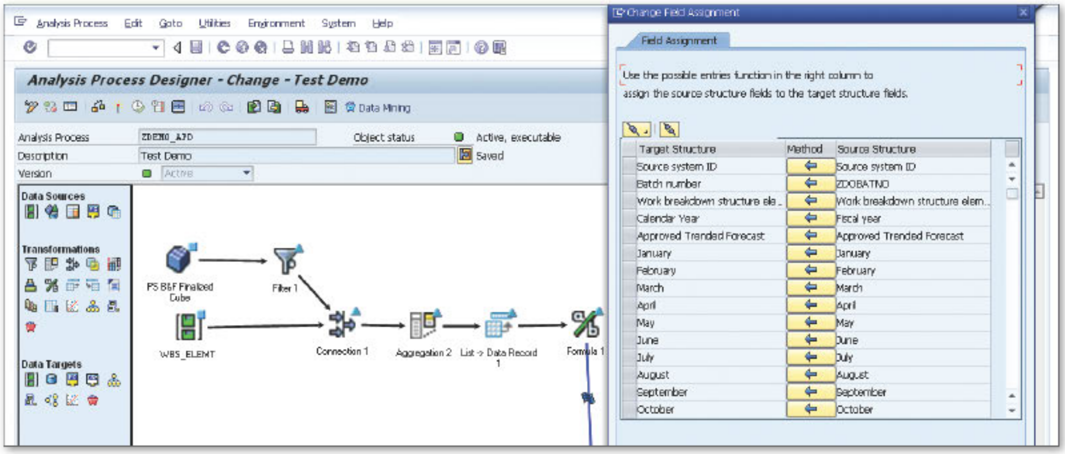
Assign a data record field to each combination of a transposition field with the transformation field values.

Details

Posting period		Transposition Field	Data Record Fi...	Description
1	➡	DAMOUNT	JANUARY	January
2	➡	DAMOUNT	FEBRUARY	February
3	➡	DAMOUNT	MARCH	March
4	➡	DAMOUNT	APRIL	April
5	➡	DAMOUNT	MAY	May
6	➡	DAMOUNT	JUNE	June
7	➡	DAMOUNT	JULY	July
8	➡	DAMOUNT	AUGUST	August
9	➡	DAMOUNT	SEPTEMBER	September
10	➡	DAMOUNT	OCTOBER	October
11	➡	DAMOUNT	NOVEMBER	November
12	➡	DAMOUNT	DECEMBER	December

Figure 3 Defining the Transformation Fields: Amount for Each Fiscal Period

7. Assign the Source Structure fields to the Target Structure fields in the transformation (see Figure 4).



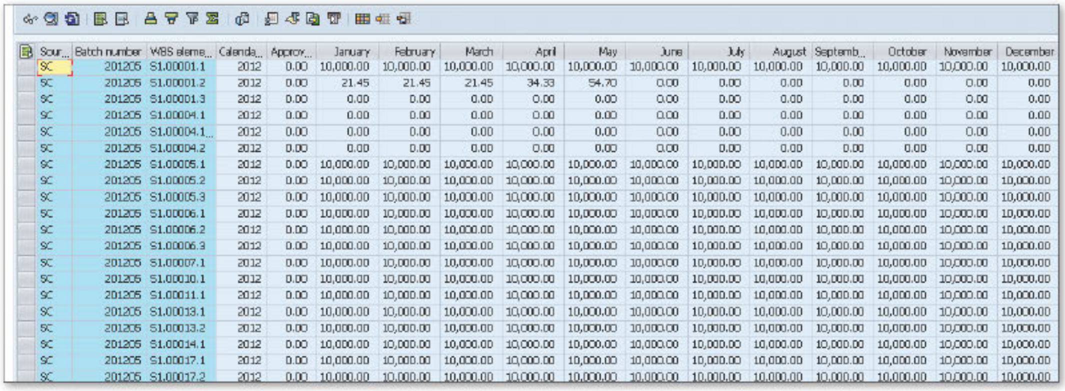
Analysis Process	ZDEMO_APD
Description	Test Demo
Version	Active
Object status	Active, executable
	Saved

Use the possible entries function in the right column to assign the source structure fields to the target structure fields.

Target Structure	Method	Source Structure
Source system ID	⇦	Source system ID
Batch number	⇦	ZDOBATNO
Work breakdown structure ele...	⇦	Work breakdown structure elem...
Calendar Year	⇦	Fiscal year
Approved Trended Forecast	⇦	Approved Trended Forecast
January	⇦	January
February	⇦	February
March	⇦	March
April	⇦	April
May	⇦	May
June	⇦	June
July	⇦	July
August	⇦	August
September	⇦	September
October	⇦	October

Figure 4 Assigning the Amount to the Transformation Fields

When the APD job is executed, data in the data target is displayed as shown in Figure 5.



Sour...	Batch number	WBS eleme...	Calenda...	Approv...	January	February	March	April	May	June	July	August	Septemb...	October	November	December
SC	201205	S1.00001.1	2012	0.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00
SC	201205	S1.00001.2	2012	0.00	21.45	21.45	21.45	34.33	54.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SC	201205	S1.00001.3	2012	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SC	201205	S1.00004.1	2012	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SC	201205	S1.00004.1...	2012	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SC	201205	S1.00004.2	2012	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SC	201205	S1.00005.1	2012	0.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00
SC	201205	S1.00005.2	2012	0.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00
SC	201205	S1.00005.3	2012	0.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00
SC	201205	S1.00006.1	2012	0.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00
SC	201205	S1.00006.2	2012	0.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00
SC	201205	S1.00006.3	2012	0.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00
SC	201205	S1.00007.1	2012	0.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00
SC	201205	S1.00010.1	2012	0.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00
SC	201205	S1.00011.1	2012	0.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00
SC	201205	S1.00013.1	2012	0.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00
SC	201205	S1.00013.2	2012	0.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00
SC	201205	S1.00014.1	2012	0.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00
SC	201205	S1.00017.1	2012	0.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00
SC	201205	S1.00017.2	2012	0.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00	10,000.00

Figure 5 Results of the APD Job

Tip 9

Using the Analytic Index as a Data Provider to Create Quick Data Prototypes

You can use the analytic index as a data provider with data saved in SAP HANA databases for quick and flexible reporting and prototyping on which you can also build queries.

One common requirement for SAP NetWeaver BW users is to create quick prototypes of data and develop ad hoc reporting scenarios. However, the SAP NetWeaver BW modeling environment requires a lot of design and development effort, making it difficult to develop prototyping scenarios.

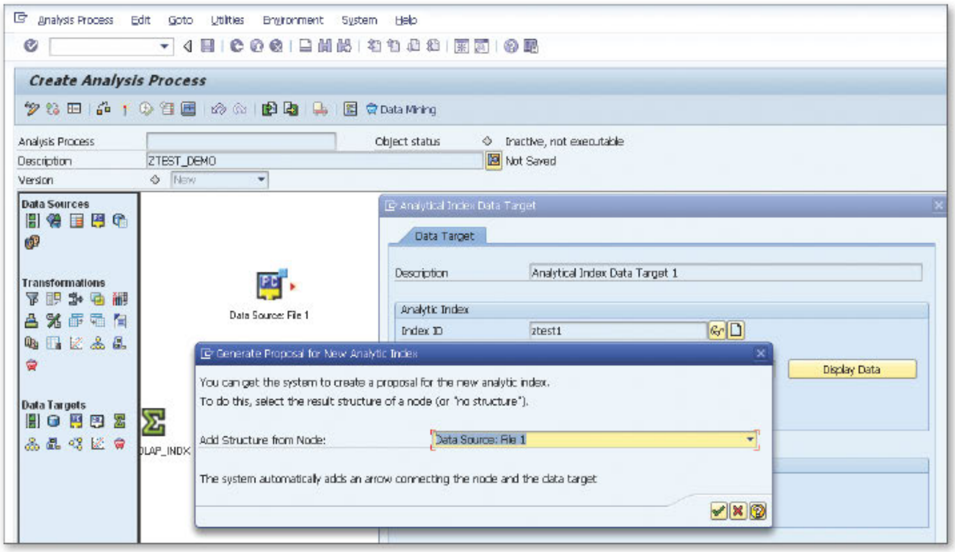
To help with this issue, users can use the analytic index to perform rapid prototyping and easily create ad hoc scenarios for SAP NetWeaver BW systems using an SAP HANA database or an SAP NetWeaver BW Accelerator (BWA). Analytic indexes can be created and filled with (transformed) data quickly, and BEx queries can be created for reporting.

This tip describes a new functionality that is available with SAP NetWeaver BW 7.3 where you can use analytic index as a data modeling object and a data container to model ad hoc reporting scenarios.

And Here's How ...

To use the analytic index, follow these steps:

1. Go to Transaction RSANWB, and create an analytic process by right-clicking on GENERAL, and choosing CREATE ANALYSIS PROCESS.
2. In the resulting SOURCE FILE popup screen, go to the DATA SOURCE tab, choose the DATA SOURCE to read from a flat file and, provide the path of the file in the FILE FORMAT and FILE fields. Also, select the DATA TARGET as ANALYTIC INDEX.
3. You'll now see the ANALYTICAL INDEX DATA TARGET popup screen, where you enter the index ID and click on the CREATE button. The window shown in Figure 1 opens. Click CONTINUE to have the system generate a proposal for the new analytic index.



Create Analysis Process

Analysis Process
Description ZTEST_DEMO
Version New
Object status Inactive, not executable
Not Saved

Analytical Index Data Target

Data Target
Description Analytical Index Data Target 1
Analytic Index
Index ID ztest1

Generate Proposal for New Analytic Index

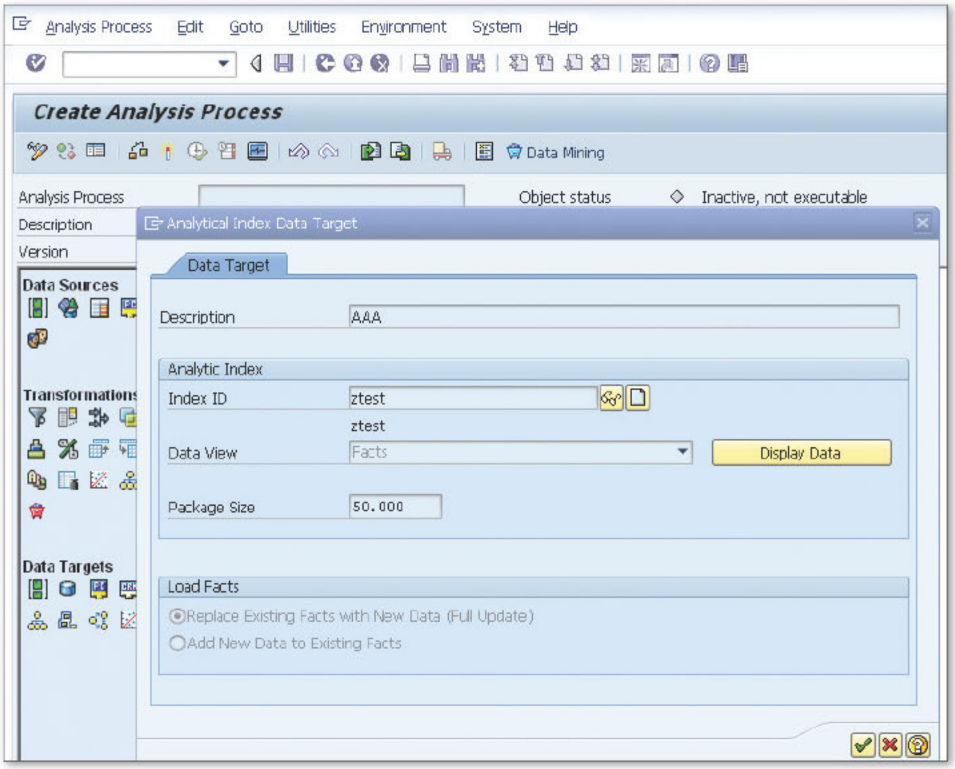
You can get the system to create a proposal for the new analytic index.
To do this, select the result structure of a node (or "no structure").

Add Structure from Node: Data Source: File 1

The system automatically adds an arrow connecting the node and the data target

Figure 1 Generate Proposal

4. Upon generating the proposal, the system will automatically generate and populate dimensions and key figures if it finds the appropriate InfoObjects. You can use these or fill in the dimensions and key figures you want represented.
5. After you have the appropriate dimensions and key figures entered, click the ACTIVATE button at the top of the screen, and click on the BACK button. You'll now see the screen shown in Figure 2.



Analysis Process	
Description	
Version	
Object status	Inactive, not executable

Analytical Index Data Target

Data Target

Description	AAA

Analytic Index

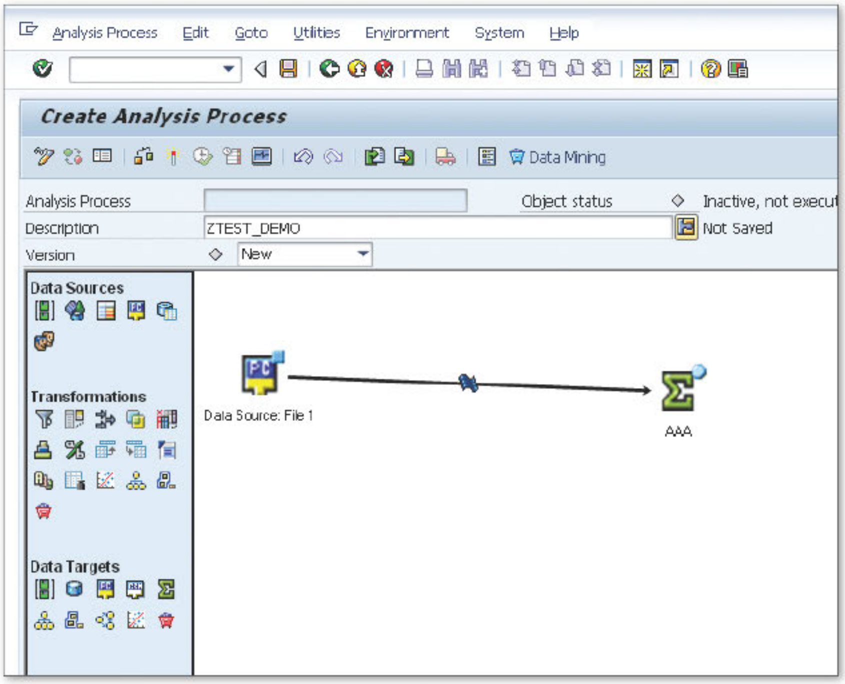
Index ID	ztest
	ztest
Data View	Facts
Package Size	50.000

Load Facts

(● Replace Existing Facts with New Data (Full Update)) ○ Add New Data to Existing Facts

Figure 2 Defining the Data Target for the Analytical Index

6. Click the CONTINUE button, and a popup appears asking if the system should make a proposal for the field assignment. Click YES.
7. Save and activate the APD. Now when you execute the APD, data from the flat file is loaded into the analytic index (see Figure 3).
8. The created analytic index will be available as an InfoProvider in the BEx Query Designer (see Figure 4). Queries can be built on top of it.



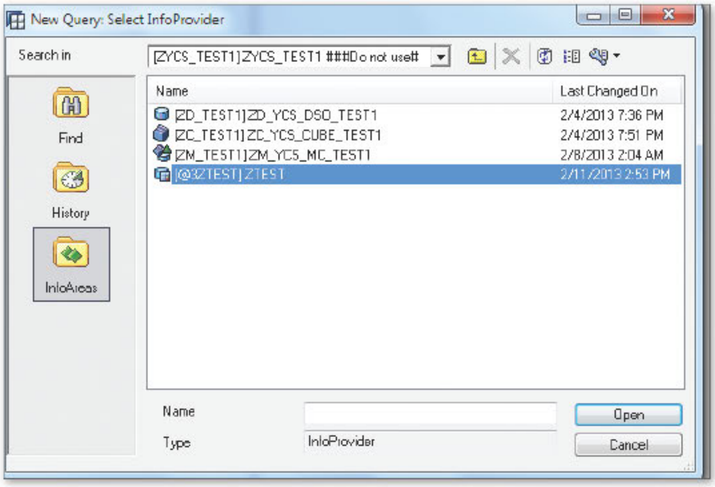
Analysis Process

Object status ◇ Inactive, not execut

Description ZTEST_DEMO Not Saved

Version ◇ New

Figure 3 The Activated Analysis Process Displaying Source and Analytical Index as Target



Search in [ZYCS_TEST1]ZYCS_TEST1 ###Do not use##

Name	Last Changed On
[ZD_TEST1]ZD_YCS_DSO_TEST1	2/4/2013 7:36 PM
[ZC_TEST1]ZC_YCS_CUBE_TEST1	2/4/2013 7:51 PM
[ZM_TEST1]ZM_YCS_MC_TEST1	2/8/2013 2:04 AM
[@3ZTEST]ZTEST	2/11/2013 2:53 PM

Name

Type InfoProvider

Figure 4 Analytical Index Available as an InfoProvider in BEx Query Designer

Tip 10

Converting Standard InfoCubes to SAP HANA Optimized InfoCubes

After migrating SAP NetWeaver BW to SAP HANA, you can reap all of SAP HANA's performance benefits by converting standard InfoCubes to SAP HANA optimized InfoCubes.

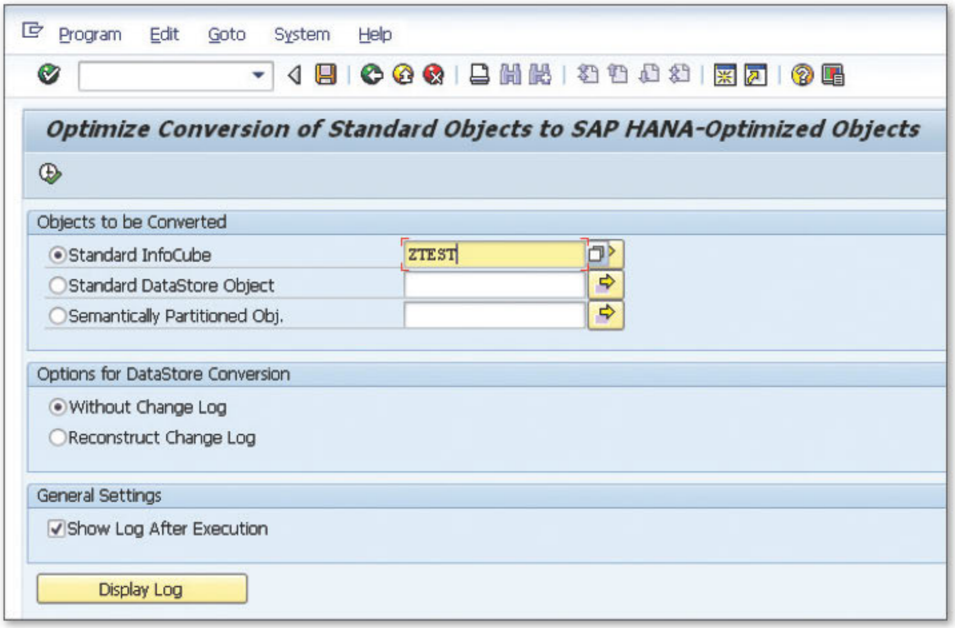
When the SAP NetWeaver BW system is migrated to an SAP HANA database, all the InfoCubes are not automatically converted to SAP HANA optimized InfoCubes. The conversion of standard InfoCubes to SAP HANA optimized InfoCubes ensures that the cubes are set up most efficiently for SAP HANA and able to leverage the benefits of the SAP HANA in-memory optimizer. If you're using an SAP HANA database, then it's highly recommended to convert the existing standard InfoCubes to SAP HANA optimized InfoCubes to get the performance benefits for data loading and reporting.

This tip describes how to convert a standard InfoCube into an SAP HANA optimized InfoCube.

✓ And Here's How ...

After the migration of an SAP NetWeaver BW platform to an SAP HANA database, the standard InfoCubes can be found in the SAP HANA database column store and can be used for reporting without any restrictions. However, to take full advantage of the power of the SAP HANA database, the standard InfoCubes can be converted to SAP HANA optimized InfoCubes using the following simple steps:

1. Login to the SAP NetWeaver BW system.
2. Call up Transaction RSMIGRHANADB or the program RSDRI_CONVERT_CUBE_TO_INMEMORY.
3. In the subsequent screen, select the standard InfoCube that needs to be converted, and click the EXECUTE button (see Figure 1).



Optimize Conversion of Standard Objects to SAP HANA-Optimized Objects

Objects to be Converted

● Standard InfoCube	ZTEST
○ Standard DataStore Object	
○ Semantically Partitioned Obj.	

Options for DataStore Conversion

● Without Change Log
○ Reconstruct Change Log

General Settings

☑ Show Log After Execution

Figure 1 Entering the Cube Name

The job is executed in the background as a stored procedure. After the job is finished, the standard InfoCube is converted to an SAP HANA optimized InfoCube.

After the conversion, the InfoCube dimension tables are removed, and the master data tables are now directly linked with the F-fact table.

Tip 11

Modeling InfoProviders with the Composite Provider

You can avoid data redundancy or extensive coding when modeling data relations with the use of the composite provider and joins.

In SAP NetWeaver BW environments today, relations between data sets are modeled in SAP NetWeaver BW transformation routines and loaded to a DSO, or by updating (overwrite) a target DSO. This typically leads to scenarios of architected data marts where multiple DSOs load into a single InfoProvider with moderate to extensive coding are required to support these scenarios. The major drawback with this modeling is the data redundancy and overhead in developing and supporting the additional objects required to model the relations between data.

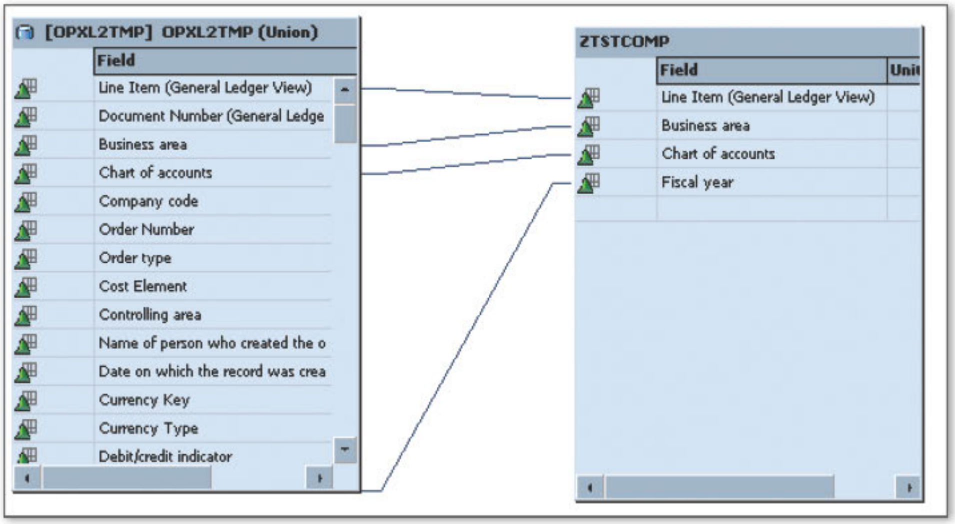
Composite Providers available with SAP NetWeaver BW 7.3 provide more flexibility in InfoProvider modeling in SAP NetWeaver BW and can be a great tool for virtualization of architected data marts. This tip shows you how to use Composite Providers in SAP NetWeaver BW and also describes some of the options for combining data from base InfoProviders (InfoCubes, DSOs, master data attributes, etc.).

✓ And Here's How ...

An SAP HANA database or BWA (BW Accelerator) is a prerequisite for using Composite Providers.

We'll now go through the step-by-step instructions needed to model a Composite Provider:

1. Execute Transaction RSLIMOBW in the SAP NetWeaver BW system.
2. Enter the name of the Composite Provider and click the CREATE button.
3. Select the InfoProvider from the INFO AREAS TREE on the left side of the screen. Drag and drop it to the right side.
4. You'll see a popup screen appears with an option to choose either UNION or JOIN as the BINDING TYPE:
   - UNION
     This operation is a data union of basic InfoProviders (InfoCube, DSO, etc.), so you may choose this option if you want to combine data without using common fields as a condition for the relation.
   - JOIN
     This operation is an intersection of the base InfoProviders and builds a data join of basic InfoProviders based on a join condition of common fields among the base InfoProviders.
   - Select UNION, and click CONTINUE.
5. Select the fields from the InfoProvider, and drag and drop to the Composite Provider as shown in Figure 1.



[OPXL2TMP] OPXL2TMP (Union)

Field
Line Item (General Ledger View)
Document Number (General Ledge
Business area
Chart of accounts
Company code
Order Number
Order type
Cost Element
Controlling area
Name of person who created the o
Date on which the record was crea
Currency Key
Currency Type
Debit/credit indicator

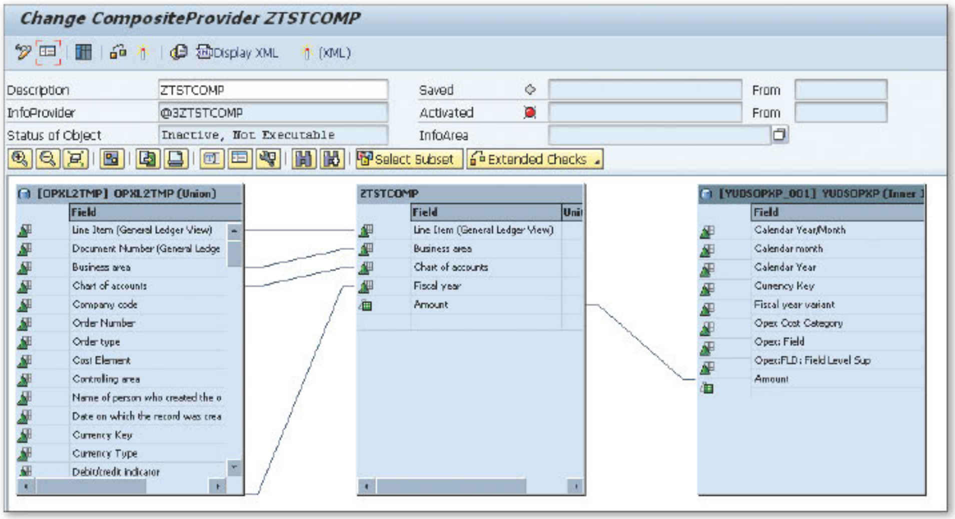
ZTSTCOMP

Field	Unit
Line Item (General Ledger View)	
Business area	
Chart of accounts	
Fiscal year	

Figure 1 Selecting Fields from the InfoProvider

6. Select the second InfoProvider to make the left outer join.

7. In the INSERT OBJECT popup, select JOIN from the BINDING TYPE dropdown list and click CONTINUE.
8. Select the fields from the InfoProvider, and add to the Composite Provider as shown in Figure 2.



Change CompositeProvider ZTSTCOMP

Description	ZTSTCOMP	Saved	◇		From	
InfoProvider	@3ZTSTCOMP	Activated	●		From	
Status of Object	Inactive, Not Executable	InfoArea				

[OPXL2TMP] OPXL2TMP (Union)

Field
Line Item (General Ledger View)
Document Number (General Ledge
Business area
Chart of accounts
Company code
Order Number
Order type
Cost Element
Controlling area
Name of person who created the o
Date on which the record was crea
Currency Key
Currency Type
Debit/credit indicator

ZTSTCOMP

Field	Unit
Line Item (General Ledger View)	
Business area	
Chart of accounts	
Fiscal year	
Amount	

[YUDSOPXP_001] YUDSOPXP (Inner J

Field
Calendar Year/Month
Calendar month
Calendar Year
Currency Key
Fiscal year variant
Opex Cost Category
Opex: Field
Opex:FLD: Field Level Sup
Amount

Figure 2 Selecting Fields from InfoProvider

9. To change the type of join, right-click on the InfoProvider, click on CONNECT AS, and change per the requirement.

If there are any navigational attributes in the InfoProvider, the system will show them as characteristic fields.

Tip 12

Keeping Runtimes Short for Large Data Loads using Semantically Partitioned Objects

You can create year-based semantically partitioned objects to partition huge volumes of data for better system performance.

A common pain point for users of SAP NetWeaver BW is handling large data volumes. The larger the data volume, the longer the runtimes required for standard DSOs and standard InfoCubes.

SAP NetWeaver BW release 7.3 comes with a new type of InfoProvider called a *semantically partitioned object (SPO)*. This allows you to partition the InfoCube or DSO so that the data sets are distributed over several data containers, which keeps query runtimes short even if the data volume is large.

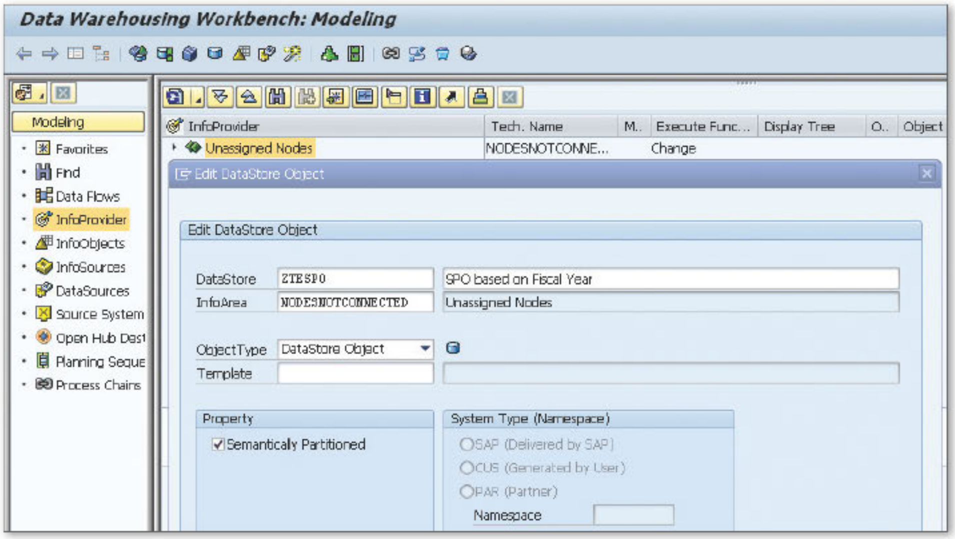
If you use a calendar year as the partition criteria, the common requirement is to make the SPO flexible enough that changes in year are automatically reflected in the partition. This tip describes how to create an SPO based on the fiscal year and also how to use a BAdI to change the year and reflect the change on the SPO.

✓ And Here's How ...

This tip will use a scenario of creating a DSO that is expected to have large volumes of data. The challenge with using the fiscal year is the handling of the change in the year; that is, how we manage the partitions when the year changes (this is where the BAdI comes in).

Note that the SPOs are created on an InfoCube or a DSO, so the basic steps in creating an InfoProvider are the same steps that you would follow during the creation of an InfoCube or DSO. Follow these steps:

1. Execute Transaction RSA1 in the SAP NetWeaver BW system.
2. Create a new InfoProvider of OBJECTTYPE DATASTORE OBJECT. Select the SEMANTICALLY PARTITIONED checkbox as shown in Figure 1.



Edit DataStore Object

DataStore: ZTESPO — SPO based on Fiscal Year
InfoArea: NODESNOTCONNECTED — Unassigned Nodes
ObjectType: DataStore Object
Template:

Property
[✓] Semantically Partitioned

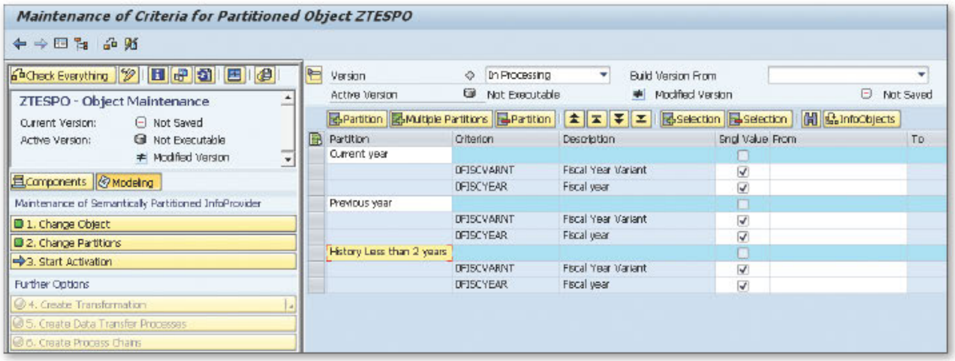
System Type (Namespace)
○ SAP (Delivered by SAP)
○ CUS (Generated by User)
○ PAR (Partner)
Namespace

Figure 1 Creating an SPO Based on a DSO

3. On the resulting screen, go to the OBJECT MAINTENANCE tab and input the structure of the DSO by adding the key fields, data fields, and navigational attributes that are required to model the DSO.
4. On the left panel, click on MAINTAIN PARTITIONS, and in the resulting window select the field on which you wanted to perform the logical partition (FISCAL YEAR in this example), and click CONTINUE.
5. The next screen, MAINTENANCE OF CRITERIA FOR PARTITIONED OBJECT, will provide more options for defining the criteria for the partitions. Using this screen, you may add as many partitions as you need for partitioning field FISCAL YEAR that you selected in the previous step. Click on the PARTITION button ( ) to add more partitions.

In addition to creating partitions as described in the previous step, you can also specify the partitioning criteria by implementing a BAdI: `RSLPO_BADI_PARTITIONING`.

You can create a BAdI implementation by using the dropdown box associated with Build Version From option in Figure 2. The BAdI contains interface `IF_RSLPO_BADI_PARTITIONING`, which provides three methods that can be used for creating new partitions programmatically.



ZTESPO - Object Maintenance

Current Version: Not Saved
Active Version: Not Executable
Modified Version

Maintenance of Semantically Partitioned InfoProvider

1. Change Object
2. Change Partitions
3. Start Activation

Further Options

4. Create Transformation
5. Create Data Transfer Processes
6. Create Process Chains

Version: In Processing — Build Version From:
Active Version: Not Executable — Modified Version — Not Saved

Partition	Criterion	Description	Sngl Value	From	To
Current year			☐		
	0FISCVARNT	Fiscal Year Variant	✓		
	0FISCYEAR	Fiscal year	✓		
Previous year			☐		
	0FISCVARNT	Fiscal Year Variant	✓		
	0FISCYEAR	Fiscal year	✓		
History Less than 2 years			☐		
	0FISCVARNT	Fiscal Year Variant	✓		
	0FISCYEAR	Fiscal year	✓		

Figure 2 Creating Partitions for Fiscal Year

- **Method 1: `GET_T_SPO`**
  Supplies name of the SPO. Example of code is as follows:

```
IF_RSLPO_BADI_PARTITIONING~GET_T_SPO
      DATA: L_SPO TYPE RSLPONAME.

    L_SPO = 'SPONAME'
    APPEND L_SPO  TO R_T_SPO.
  ENDMETHOD
```

- **Method 2: `GET_T_PART`**
  Accesses the partitions of the SPO.

```
IF_RSLPO_BADI_PARTITIONING~GET_T_PART
    DATA: L_S_PART TYPE RSLPO_BADI_S_PART.

  CASE I_SPO.
    WHEN 'SPONAME'
"Partition 1
      L_S_PART-IDPART = '01'.
      L_S_PART-POSIT = 1.
      APPEND L_S_PART TO R_T_PART.
"Partition 2
      L_S_PART-IDPART = '02'.
      L_S_PART-POSIT = 2.
      APPEND L_S_PART TO R_T_PART.

      "Partition 3
      L_S_PART-IDPART = '03'.
      L_S_PART-POSIT = 3.
      APPEND L_S_PART TO R_T_PART.
ENDCASE.
```

- **Method 3:** `GET_T_PART_CRIT`
  Accesses the partitioning criteria. You may code your logic to create new partitions based on the current year.

```
DATA: L_S_PART_CRIT TYPE RSLPO_BADI_S_PART_CRIT,
        L_DATE TYPE SY-DATUM,
        L_CYEAR TYPE T009B-BDATJ,
        L_PYEAR TYPE T009B-BDATJ,
        L_PPYEAR TYPE T009B-BDATJ,
        L_NYEAR TYPE T009B-BDATJ,
        L_FYEAR TYPE T009B-BDATJ VALUE 2005.

* Identify current Fiscal Year
  L_DATE = SY-DATUM.

  CALL FUNCTION 'DATE_TO_PERIOD_CONVERT'
    EXPORTING
      I_DATE         = L_DATE
*     I_MONMIT       = 00
      I_PERIV        = 'K4'
    IMPORTING
*     E_BUPER        =
      E_GJAHR        = L_CYEAR
    EXCEPTIONS
      INPUT_FALSE    = 1
      T009_NOTFOUND  = 2
      T009B_NOTFOUND = 3
      OTHERS         = 4.
  IF SY-SUBRC NE 0.
    EXIT.
  ENDIF.
  "Previous Fiscal Year
  L_PYEAR = L_CYEAR - 1.
  "Current year - 2
  L_PPYEAR = L_CYEAR - 2.
  "Current year + 1
  L_NYEAR = L_CYEAR + 1.
CASE I_SPO.
    WHEN 'SPONAME'.
      "Partition 1
      L_S_PART_CRIT-IDPART = 'A1'.
      L_S_PART_CRIT-IOBJNM = '0FISCYEAR'.
      L_S_PART_CRIT-POSIT  = 1.
      L_S_PART_CRIT-OPT    = 'EQ'.
      L_S_PART_CRIT-LOW    = L_CYEAR.
      APPEND L_S_PART_CRIT TO R_T_PART_CRIT.

      "Partition 2
      L_S_PART_CRIT-IDPART = 'A2'.
      L_S_PART_CRIT-IOBJNM = '0FISCYEAR'.
      L_S_PART_CRIT-POSIT  = 2.
      L_S_PART_CRIT-OPT    = 'EQ'.
      L_S_PART_CRIT-LOW    = L_PYEAR.
      APPEND L_S_PART_CRIT TO R_T_PART_CRIT.

      "Partition 3
      L_S_PART_CRIT-IDPART = 'A3'.
      L_S_PART_CRIT-IOBJNM = '0FISCYEAR'.
      L_S_PART_CRIT-POSIT  = 3.
      L_S_PART_CRIT-OPT    = 'BT'.
      L_S_PART_CRIT-LOW    = L_FYEAR.
      L_S_PART_CRIT-HIGH   = L_PPYEAR.
      APPEND L_S_PART_CRIT TO R_T_PART_CRIT.

  ENDCASE.

ENDMETHOD.
```

Tip 13

Converting Key Figure Models to Account Models via a Rule Group

You can use an SAP-provided functionality in transformation to convert a key figure model in a source system into an account model to suit business requirements.

It's common to find that source data models exist in a source system key figure model in which each key performance indicator (KPI) is represented as a separate field. However, the InfoCube design requirements may call for you to organize these KPIs in an account model on the SAP NetWeaver BW side. Typically, you would implement this by writing ABAP code in the end routines or the expert routines of the transformations; however, coding this kind of logic can be complex and hard to maintain.

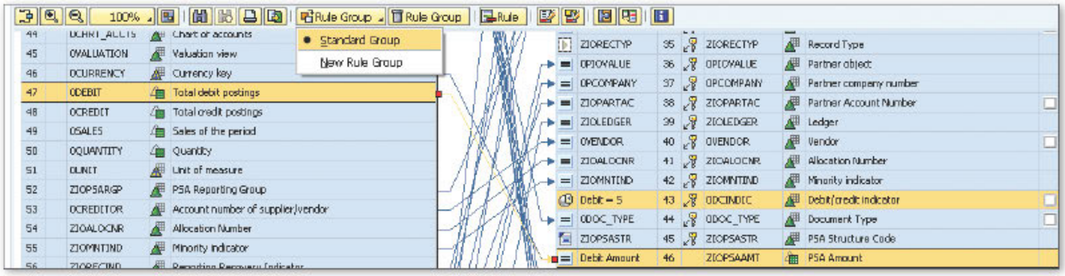
You can accomplish this task without writing a single line of code using rule groups in transformations. A rule group is a group of transformation rules that contain one transformation rule for each key field of the target. A transformation can contain multiple rule groups. This tip explains how to convert a key figure model to an account model using the rule group functionality in the transformations.

✓ And Here's How ...

In our example, the source model has a debit posting amount and the credit posting amount in one record. However, the target DSO has only one key figure to hold the amount. So in this case, one single record will need to split into two records and also need to provide an indicator for debit or credit to identify the records.

You can create a separate rule group in the transformation to achieve this by executing Transaction RSA1 in the SAP NetWeaver BW system and following these steps:

1. Create a transformation between source and target; by default, rule group is the standard group. Click on Rule Group Standard Group.
2. Map the Amount field to Debit from the source and give a constant value for Debit/Credit Indicator (here, Debit = S) to differentiate between the debit amount and credit amount (see Figure 1).

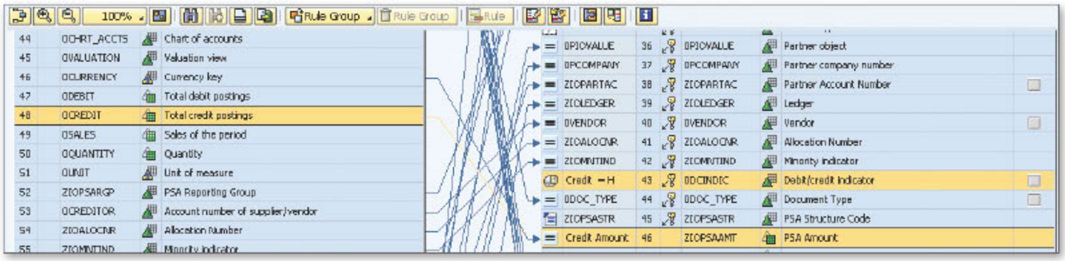


44	0CHRT_ACCTS	Chart of accounts
45	0VALUATION	Valuation view
46	0CURRENCY	Currency key
47	0DEBIT	Total debit postings
48	0CREDIT	Total credit postings
49	0SALES	Sales of the period
50	0QUANTITY	Quantity
51	0UNIT	Unit of measure
52	ZIOPSARGP	PSA Reporting Group
53	0CREDITOR	Account number of supplier/vendor
54	ZIOALOCNR	Allocation Number
55	ZIOMNTIND	Minority indicator
56	ZIORECIND	Reporting Recovery Indicator

ZIORECTYP	35	ZIORECTYP	Record Type
0PIOVALUE	36	0PIOVALUE	Partner object
0PCOMPANY	37	0PCOMPANY	Partner company number
ZIOPARTAC	38	ZIOPARTAC	Partner Account Number
ZIOLEDGER	39	ZIOLEDGER	Ledger
0VENDOR	40	0VENDOR	Vendor
ZIOALOCNR	41	ZIOALOCNR	Allocation Number
ZIOMNTIND	42	ZIOMNTIND	Minority indicator
Debit = S	43	0DCINDIC	Debit/credit indicator
0DOC_TYPE	44	0DOC_TYPE	Document Type
ZIOPSASTR	45	ZIOPSASTR	PSA Structure Code
Debit Amount	46	ZIOPSAAMT	PSA Amount

Figure 1 Map Amount to Debit

3. Map the 0DEBIT Total Debit Postings to the amount InfoObject ZIOPSAAMT.
4. Create a new rule group by clicking on the Rule Group dropdown and then by choosing New Rule Group.
5. In the resulting popup, provide a description for the new rule group and click the Continue button. Here, we've named our new rule group "Credit Posting."
6. Map the amount field to credit from the source, and give a constant value for Debit/Credit Indicator (here, Debit = H) to differentiate between the debit amount and credit amount as shown in Figure 2.



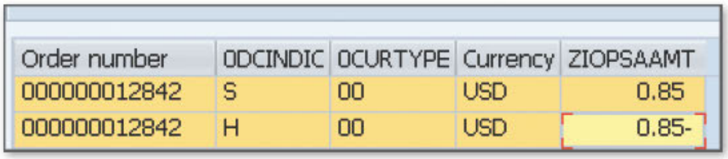
44	0CHRT_ACCTS	Chart of accounts
45	0VALUATION	Valuation view
46	0CURRENCY	Currency key
47	0DEBIT	Total debit postings
48	0CREDIT	Total credit postings
49	0SALES	Sales of the period
50	0QUANTITY	Quantity
51	0UNIT	Unit of measure
52	ZIOPSARGP	PSA Reporting Group
53	0CREDITOR	Account number of supplier/vendor
54	ZIOALOCNR	Allocation Number
55	ZIOMNTIND	Minority indicator

0PIOVALUE	36	0PIOVALUE	Partner object
0PCOMPANY	37	0PCOMPANY	Partner company number
ZIOPARTAC	38	ZIOPARTAC	Partner Account Number
ZIOLEDGER	39	ZIOLEDGER	Ledger
0VENDOR	40	0VENDOR	Vendor
ZIOALOCNR	41	ZIOALOCNR	Allocation Number
ZIOMNTIND	42	ZIOMNTIND	Minority indicator
Credit = H	43	0DCINDIC	Debit/credit Indicator
0DOC_TYPE	44	0DOC_TYPE	Document Type
ZIOPSASTR	45	ZIOPSASTR	PSA Structure Code
Credit Amount	46	ZIOPSAAMT	PSA Amount

Figure 2 Mapping Rules for Credit: Map Amount to Credit

7. Map the 0CREDIT Total credit postings from the source to the amount InfoObject ZIOPSAAMT in the target as shown in Figure 2.

In these mappings, we have mapped both the credit postings and the debit postings from the source to the same InfoObject in the target. When data is loaded to the InfoProvider, as shown in Figure 3 there will be one key figure that holds both credit and debit data.



Order number	0DCINDIC	0CURTYPE	Currency	ZIOPSAAMT
000000012842	S	00	USD	0.85
000000012842	H	00	USD	0.85-

Figure 3 Contents of the InfoCube

Using the rule groups, you can split one record with separate key figures for debit and credit postings into two separate records by using simple mapping rules without any ABAP coding. Thus, you can use this rule group concept to simplify the transformation logic.

Tip 14

Modeling Planning Scenarios on Direct Update DataStore Objects

To avoid the system and performance issues associated with modeling using real-time InfoCubes, you can use DSOs as the data basis for Integrated Planning.

Integrated Planning is widely used for modeling planning scenarios in SAP NetWeaver BW. Currently, Integrated Planning data comes from InfoProviders that contain real-time InfoCubes. However, there are some system and performance drawbacks to using real-time InfoCubes, such as reduced read-only performance. Another major drawback is the complexity of implementing look-ups on the real-time InfoCubes because the InfoCube model is based on a combination of several tables (dimensions, facts, and attributes), and you cannot look up an InfoCube as you would with a DSO or a relational table.

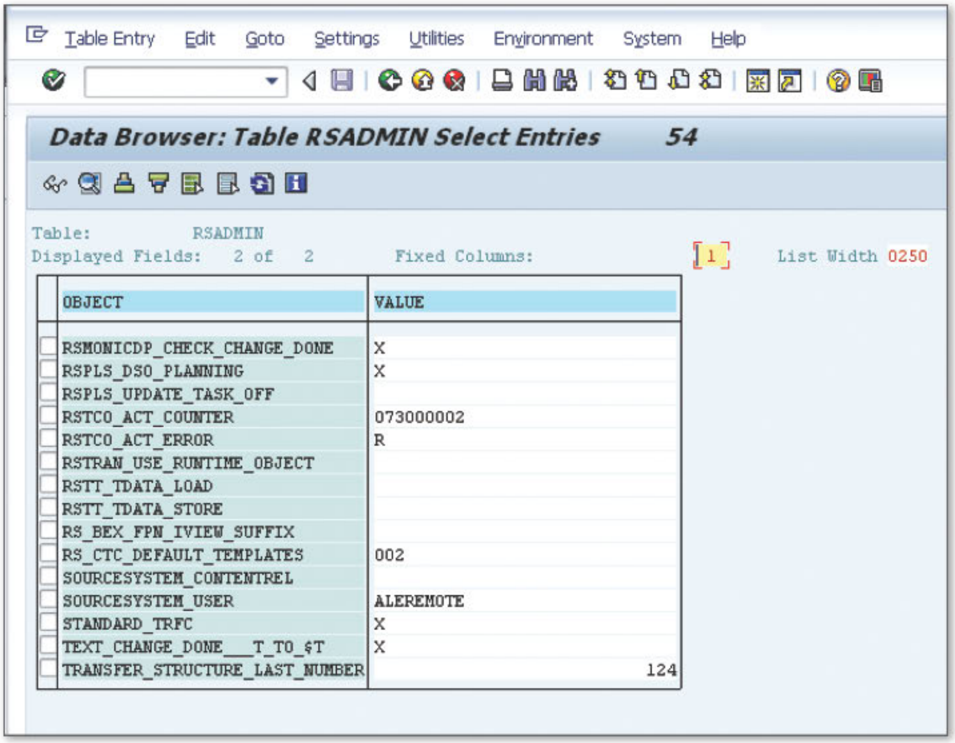
With SAP NetWeaver BW 7.3, SP08, planning scenarios can also be modeled on top of DSOs, which eliminates some of these drawbacks. This tip describes how to enable a DSO for planning.

✓ And Here's How ...

The basic steps for creating the direct update DSO are the same as creating a regular DSO such as defining Key Fields, Data Fields, Navigational Attributes, and so on. This tip describes the additional steps to enable the DSO for planning.

To enable a DSO for planning, execute Transaction SE11 in the SAP NetWeaver BW system, and then follow these steps:

1. Go to Table RSADMIN, and set the parameter RSPLS_DSO_PLANNING as X (see Figure 1). This is a one-time system activity.



Data Browser: Table RSADMIN Select Entries 54

```
Table:           RSADMIN
Displayed Fields:   2 of   2       Fixed Columns:          1      List Width 0250
```

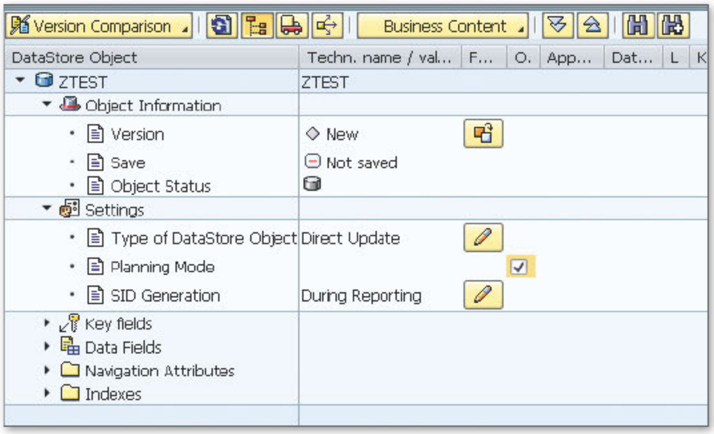
OBJECT	VALUE
RSMONICDP_CHECK_CHANGE_DONE	X
RSPLS_DSO_PLANNING	X
RSPLS_UPDATE_TASK_OFF	
RSTCO_ACT_COUNTER	073000002
RSTCO_ACT_ERROR	R
RSTRAN_USE_RUNTIME_OBJECT	
RSTT_TDATA_LOAD	
RSTT_TDATA_STORE	
RS_BEX_FPN_IVIEW_SUFFIX	
RS_CTC_DEFAULT_TEMPLATES	002
SOURCESYSTEM_CONTENTREL	
SOURCESYSTEM_USER	ALEREMOTE
STANDARD_TRFC	X
TEXT_CHANGE_DONE__T_TO_$T	X
TRANSFER_STRUCTURE_LAST_NUMBER	124

Figure 1 RSADMIN Setting to Enable DSO Planning

This setting will enable the checkbox for turning ON the PLANNING MODE in the settings of a DSO.

2. Go to the Administrative Workbench, and create a direct update DSO by clicking the EDIT button next to TYPE OF DATASTORE OBJECT.
3. You'll now see the DATASTORE OBJECT: SELECT TYPE pop-up screen; select the DIRECT UPDATE checkbox and click CONTINUE.
4. Another pop up will appear that says PROPERTY "IN-MEMORY OPTIMIZED" WILL BE RESET. Click the CONTINUE button.

5. The Direct Update setting that we made in the previous step will change the options available in the Settings section of the DSO. A Planning Mode check-box will appear in the settings of the DSO as shown in Figure 2. Check the box to use this DSO for planning.



DataStore Object	Techn. name / val...	F...	O.	App...	Dat...	L	K
ZTEST	ZTEST						
Object Information							
Version	New						
Save	Not saved						
Object Status							
Settings							
Type of DataStore Object	Direct Update						
Planning Mode			✓				
SID Generation	During Reporting						
Key fields							
Data Fields							
Navigation Attributes							
Indexes							

Figure 2 Planning Mode Option

All characteristics that are required for planning (including unit fields) should be part of the DSO key. In other words, only key figures can be part of the data fields of a plan-enabled DSO.

As with a regular DSO, you can add navigational attributes and indexes as needed and activate the DSO on completion of the configuration.

Tip 15

Analyzing the Effectiveness of InfoCube Design per SAP Recommendations

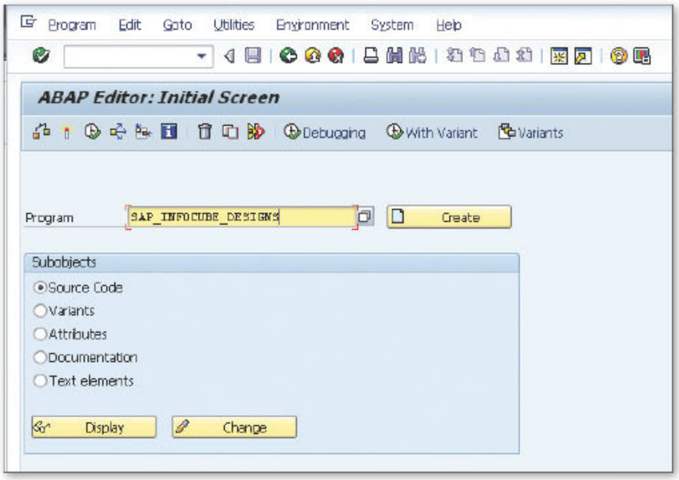
You can analyze the efficiency of your InfoCube design and troubleshoot any performance issues that may exist due to the modeling of the InfoCube.

SAP NetWeaver BW InfoCubes are designed using a multidimensional modeling approach that offers 1 fact table and 16 dimensions in an InfoCube. Within each dimension, there can be up to 256 attributes (InfoObjects). In a multidimensional model, the fact table is joined by dimension tables, and the number of records in a dimension table directly affects the performance of queries against the InfoCube. Therefore, SAP recommends that each dimension table should not contain more than 20% of rows as compared to the fact table.

This tip describes how an SAP standard program can be used to analyze the distribution of the InfoCube data among the dimension tables and identify potential problematic dimension table(s).

✓ And Here's How ...

To use the SAP standard program for analysis, log on to the SAP NetWeaver BW system and execute Transaction SE38. In the Program field, enter program name "SAP_INFOCUBE_DESIGNS", and click on the Execute button (see Figure 1).

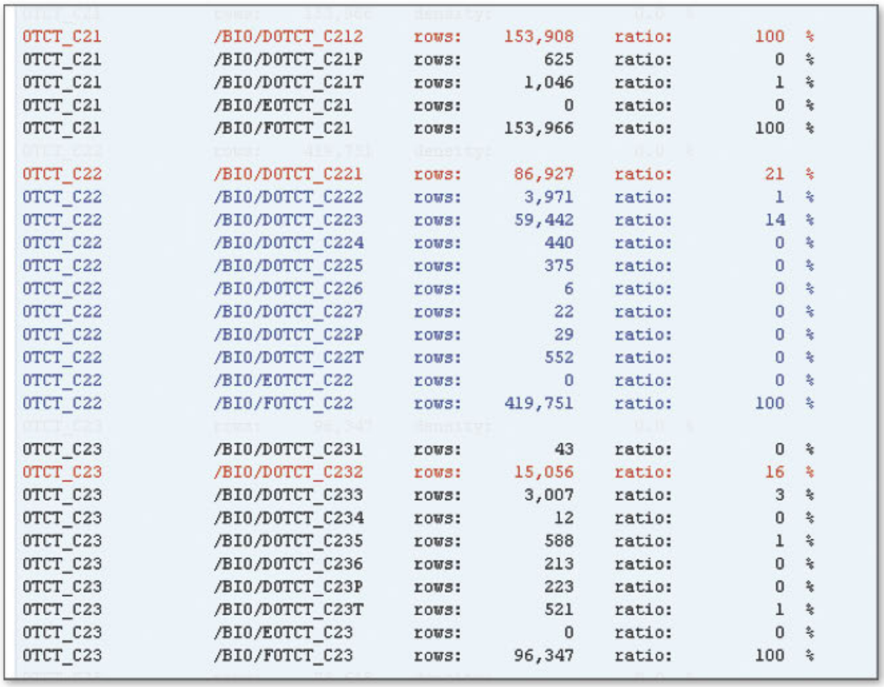


Program: SAP_INFOCUBE_DESIGNS

Subobjects
[✓ Source Code]
Variants
Attributes
Documentation
Text elements

Figure 1 ABAP Editor Initial Screen with Program SAP_INFOCUBE_DESIGNS

The output of SAP_INFOCUBE_DESIGNS displays the number of rows in fact tables and corresponding dimension tables (see Figure 2). Fact table names start with /BIO/F, whereas dimension table names start with /BIO/D.



```
OTCT_C21         /BIO/DOTCT_C212     rows:   153,908    ratio:        100  %
OTCT_C21         /BIO/DOTCT_C21P     rows:       625    ratio:          0  %
OTCT_C21         /BIO/DOTCT_C21T     rows:     1,046    ratio:          1  %
OTCT_C21         /BIO/EOTCT_C21      rows:         0    ratio:          0  %
OTCT_C21         /BIO/FOTCT_C21      rows:   153,966    ratio:        100  %

OTCT_C22         /BIO/DOTCT_C221     rows:    86,927    ratio:         21  %
OTCT_C22         /BIO/DOTCT_C222     rows:     3,971    ratio:          1  %
OTCT_C22         /BIO/DOTCT_C223     rows:    59,442    ratio:         14  %
OTCT_C22         /BIO/DOTCT_C224     rows:       440    ratio:          0  %
OTCT_C22         /BIO/DOTCT_C225     rows:       375    ratio:          0  %
OTCT_C22         /BIO/DOTCT_C226     rows:         6    ratio:          0  %
OTCT_C22         /BIO/DOTCT_C227     rows:        22    ratio:          0  %
OTCT_C22         /BIO/DOTCT_C22P     rows:        29    ratio:          0  %
OTCT_C22         /BIO/DOTCT_C22T     rows:       552    ratio:          0  %
OTCT_C22         /BIO/EOTCT_C22      rows:         0    ratio:          0  %
OTCT_C22         /BIO/FOTCT_C22      rows:   419,751    ratio:        100  %

OTCT_C23         /BIO/DOTCT_C231     rows:        43    ratio:          0  %
OTCT_C23         /BIO/DOTCT_C232     rows:    15,056    ratio:         16  %
OTCT_C23         /BIO/DOTCT_C233     rows:     3,007    ratio:          3  %
OTCT_C23         /BIO/DOTCT_C234     rows:        12    ratio:          0  %
OTCT_C23         /BIO/DOTCT_C235     rows:       588    ratio:          1  %
OTCT_C23         /BIO/DOTCT_C236     rows:       213    ratio:          0  %
OTCT_C23         /BIO/DOTCT_C23P     rows:       223    ratio:          0  %
OTCT_C23         /BIO/DOTCT_C23T     rows:       521    ratio:          1  %
OTCT_C23         /BIO/EOTCT_C23      rows:         0    ratio:          0  %
OTCT_C23         /BIO/FOTCT_C23      rows:    96,347    ratio:        100  %
```

Figure 2 Analysis of the InfoCube

This output can be used to make decisions on data modeling or redesign of the InfoCubes.

When the number of rows in a dimension table exceeds 20% of rows in the fact table, it could represent a possible source of query performance. For instance, in Figure 2, the first record shows that the dimension table /BIO/D0TCT_C212 has a 100% ratio, which means that this dimension table has the same number of rows as the fact table /BIO/F0TCT_C21. This could point to design flaws and if indeed the number of rows in the dimension table has such a high ratio, then you could consider designating this dimension as a line item dimension.

This method can be used during the development of the InfoCube to ensure that correct modeling decisions are made.

If you are troubleshooting a performance issue in a performance environment, the findings of this analysis can help you decide whether the performance can be improved by remodeling the InfoCube.

Tip 16

Implementing Date and Time Fields as Key Figures

For some modeling scenarios, you can use date and time fields as key figures to allow further calculation and delta updates.

Typically, in SAP NetWeaver BW systems, the date and time fields are modeled using standard InfoObjects 0DATE and 0TIME. These InfoObjects are treated as characteristics and can be used for activities such as reporting and filtering. However, some reporting scenarios require calculations based on date and time fields, which requires a more flexible SAP NetWeaver BW design for date/time calculations. The 0DATE and 0TIME characteristics do not provide that kind of functionality.

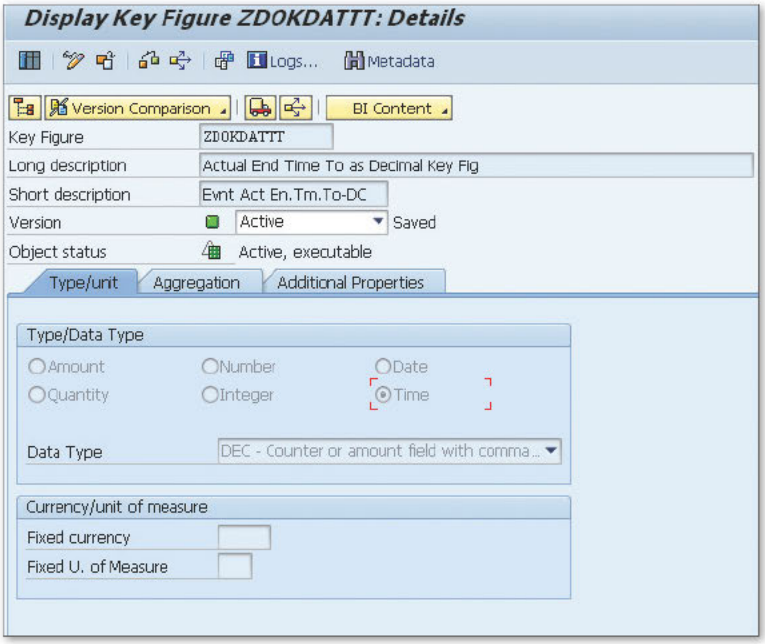
The alternative is to model the data and times fields as key figures. This tip shows specific design guidelines to be followed when implementing this in SAP NetWeaver BW.

✓ And Here's How ...

Let's assume that we have four date and time fields available in the source DSO that represent actual event start and actual end date and times. We'll show you how to store them as key figures, which can be used to created restricted key figures using date and time fields stored as key figures.

To implement the date and time fields as key figures, log in to the SAP NetWeaver BW system and follow these steps:

1. Using Transaction RSA1, create two new key figures, "Event Actual Start Date From as Decimal Key Fig" and "Event Actual End Date To as Decimal Key Fig" with TYPE/DATA TYPE as DATE and DATA TYPE as DEC - COUNTER OR AMOUNT FIELD WITH COMMA AND SIGN (see Figure 1).



Display Key Figure ZDOKDATTT: Details

Key Figure	ZDOKDATTT
Long description	Actual End Time To as Decimal Key Fig
Short description	Evnt Act En.Tm.To-DC
Version	🟩 Active Saved
Object status	Active, executable

Type/Data Type

Amount Number Date
Quantity Integer [✓ Time]

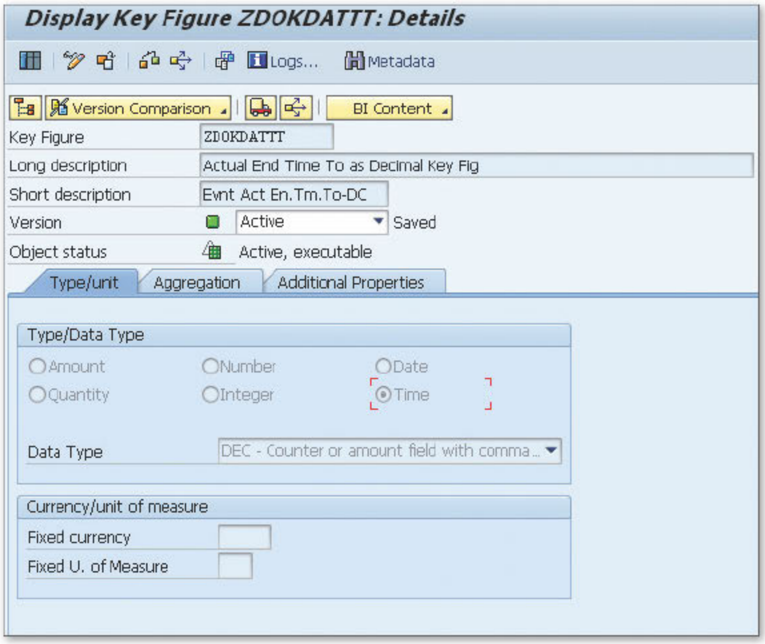
Data Type	DEC - Counter or amount field with comma...

Currency/unit of measure

Fixed currency	
Fixed U. of Measure	

Figure 1 New Date Field as Key Figure

2. Create two new key figures, "Actual Start Time From as Decimal Key Fig" and "Actual End Time To as Decimal Key Fig" with TYPE/DATA TYPE as TIME and DATA TYPE as DEC - COUNTER OR AMOUNT FIELD WITH COMMA AND SIGN (see Figure 2).
3. Include the newly created key figures in the source DSO that we are using for this scenario.


Display Key Figure ZDOKDATTT: Details

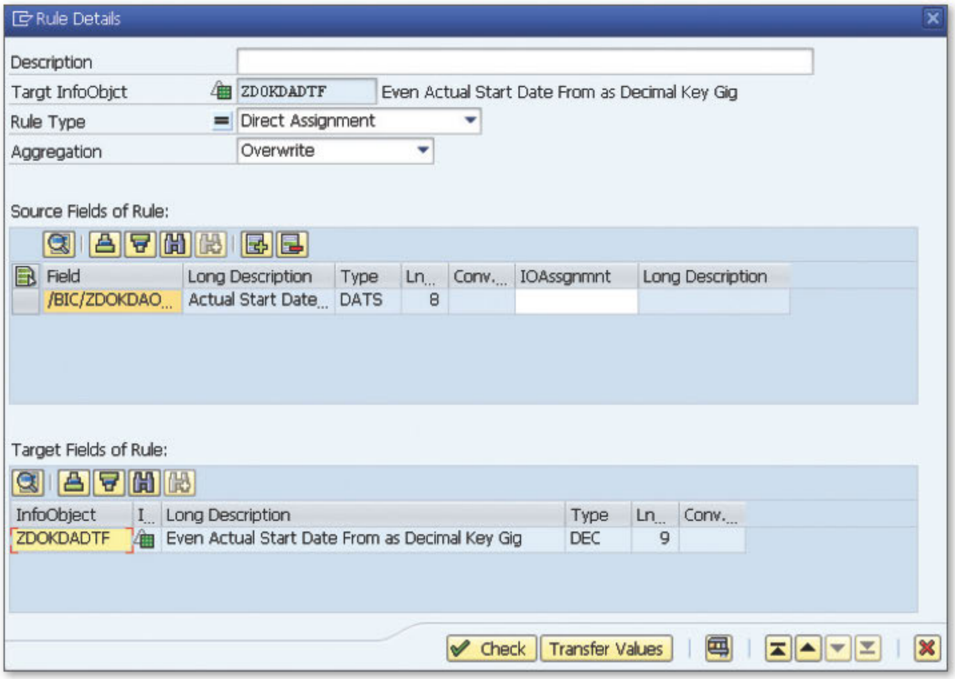
Key Figure: ZDOKDATTT
Long description: Actual End Time To as Decimal Key Fig
Short description: Evnt Act En.Tm.To-DC
Version: Active Saved
Object status: Active, executable

Type/Data Type: Amount, Number, Date, Quantity, Integer, [✓ Time]
Data Type: DEC - Counter or amount field with comma...

Currency/unit of measure
Fixed currency:
Fixed U. of Measure:

Figure 2 New Time Field as Key Figure

4. In the transformations for the source DSO, map the following:
   - Event Actual Start Date From as Decimal Key Fig to Actual Start Date From
   - Actual Start Time From as Decimal Key Fig to Actual Start Time From
   - Event Actual End Date To as Decimal Key Fig to Actual End Date To
   - Event Actual End Time To as Decimal Key Fig to Actual End Time To (see Figure 3)
5. The date and time fields will be stored in the DSO as a numeric value format (see Figure 4).



Description		
Targt InfoObjct	ZDOKDADTF	Even Actual Start Date From as Decimal Key Gig
Rule Type	Direct Assignment	
Aggregation	Overwrite	

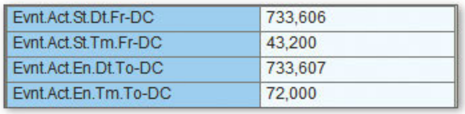
Source Fields of Rule:

Field	Long Description	Type	Ln...	Conv....	IOAssgnmnt	Long Description
/BIC/ZDOKDAO...	Actual Start Date...	DATS	8			

Target Fields of Rule:

InfoObject	I...	Long Description	Type	Ln...	Conv....
ZDOKDADTF		Even Actual Start Date From as Decimal Key Gig	DEC	9	

Figure 3 Mapping of Date/Time InfoObject to Key Figures



Evnt.Act.St.Dt.Fr-DC	733,606
Evnt.Act.St.Tm.Fr-DC	43,200
Evnt.Act.En.Dt.To-DC	733,607
Evnt.Act.En.Tm.To-DC	72,000

Figure 4 DSO Result Showing the Date and Time Values as Key Figures

Because the SAP system stores the date and time as numeric values, the delta updates to this key figure in the DSO will be handled correctly. These key figures can now be used for calculations.

Tip 17

Using an Existing BEx Query to Transform Your Data

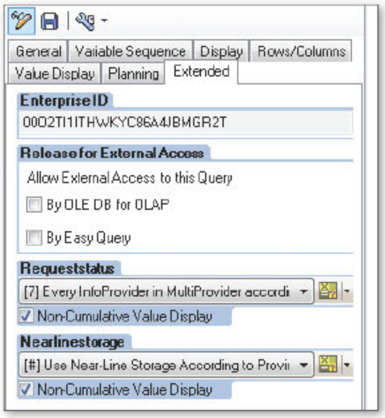
You can transform data for your reporting needs without development work in the system; instead, just use an existing BEx query in SAP NetWeaver BW 7.3!

BEx queries in an SAP NetWeaver BW environment typically have a lot of logic and business rules built into the definition. Some reporting scenarios may require you to leverage the business rules built into the BEx queries and further transform the data to meet your specific reporting requirements. For instance, if the BEx query has restricted and calculated key figures, then you can leverage these key figures and further transform the data. The common approach to modeling this kind of requirement is to create an Analysis Process Designer on the BEx query or programmatically derive the key figures in transformations. Both options require significant development work.

This tip describes an option available with SAP NetWeaver BW 7.3 that allows you to use a BEx query as a source for transformations. You can use this approach to leverage an existing BEx query, further transform the data, and save the results to an InfoProvider.

✓ And Here's How ...

A prerequisite to using any query as a source for transformations is to uncheck the BY OLE DB FOR OLAP option in the query in the BEx Query Designer. To do this, select the PROPERTIES option for the BEx query to see the properties (as shown in Figure 1).



EnterpriseID
0002TI1ITHWKYC86A4JBMGR2T

Release for External Access
Allow External Access to this Query
☐ By OLE DB for OLAP
☐ By Easy Query

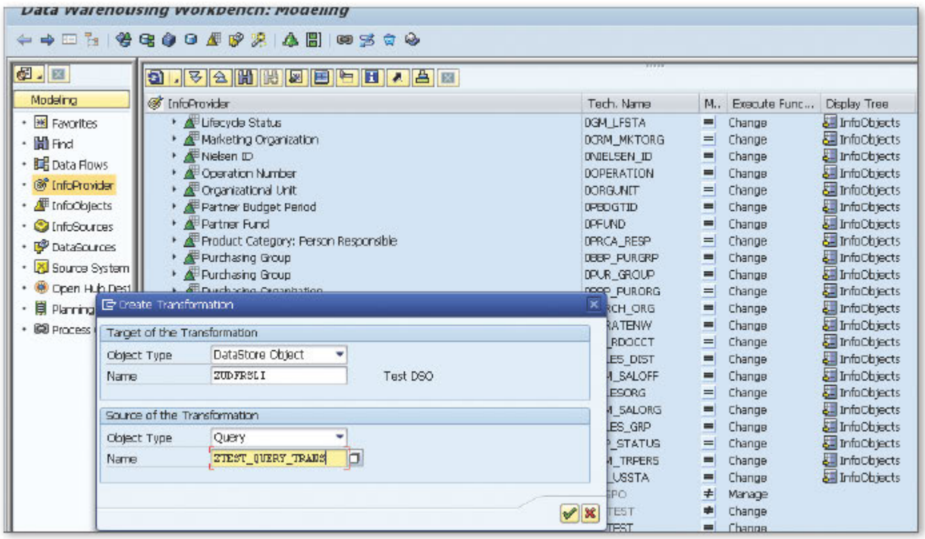
Requeststatus
[7] Every InfoProvider in MultiProvider accordi
☑ Non-Cumulative Value Display

Nearlinestorage
[#] Use Near-Line Storage According to Provi
☑ Non-Cumulative Value Display

« *Figure 1 Query Properties in BEx Query Designer*

Now, follow these steps:

1. Log in to the SAP NetWeaver BW 7.3 system, and execute Transaction RSA1.
2. Select the InfoProvider that you want to save the query results to. In this case, we'll use a test DSO that we created for this purpose. Right-click on the Test DSO, and click on Create Transformation.
3. On the resulting screen, select Query from the Object Type dropdown.
4. Input the query name that you want to use as a source (see Figure 2), and click on Continue.



InfoProvider	Tech. Name	M..	Execute Func...	Display Tree
Lifecycle Status	0GM_LFSTA	▬	Change	InfoObjects
Marketing Organization	0CRM_MKTORG	=	Change	InfoObjects
Nielsen ID	0NIELSEN_ID	▬	Change	InfoObjects
Operation Number	0OPERATION	▬	Change	InfoObjects
Organizational Unit	0ORGUNIT	=	Change	InfoObjects
Partner Budget Period	0PBDGTID	▬	Change	InfoObjects
Partner Fund	0PFUND	▬	Change	InfoObjects
Product Category: Person Responsible	0PRCA_RESP	=	Change	InfoObjects
Purchasing Group	0BBP_PURGRP	▬	Change	InfoObjects
Purchasing Group	0PUR_GROUP	▬	Change	InfoObjects
Purchasing Organization	[illegible]_PURORG	=	Change	InfoObjects
	RCH_ORG	▬	Change	InfoObjects
	RATENW	▬	Change	InfoObjects
	_RDOCCT	=	Change	InfoObjects
	LES_DIST	▬	Change	InfoObjects
	M_SALOFF	▬	Change	InfoObjects
	LESORG	=	Change	InfoObjects
	M_SALORG	▬	Change	InfoObjects
	LES_GRP	▬	Change	InfoObjects
	P_STATUS	=	Change	InfoObjects
	M_TRPERS	▬	Change	InfoObjects
	_USSTA	▬	Change	InfoObjects
	EPO	≠	Manage	
	TEST	▬	Change	
	TEST	▬	Change	

Create Transformation

Target of the Transformation

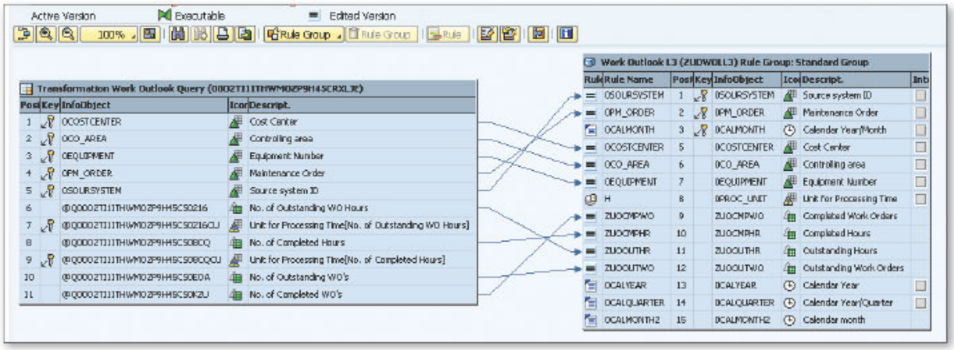
Object Type	DataStore Object	
Name	ZUDFRSLI	Test DSO

Source of the Transformation

Object Type	Query
Name	ZTEST_QUERY_TRANS

⇑ *Figure 2 Selecting the Query to Be Used for the Transformation*

5. Map the appropriate fields in the transformation, and activate the transformation (see Figure 3). You can map fields in the BEx query to the fields on the Test DSO. You may also define further transformation rules using start routines, end routines, expert routines, or simple mapping rules.



Transformation Work Outlook Query (0002T111THWM0ZP9H45CRXL3E)

Pos	Key	InfoObject	Icon	Descript.
1	🔑	0COSTCENTER		Cost Center
2	🔑	0CO_AREA		Controlling area
3	🔑	0EQUIPMENT		Equipment Number
4	🔑	0PM_ORDER		Maintenance Order
5	🔑	0SOURSYSTEM		Source system ID
6		@Q0002T111THWM0ZP9H45CS0216		No. of Outstanding WO Hours
7	🔑	@Q0002T111THWM0ZP9H45CS0216CU		Unit for Processing Time[No. of Outstanding WO Hours]
8		@Q0002T111THWM0ZP9H45CS08CQ		No. of Completed Hours
9	🔑	@Q0002T111THWM0ZP9H45CS08CQCU		Unit for Processing Time[No. of Completed Hours]
10		@Q0002T111THWM0ZP9H45CS0E0A		No. of Outstanding WO's
11		@Q0002T111THWM0ZP9H45CS0K2U		No. of Completed WO's

Work Outlook L3 (ZUDWOLL3) Rule Group: Standard Group

Rule	Rule Name	Pos	Key	InfoObject	Icon	Descript.	Int
	0SOURSYSTEM	1	🔑	0SOURSYSTEM		Source system ID	☐
	0PM_ORDER	2	🔑	0PM_ORDER		Maintenance Order	☐
	0CALMONTH	3	🔑	0CALMONTH		Calendar Year/Month	☐
	0COSTCENTER	5		0COSTCENTER		Cost Center	☐
	0CO_AREA	6		0CO_AREA		Controlling area	☐
	0EQUIPMENT	7		0EQUIPMENT		Equipment Number	☐
	H	8		0PROC_UNIT		Unit for Processing Time	☐
	ZUOCMPWO	9		ZUOCMPWO		Completed Work Orders	
	ZUOCMPHR	10		ZUOCMPHR		Completed Hours	
	ZUOOUTHR	11		ZUOOUTHR		Outstanding Hours	
	ZUOOUTWO	12		ZUOOUTWO		Outstanding Work Orders	
	0CALYEAR	13		0CALYEAR		Calendar Year	☐
	0CALQUARTER	14		0CALQUARTER		Calendar Year/Quarter	☐
	0CALMONTH2	15		0CALMONTH2		Calendar month	

Figure 3 Defining Transformation Rules

6. After the transformation is created, create the DTP and execute it to load data using the BEx query to the Test DSO.

Using this approach, you can use any BEx query as a source, transform the data further using the standard transformations functionality available in SAP NetWeaver BW, and then save the results to an InfoProvider.

Tip 18

Using Integrated Planning to Enable Planning with Master Data That Doesn't Yet Exist

You can plan for future data that doesn't exist yet in SAP NetWeaver BW by implementing custom read class method logic in Integrated Planning.

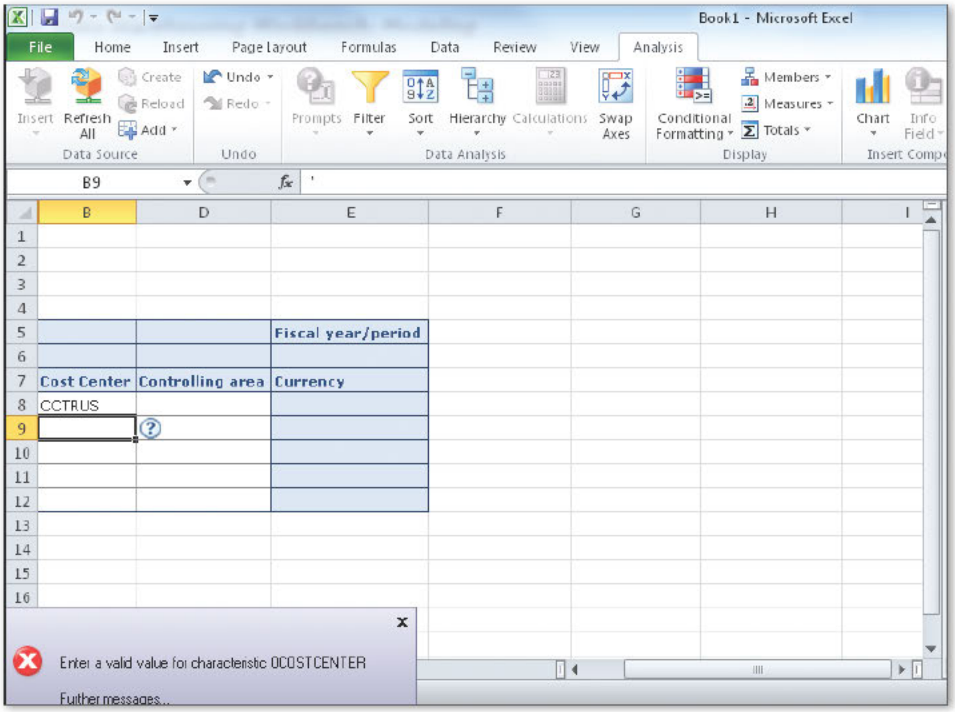
Some business scenarios require you to enter master data directly in SAP NetWeaver BW (rather than in a specific SAP component) either using master data maintenance or using an input screen created by the Integrated Planning function. This scenario is common in planning applications where you need to plan for future years, but the master data for the planning object does not exist in the SAP NetWeaver BW system. If you enter a value for an InfoObject that doesn't exist in the SAP NetWeaver BW system, then the system will generate an error stating that the value is not valid. This error is due to the validation check for SID that is implemented automatically for the InfoObject.

This tip describes a method to overcome this validation check by creating a read class method that will have logic to bypass this check.

✓ And Here's How ...

Before you get started, note that the implementation of this tip requires knowledge of ABAP OO concepts and knowledge of classes, methods, and interfaces.

We'll use a simple scenario to illustrate this tip. The user is expected to enter the cost center, controlling area, fiscal period, and amount for a planning scenario. Figure 1 shows an Integrated Planning screen for this data entry. When you enter the value "CCTRUS" in the COST CENTER field, the system generates an error as shown at the bottom of the screen. The error occurs due to the validation check because the cost center doesn't yet exist in the system.



	B	D	E
5			Fiscal year/period
6			
7	Cost Center	Controlling area	Currency
8	CCTRUS		
9			

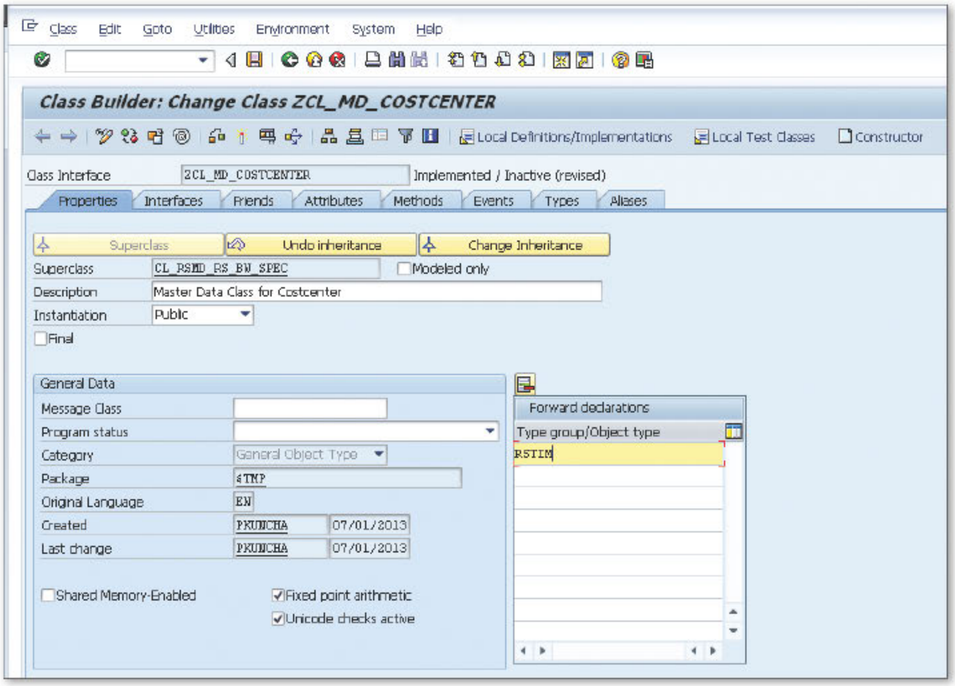
Enter a valid value for characteristic 0COSTCENTER

Further messages...

Figure 1 *Data Entry Screen for Plan Data for Cost Center with Error on Validation*

Follow these steps to enable entry of master data value for 0COSTCENTER, even if the value does not exist in the master data for the InfoObject:

1. Log in to SAP NetWeaver BW, and go to Transaction SE24. Create a new class by entering "ZCL_MD_COSTCENTER" for the OBJECT TYPE, and then click on the CREATE button.
2. You'll now see the CLASS BUILDER screen where you can define the class. Go to the INTERFACE tab, and enter "IF_RSMD_RS_ACCESS" in the INTERFACE column.
3. Go to the PROPERTIES tab, and click on the SUPERCLASS button.
4. Enter the SUPERCLASS name as "CL_RSMD_RS_BW_SPEC" and enter "RSTIM" in the FORWARD DECLARATIONS field (see Figure 2). Save your changes.



Class Builder: Change Class ZCL_MD_COSTCENTER

Class Interface: ZCL_MD_COSTCENTER Implemented / Inactive (revised)

Superclass: CL_RSMD_RS_BW_SPEC ☐ Modeled only
Description: Master Data Class for Costcenter
Instantiation: Public
☐ Final

General Data
Message Class:
Program status:
Category: General Object Type
Package: $TMP
Original Language: EN
Created: PKUNCHA 07/01/2013
Last change: PKUNCHA 07/01/2013
☐ Shared Memory-Enabled
☑ Fixed point arithmetic
☑ Unicode checks active

Forward declarations
Type group/Object type: RSTIM

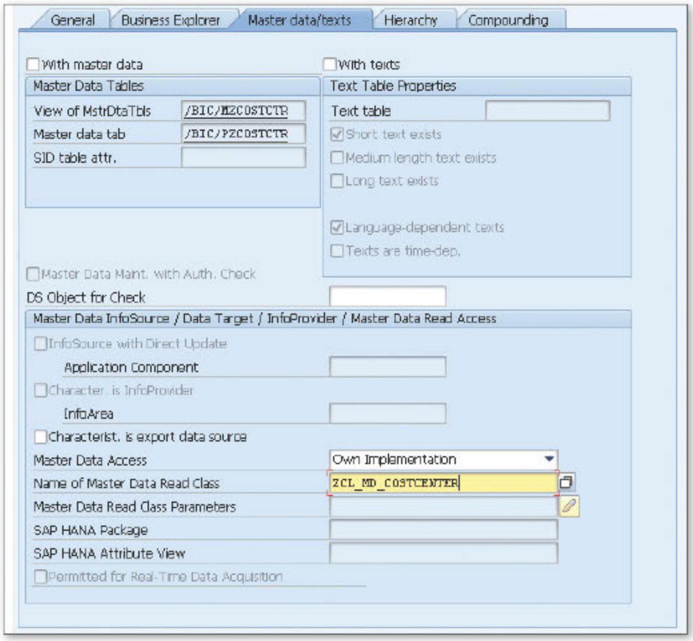
Figure 2 Defining the Superclass

5. Include the following code in the method. The code will have logic to bypass the SID validation check for the InfoObject.

```
l_s_chavlinfo-i_read_mode = rsdm_c_read_mode-text.
LOOP AT o_t_selopt ASSIGNING <l_s_selopt>.

  l_s_chavlinfo-c_chavl = <l_s_selopt>-low.
  APPEND l_s_chavlinfo TO e_t_chavlinfo.

ENDLOOP.

* H1403320
IF o_t_selopt IS NOT INITIAL.
  o_maxrows = 10000.
ENDIF.
```

6. Configure the InfoObject to read the class you just created.

7. Go to Transaction RSD1 for InfoObject maintenance, enter the InfoObject name "0COSTCENTER", and click on the CHANGE option. In the next screen, go to the MASTER DATA/TEXTS tab, and select OWN IMPLEMENTATION from the MASTER DATA ACCESS dropdown (see Figure 3).
8. Enter the NAME OF MASTER DATA READ CLASS you just created, and then save and activate the InfoObject.



☐ With master data
☐ With texts

Master Data Tables	
View of MstrDtaTbls	/BIC/MZCOSTCTR
Master data tab	/BIC/PZCOSTCTR
SID table attr.	

Text Table Properties	
Text table	

☑ Short text exists
☐ Medium length text exists
☐ Long text exists
☑ Language-dependent texts
☐ Texts are time-dep.

☐ Master Data Maint. with Auth. Check

DS Object for Check	

Master Data InfoSource / Data Target / InfoProvider / Master Data Read Access

☐ InfoSource with Direct Update

Application Component	

☐ Character. is InfoProvider

InfoArea	

☐ Characterist. is export data source

Master Data Access	Own Implementation
Name of Master Data Read Class	ZCL_MD_COSTCENTER
Master Data Read Class Parameters	
SAP HANA Package	
SAP HANA Attribute View	

☐ Permitted for Real-Time Data Acquisition

Figure 3 Adding the Custom Class to the InfoObject 0COSTCENTER

The system now allows users to enter the values of a cost center even if it isn't available in the master data without causing an error. Using this approach, you can implement a scenario to allow users to enter planning data for master data values that don't exist.

Tip 19

Reporting on Base Tables Using Transient Providers

You can avoid the need for system development to build reports on a base table with BEx Query Designer by using the more flexible transient provider object in two main steps.

Sometimes you'll need to report on a base table in SAP NetWeaver BW using a BEx query. This requirement to report on a base table is very common for technical support folks where reports on base tables can be built to report on internal SAP NetWeaver BW system tables with log, security, or statistics information. The business users might also need to report on some base tables such as currency or unit of measurement tables or some custom tables created to support specific functionality.

However, BEx queries can only be created on standard SAP NetWeaver BW InfoProviders such as InfoCubes, DSOs, or MultiProviders. Furthermore, this process requires significant development effort to create the metadata and SAP NetWeaver BW objects such as InfoObjects, InfoProviders, and transformations and cannot be implemented in an agile manner.

This tip describes how these types of scenarios can be implemented using transient providers on top of the database table in SAP NetWeaver BW. The advantage of using a transient provider is that it doesn't have any persistent SAP NetWeaver BW metadata, so it can be implemented in an agile manner without creating all of the metadata required for regular SAP NetWeaver BW modeling.

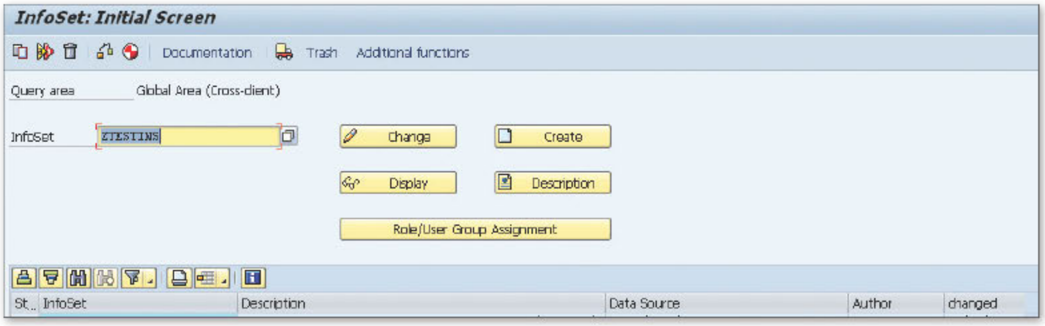
And Here's How ...

To illustrate this tip, we'll use a simple scenario where the requirement is to report on the currency exchange rate Table TCURR. This scenario can be implemented using a two-step approach. You'll first create a classic InfoSet on top of Table TCURR and then convert the classic InfoSet to a transient provider.

Step 1

Follow these steps to create a classic InfoSet on Table TCURR:

1. Log in to the SAP NetWeaver BW system. Execute Transaction SQ02, and the screen shown in Figure 1 appears.
2. Enter a technical name for the InfoSet and click CREATE.



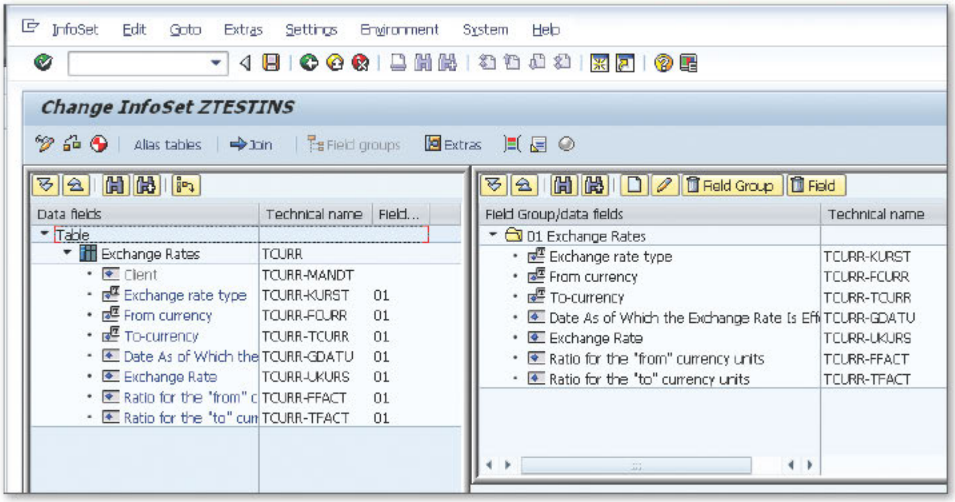
Query area Global Area (Cross-client)

InfoSet ZTESTINS

Figure 1 Transaction SQ02 to Create an InfoSet

3. In the INFOSET: TITLE AND DATABASE popup screen that appears, provide details on the InfoSet design:
   - If you want to create an InfoSet on multiple tables and join them, select the TABLE JOIN USING BASIS TABLE radio button. Use this option if you need to create InfoSet by joining multiple tables.
   - If you want to create an InfoSet directly on the table, select the DIRECT READ OF TABLE radio button.
4. Select the DIRECT READ OF TABLE radio button, and enter the name of the table, "TCURR". Click CONTINUE.
5. A popup appears with three different options for selecting the fields of the table that should be included in the InfoSet. Select the INCLUDE ALL TABLE FIELDS radio button, and click on CONTINUE.

6. The Change InfoSet ZTESTINS screen appears as shown in Figure 2; you can see the fields on the right side.



Data fields	Technical name	Field...
Table		
Exchange Rates	TCURR	
Client	TCURR-MANDT	
Exchange rate type	TCURR-KURST	01
From currency	TCURR-FCURR	01
To-currency	TCURR-TCURR	01
Date As of Which the	TCURR-GDATU	01
Exchange Rate	TCURR-UKURS	01
Ratio for the "from" c	TCURR-FFACT	01
Ratio for the "to" curr	TCURR-TFACT	01

Field Group/data fields	Technical name
01 Exchange Rates	
Exchange rate type	TCURR-KURST
From currency	TCURR-FCURR
To-currency	TCURR-TCURR
Date As of Which the Exchange Rate Is Effi	TCURR-GDATU
Exchange Rate	TCURR-UKURS
Ratio for the "from" currency units	TCURR-FFACT
Ratio for the "to" currency units	TCURR-TFACT

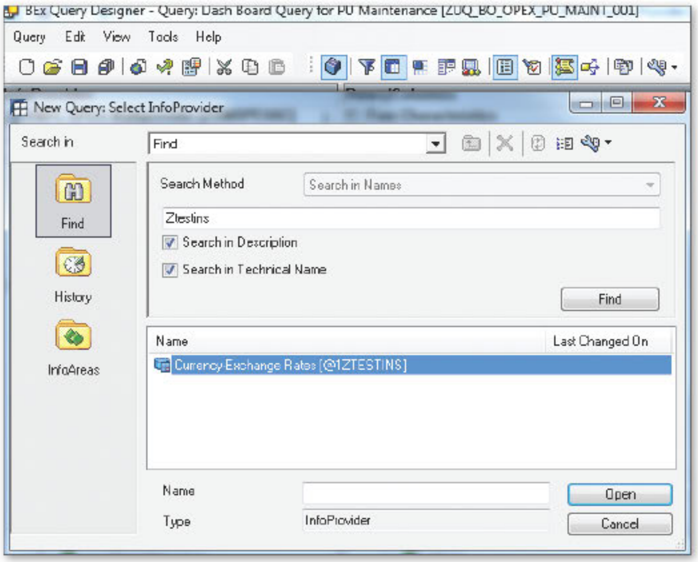
Figure 2 Displaying All the Fields from Table TCURR in the InfoSet

7. Click on the Save button. Generate the InfoSet by clicking on the Generate icon, which is the third icon from the top left of the screen.

Step 2

To convert the InfoSet into a transient provider, go to Transaction SQBWPROP in SAP NetWeaver BW; you'll see the InfoSet just created on the screen. Check the checkbox beside InfoSet ZTESTINS, and click on the Save and Generate button at the top of the screen (see Figure 3). A message appears confirming the generation of the InfoSet.

A transient provider is now created and will be available for reporting in the BEx Query Designer. The transient provider appears with a prefix of "@1" in the BEx Query Designer as shown in Figure 3.


BEx Query Designer - Query: Dash Board Query for PU Maintenance [ZUQ_BO_OPEX_PU_MAINT_001]
Query
Edit
View
Tools
Help
New Query: Select InfoProvider
Search in
Find
History
InfoAreas
Find
Open
Cancel

Search Method: Search in Names
Ztestins
☑ Search in Description
☑ Search in Technical Name

Name	Last Changed On
Currency Exchange Rates [@1ZTESTINS]	

Name:
Type: InfoProvider

Figure 3 BEx Query Designer Showing the Generated Transient Provider

This transient provider will support all analysis and reporting features, including SAP NetWeaver BW OLAP functionalities. You can now create BEx queries on top of this transient provider.

Tip 20

Allowing Flexible Calculations on Date and Time Fields

Using simple configuration changes and minimal coding, you can derive time-based metrics on the time characteristics for your specific reporting needs.

Some reporting scenarios require information from metrics that are based on calculations on date and time fields, such as finding the elapsed time between events. However, 0DATE and 0TIME characteristics are not flexible enough to perform calculations, and usually developers will have to write extensive code for derivations based on date and time fields.

This tip describes a simple approach with minimal coding to enable flexible calculations on date and time fields and to derive time-based metrics on an SAP NetWeaver BW system.

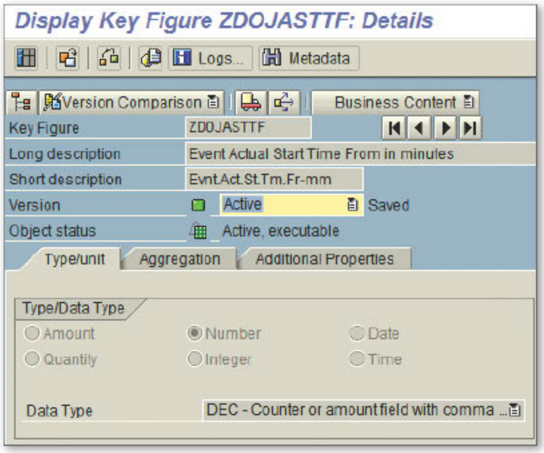
✓ And Here's How ...

Let's assume that we want to use the four date and time fields available in the source DSO that represents actual event start and actual end date and times in calculations such as finding the elapsed time between the dates or adding or subtracting the date and time.

To perform derivations based on time characteristics, follow these steps:

1. Using Transaction RSA1, create two new key figures:
   - Event Actual Start Time From in minutes
   - Event Actual End Time To in minutes

2. Set the TYPE/DATA TYPE as NUMBER, and set the DATA TYPE option as DEC - COUNTER OR AMOUNT FIELD WITH COMMA AND SIGN (see Figure 1).



Display Key Figure ZDOJASTTF: Details

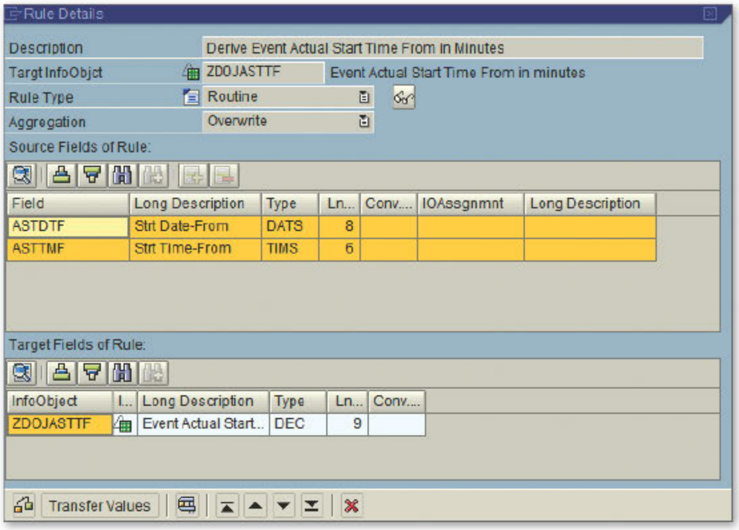
Key Figure	ZDOJASTTF
Long description	Event Actual Start Time From in minutes
Short description	Evnt.Act.St.Tm.Fr-mm
Version	Active Saved
Object status	Active, executable

Type/Data Type: Amount, Quantity, [✓ Number], Integer, Date, Time

Data Type: DEC - Counter or amount field with comma ...

« Figure 1 Example of Key Figure for Time of Data Type "DEC"

3. In the transformation routine, write an ABAP routine to concatenate the START DATE FROM (as shown in Figure 2) and START TIME FROM fields into a `t_date-time_to` field defined as a numeric field with length 14 (e.g., to concatenate the date and time value as `YYYYMMDDHHMMSS`). Define the reference date as a constant in the ABAP routine (e.g. `c_julian_ref_dt(14) type n value '19910101000000'`).



Rule Details

Description	Derive Event Actual Start Time From in Minutes	
Targt InfoObjct	ZDOJASTTF	Event Actual Start Time From in minutes
Rule Type	Routine	
Aggregation	Overwrite	

Source Fields of Rule:

Field	Long Description	Type	Ln...	Conv....	IOAssgnmnt	Long Description
ASTDTF	Strt Date-From	DATS	8			
ASTTMF	Strt Time-From	TIMS	6			

Target Fields of Rule:

InfoObject	I...	Long Description	Type	Ln...	Conv....
ZDOJASTTF		Event Actual Start...	DEC	9	

⮝ Figure 2 Transformation Routine for ZDOJASTTF Event Actual Start Time From in Minutes

Code Logic

To facilitate ease of date/time differences in a BEx query, you can store the date and time or timestamp field in minutes. This can then be easily used in calculated key figures to find date/time differences. To accomplish this, first you need to convert the date/time or timestamp field to minutes using a reference date and time. This reference date is picked as a date/time in the distant past so that the reference date will always be older than any date value expected to be stored in SAP NetWeaver BW. For this illustration, we'll use the reference date of `19910101000000`.

In the ABAP routine, define the necessary data declarations and call function module RSSM_SUBSTRACT_TIMESTAMPS by passing `t_datetime_to` and `c_julian_ref_dttm` as input parameters. The function module will calculate the time difference between reference date/time and the event start date/time as a numeric value in minutes.

See the following sample code to calculate the event start time in minutes:

```
*-----------------------------------------------------------------
-----------------------------------*
* Using the reference timestamp of "19910101000000", find out the
* difference between the Event Start Time and the reference date/
time
* using the function module RSSM_SUBSTRACT_TIMESTAMPS.
* Convert the difference in days and hh:mm:ss into minutes.
*-----------------------------------------------------------------
-----------------------------------*
    data:      t_datetime_to(14) type n,
    data:      t_time_to like SOURCE_FIELDS-ASTTMF,
    data:      c_julian_ref_dttm(14) type n value
'19910101000000',
    data:      t_no_days type i,
    data:      t_time(6) type n,
    data:      t_no_hours(6) type n,
    data:      t_no_min(6) type n,
    data:      t_no_days_in_mm(14) type n,
    data:      t_no_hrs_in_mm(14) type n.

    if SOURCE_FIELDS-astdtf IS NOT INITIAL.
```

```
      concatenate SOURCE_FIELDS-astdtf
         SOURCE_FIELDS-asttmf
             into t_datetime_to.

      CALL FUNCTION 'RSSM_SUBSTRACT_TIMESTAMPS'
        EXPORTING
          TIMESTAMP1 = t_datetime_to
          TIMESTAMP2 = c_julian_ref_dttm
       IMPORTING
         TIME = t_time_to
         DAYS  = t_no_days.

      if sy-subrc = 0.
        t_time = t_time_to.
        t_no_days_in_mm = t_no_days * 1440.
        t_no_hours      = t_time+0(2).
        t_no_min        = t_time+2(2).
        t_no_hrs_in_mm  = t_no_hours * 60 + t_no_min.
        RESULT = t_no_days_in_mm + t_no_hrs_in_mm.
      else.
        RESULT = 0.
      endif.
    else.
      RESULT = 0.
        endif.
```

Because the date and time will be stored as minutes using a numeric field, the delta updates to this field will be handled correctly as any normal numeric key figure in the DSO.

Part 2

SAP NetWeaver BW Reporting and Analysis

Things You'll Learn in this Section

This section will provide you with tips and tricks that will help you to implement, execute, and optimize your reporting and analysis functions in the SAP NetWeaver BW Business Explorer tool.

First, we will cover some basic topics to get you working quickly with the BEx Query Designer. Then we will discuss methods you can use to optimize reporting and analysis functions such as query pruning and using exception aggregation in BEx. We'll also look at more advanced topics such as reporting on SAP ERP data using virtual providers and integrating custom formulas into the formula builder.

Tip 21

Determining Which Queries Are Being Used in a Workbook

You can use Transaction RRMX_WORKBOOK_QUERIES_GET to find out which queries are being used in a workbook without opening it in design mode.

You have the ability to create a workbook with one or more queries. After the creation and during maintenance, you might need to get a list of queries that were used to build that workbook to support housekeeping activities on the queries. Determining which queries are being used without opening up the workbook in design mode may become complicated, especially if the workbook holds multiple design elements. Even in the workbook design mode, it's time consuming to get a list of queries used in the workbook. Also, in some cases, not everyone has access to the workbook in design mode.

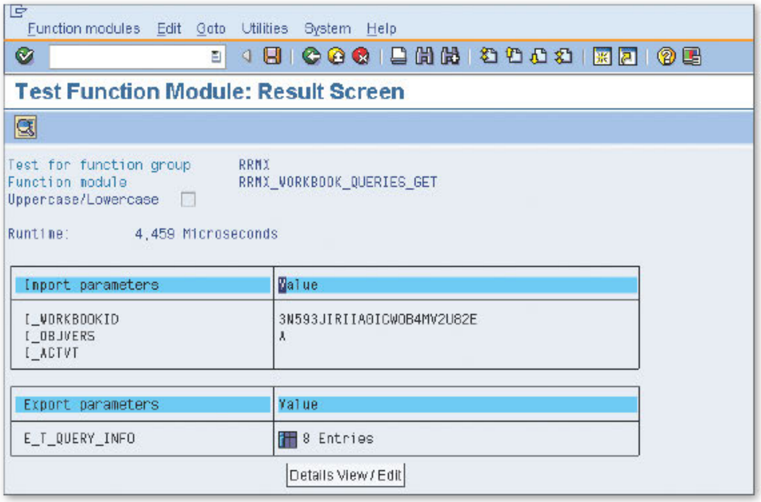
This tip describes how Transaction RRMX_WORKBOOK_QUERIES_GET provides a simpler way to find out which queries are being used in the workbook.

✓ And Here's How ...

After you determine the technical name of the workbook (you can use Table RSR-WBINDEXT via Transaction SE16), follow these steps:

1. Specify the OBJVERS as "A" in the VERSION column to select active workbooks.
2. Look up the workbook by the title, and select the Workbook ID, for example, E 4EA7YQQ9B0S7CXEMXFB9DATGFA Deliveries, for that workbook.
3. Call up Transaction RRMX_WORKBOOK_QUERIES_GET. In the selection screen, specify the workbook ID (I_WORKBOOKID) as determined in Step 1. Also specify the version (I_OBJVERS) as active (A).

4. Execute the query. On the subsequent screen (shown in Figure 1), click on the values of parameter E_T_QUERY_INFO to see the queries that belong to the workbook.



Test Function Module: Result Screen

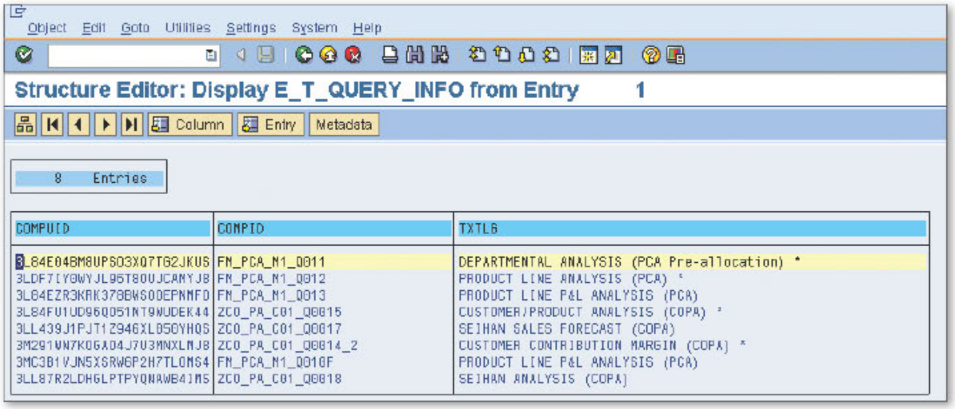
```
Test for function group      RRMX
Function module              RRMX_WORKBOOK_QUERIES_GET
Uppercase/Lowercase    ☐

Runtime:        4,459 Microseconds
```

Import parameters	Value
I_WORKBOOKID	3N593JIRIIA0ICWOB4MV2U82E
I_OBJVERS	A
I_ACTVT	

Export parameters	Value
E_T_QUERY_INFO	8 Entries

Figure 1 Output of the Function Module RRMX_WORKBOOK_QUERIES_GET

The results are shown in Figure 2. The list displays all the queries within the workbook.



Structure Editor: Display E_T_QUERY_INFO from Entry 1

8 Entries

COMPUID	COMPID	TXTLG
3L84E04BM8UPS03XQ7TG2JKUS	FM_PCA_M1_Q011	DEPARTMENTAL ANALYSIS (PCA Pre-allocation) *
3LDF7IY0WYJL95T80UJCANYJB	FM_PCA_M1_Q012	PRODUCT LINE ANALYSIS (PCA) *
3L84EZR3KRK378BWS0DEPNMFD	FM_PCA_M1_Q013	PRODUCT LINE P&L ANALYSIS (PCA)
3L84FU1UD96Q051NT9WUDEK44	ZCO_PA_C01_Q0015	CUSTOMER/PRODUCT ANALYSIS (COPA) *
3LL439J1PJT1Z946XL050YHQS	ZCO_PA_C01_Q0017	SEIHAN SALES FORECAST (COPA)
3M291WN7K0GAD4J7U3MNXLMJB	ZCO_PA_C01_Q0014_2	CUSTOMER CONTRIBUTION MARGIN (COPA) *
3MC3B1VJN5XSRW6P2H7TLOMS4	FM_PCA_M1_Q010F	PRODUCT LINE P&L ANALYSIS (PCA)
3LL87R2LDHGLPTPYQNAWB4IM5	ZCO_PA_C01_Q0018	SEIHAN ANALYSIS (COPA)

Figure 2 Output of RRMX_WORKBOOK_QUERIES_GET

Using this approach, you can get a listing of queries for any workbooks in the system.

Tip 22

Creating Ad Hoc Key Figures on the Fly and Saving Them in Workbooks

You can create local ad hoc key figures on a BEx workbook on the fly and save or distribute them to select users as required.

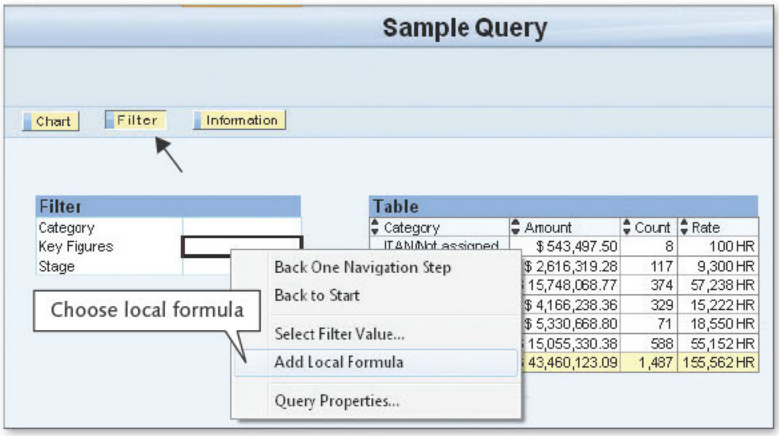
Sometimes, business users need to calculate a key figure for BEx workbooks on the fly and then use it as a part of a query, with the requirement being that it doesn't disappear when the query is refreshed. You can't create a formula in the query definition because this would result in a global change for every user who uses that query.

In this tip, we show you how to create the ad hoc key figure that will only be relevant to the creator, but is also able to be viewed by other users. You'll use a local formula feature that can be implemented without impacting the standard BEx query or workbook. This is very useful for customizing the workbook for a particular user need.

✓ And Here's How ...

To create the ad hoc key figure, follow these steps:

1. Open a BEx query with a few key figures in the column. As shown in Figure 1, choose the FILTER button in the top, and right-click on the key figures. Choose the ADD LOCAL FORMULA option, and add a new local formula (e.g., Revenue).

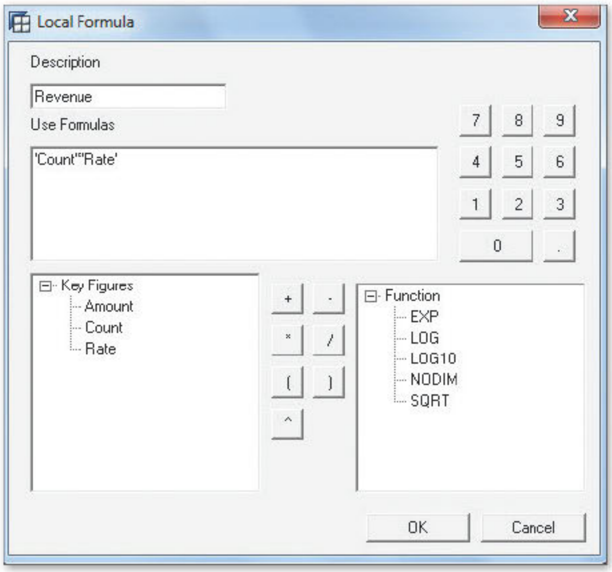


Table

Category	Amount	Count	Rate
ITAN/Not assigned	$ 543,497.50	8	100 HR
	$ 2,616,319.28	117	9,300 HR
	15,748,068.77	374	57,238 HR
	$ 4,166,238.36	329	15,222 HR
	$ 5,330,668.80	71	18,550 HR
	15,055,330.38	588	55,152 HR
	43,460,123.09	1,487	155,562 HR

Figure 1 BEx Query Results Screen

2. Fill in the REVENUE formula details as shown in Figure 2.



Description

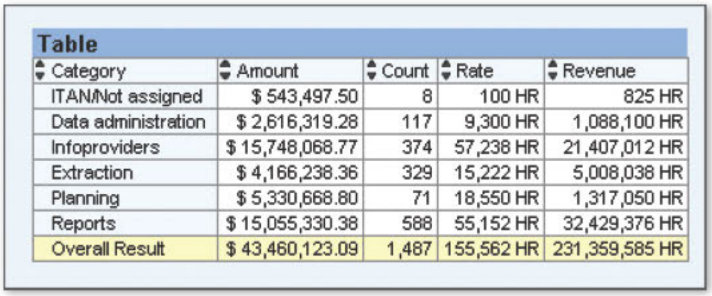
Revenue

Use Fomulas

'Count'*'Rate'

Figure 2 Defining Local Formula

3. Click OK to see the formula added to the query (see Figure 3).



Table

Category	Amount	Count	Rate	Revenue
ITAN/Not assigned	$ 543,497.50	8	100 HR	825 HR
Data administration	$ 2,616,319.28	117	9,300 HR	1,088,100 HR
Infoproviders	$ 15,748,068.77	374	57,238 HR	21,407,012 HR
Extraction	$ 4,166,238.36	329	15,222 HR	5,008,038 HR
Planning	$ 5,330,668.80	71	18,550 HR	1,317,050 HR
Reports	$ 15,055,330.38	588	55,152 HR	32,429,376 HR
Overall Result	$ 43,460,123.09	1,487	155,562 HR	231,359,585 HR

Figure 3 Query Results Showing the Newly Created Local Formula for Revenue

Now the revenue is an integral part of the query and can be drilled down and filtered as a part of the query. This query can now be saved in a local workbook and reused whenever necessary; it also can be distributed to other users.

Once created, you can edit/delete the local formula by right-clicking on the name, and then choosing CHANGE REVENUE or DELETE REVENUE (this option will change depending on the name given to the formula). You also have the option to create multiple local formulas.

Tip 23

Using Alternate Unit of Measures for Key Figures in BEx Reporting

You can set up the system to automatically translate key figures that have a unit field into a target unit amount at runtime during query execution.

In BEx reporting, a common business requirement is to be able to view key figures based on multiple units of measure (UOMs) such as units, pounds, ounces, and kilograms. One approach is to convert the key figures in the backend in the transformations and store the converted key figures in the InfoCubes or DSOs. This method requires a lot of development work and coding that can be avoided by doing the conversion at the query runtime.

To meet this need, SAP has introduced many new features in SAP NetWeaver BW 7.x, including Transaction RSUOM, which maintains UOM conversion types for BEx reporting. You can use these features to translate key figures with a unit field into a target unit at runtime during query execution.

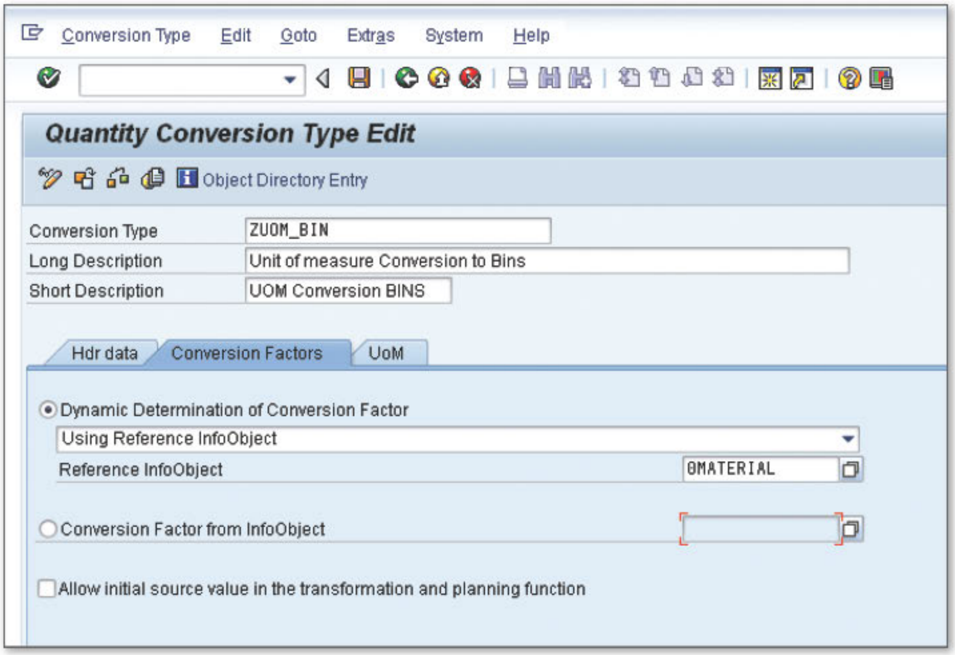
✓ And Here's How ...

To use the new tools in SAP NetWeaver BW, follow these steps:

1. Access Transaction RSUOM. On the resulting screen, specify the new quantity conversion type and click CREATE.
2. On the next screen, you have three different tabs for configuring the conversion type. In the HEADER tab, you can provide the description of the conversion type. Provide the long and short descriptions.

3. In the Conversion Factors tab, you have two options for deriving the conversion factors:
   - Dynamic Determination of Conversion Factor
     With this indicator, you can determine how values are converted from one UOM to another. If this indicator is set, the system first tries to determine the InfoObject or material-specific factors. If these cannot be determined, the system checks Table T006 to see whether source and target quantity unit are part of a dimension and whether a conversion can be made using function module UNIT_CONVERSION_SIMPLE.
   - Conversion Factor from InfoObject
     With this option, the conversion factor is taken from the associated key figure and is not calculated using the data at runtime.

   For our scenario, we choose Dynamic Determination of Conversion Factor because the requirement is to calculate UOM at runtime (see Figure 1).



Conversion Type: ZUOM_BIN
Long Description: Unit of measure Conversion to Bins
Short Description: UOM Conversion BINS

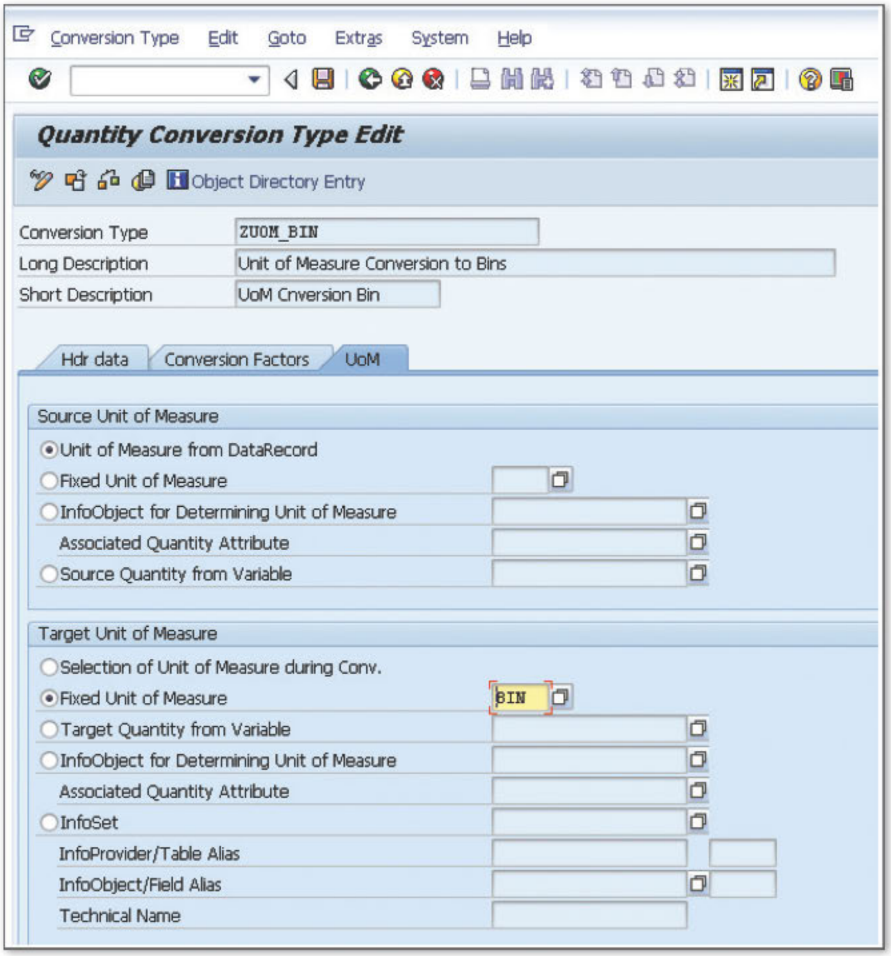
(●) Dynamic Determination of Conversion Factor
Using Reference InfoObject
Reference InfoObject: 0MATERIAL

( ) Conversion Factor from InfoObject

☐ Allow initial source value in the transformation and planning function

Figure 1 Defining the Reference Object for Dynamic Conversion

4. From the dropdown, select Using Reference InfoObject. In this example, the reference object is 0MATERIAL. The alternative UOM is to be referenced from the material master.
5. In the UoM tab (see Figure 2), you can specify the source and target UOMs to be converted:
   - Select Unit of Measure from DataRecord as the source UOM, and select Fixed Unit of Measure as the target UOM.
   - Provide a valid UOM as the target UOM (e.g., "BIN").



Quantity Conversion Type Edit

Conversion Type	ZUOM_BIN
Long Description	Unit of Measure Conversion to Bins
Short Description	UoM Cnversion Bin

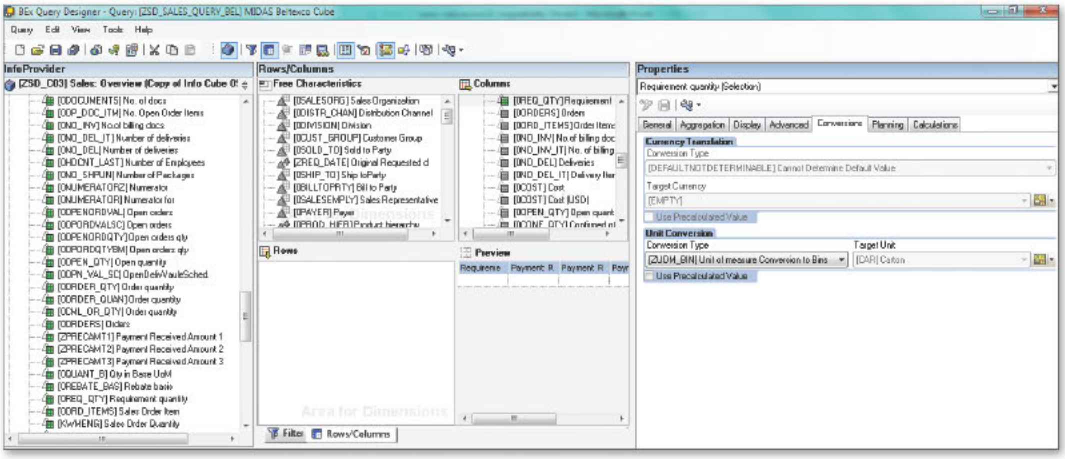
Source Unit of Measure
- [✓ Unit of Measure from DataRecord]
- Fixed Unit of Measure
- InfoObject for Determining Unit of Measure
- Associated Quantity Attribute
- Source Quantity from Variable

Target Unit of Measure
- Selection of Unit of Measure during Conv.
- [✓ Fixed Unit of Measure] BIN
- Target Quantity from Variable
- InfoObject for Determining Unit of Measure
- Associated Quantity Attribute
- InfoSet
- InfoProvider/Table Alias
- InfoObject/Field Alias
- Technical Name

Figure 2 Specifying Source and Target UOMs to Be Converted

Now you can use the quantity conversion type in your BEx query. Follow these steps:

1. Open an existing BEx query or create a new one, and select the key figure to be converted.
2. Select the properties of the key figure, and go to the CONVERSIONS tab as shown in Figure 3. Under UNIT CONVERSION, select the newly created conversion type.


BEx Query Designer - Query: [ZSD_SALES_QUERY_BEL] MIDAS Beltexco Cube
Query Edit View Tools Help
InfoProvider
Rows/Columns
Free Characteristics
Columns
Rows
Preview
Filter
Rows/Columns
Properties
General Aggregation Display Advanced Conversions Planning Calculations

[ZSD_C03] Sales: Overview (Copy of Info Cube 0!

[0DOCUMENTS] No. of docs
[0OP_DOC_ITM] No. Open Order Items
[0NO_INV] No.of billing docs
[0NO_DEL_IT] Number of deliveries
[0NO_DEL] Number of deliveries
[0HDCNT_LAST] Number of Employees
[0NO_SHPUN] Number of Packages
[0NUMERATORZ] Numerator
[0NUMERATOR] Numerator for
[0OPENORDVAL] Open orders
[0OPORDVALSC] Open orders
[0OPENORDQTY] Open orders qty
[0OPORDQTYBM] Open orders qty
[0OPEN_QTY] Open quantity
[0OPN_VAL_SC] OpenDelivVauleSched
[0ORDER_QTY] Order quantity
[0ORDER_QUAN] Order quantity
[0CML_OR_QTY] Order quantity
[0ORDERS] Orders
[ZPRECAMT1] Payment Received Amount 1
[ZPRECAMT2] Payment Received Amount 2
[ZPRECAMT3] Payment Received Amount 3
[0QUANT_B] Qty in Base UoM
[0REBATE_BAS] Rebate basis
[0REQ_QTY] Requirement quantity
[0ORD_ITEMS] Sales Order Item
[KWMENG] Sales Order Quantity

Free Characteristics:
[0SALESORG] Sales Organization
[0DISTR_CHAN] Distribution Channel
[0DIVISION] Division
[0CUST_GROUP] Customer Group
[0SOLD_TO] Sold to Party
[ZREQ_DATE] Original Requested d
[0SHIP_TO] Ship toParty
[0BILLTOPRTY] Bill to Party
[0SALESEMPLY] Sales Representative
[0PAYER] Payer
[0PROD_HIER] Product hierarchy

Columns:
[0REQ_QTY] Requirement
[0ORDERS] Orders
[0ORD_ITEMS] Order Items
[0NO_INV] No.of billing doc
[0NO_INV_IT] No. of billing
[0NO_DEL] Deliveries
[0NO_DEL_IT] Delivery Iter
[0COST] Cost
[0COST] Cost (USD)
[0OPEN_QTY] Open quant
[0CONF_QTY] Confirmed qt

Preview: Requireme | Payment R | Payment R | Payr

Properties: Requirement quantity (Selection)

Currency Translation
Conversion Type: [DEFAULTNOTDETERMINABLE] Cannot Determine Default Value
Target Currency: [EMPTY]
☐ Use Precalculated Value

Unit Conversion
Conversion Type: [ZUOM_BIN] Unit of measure Conversion to Bins
Target Unit: [CAR] Carton
☐ Use Precalculated Value

Figure 3 Query Definition and Key Figure Properties

When this final setting is complete, the query will dynamically use the conversion type at runtime.

Tip 24

Implementing BEx User Exit Variables Using a BAdI for en Masse Modification

You can enable multiple developers to make changes to BEx user exit variables at the same time by creating custom BAdI implementations.

Usually, BEx user-exit variable coding is maintained directly in the exit Program ZXRSRU01. This causes transport coordination issues and other challenges during development/maintenance of user-exit variables. In addition, multiple developers cannot simultaneously modify the code for user exits. To overcome these issues, you can create BAdI implementations and write code to call the BAdI in `EXIT_SAPLRRS0_001`.

This tip provides step-by-step instructions for defining an enhancement spot and implementing a BAdI for user-exit variables in BEx. Modularization of each customer user-exit variable coding as separate BAdI implementations helps to avoid all the previously mentioned issues, and developers can work on a specific user-exit variable coding without affecting others.

✓ And Here's How ...

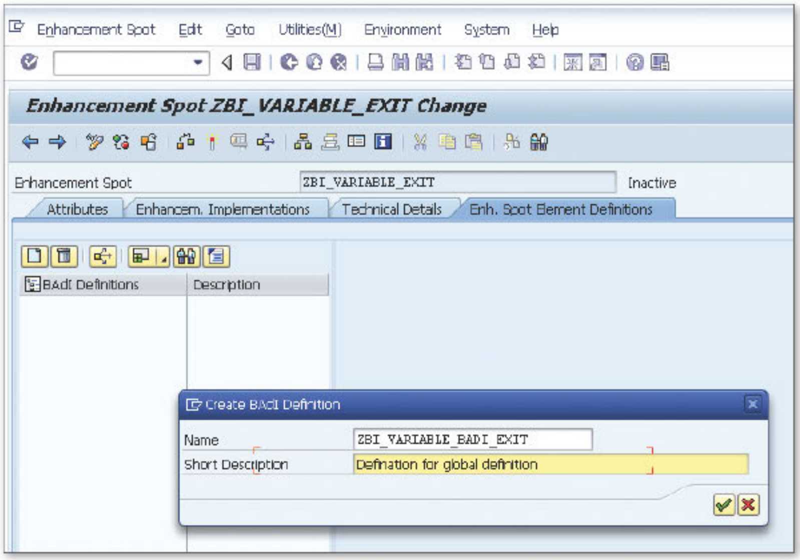
Before getting started with this tip, note that we're assuming you have basic knowledge of programming using ABAP Objects and understand the concepts of methods and classes.

Modularization using a BAdI is achieved by implementing a custom BAdI definition in the exit Program ZXRSRU01 and any required coding for each customer exit variable needed to be added in the BAdI implementation with a filter for specific user-exit variables. Implementing a BEx user exit using a BAdI implementation is a three-step process.

Step 1

Create an enhancement spot for the BAdI to specify the places where you can add your code in standard SAP programs. An enhancement spot is an object that can contain one or more BAdIs. To add/create coding to a BEx customer exit variable, go to Transaction SE19 and follow these steps:

1. Create an enhancement spot for the BAdI. Enter the name "ZBI_VARIABLE_EXIT" in the Enhancement Spot field. Then click on the Create button, and the system prompts you to enter the text and some additional information on the enhancement spot.
2. After you are in the Enhancement Spot Change screen, go to the Enh. Spot Element Definitions tab. Under this tab, click on the Create BAdI Definition option on the left side.
3. In the pop-up window, enter the technical name "ZBI_VARIABLE_BADI_EXIT" and a description for BAdI definition as shown in Figure 1.



Enhancement Spot ZBI_VARIABLE_EXIT Change

Enhancement Spot ZBI_VARIABLE_EXIT Inactive

Create BAdI Definition

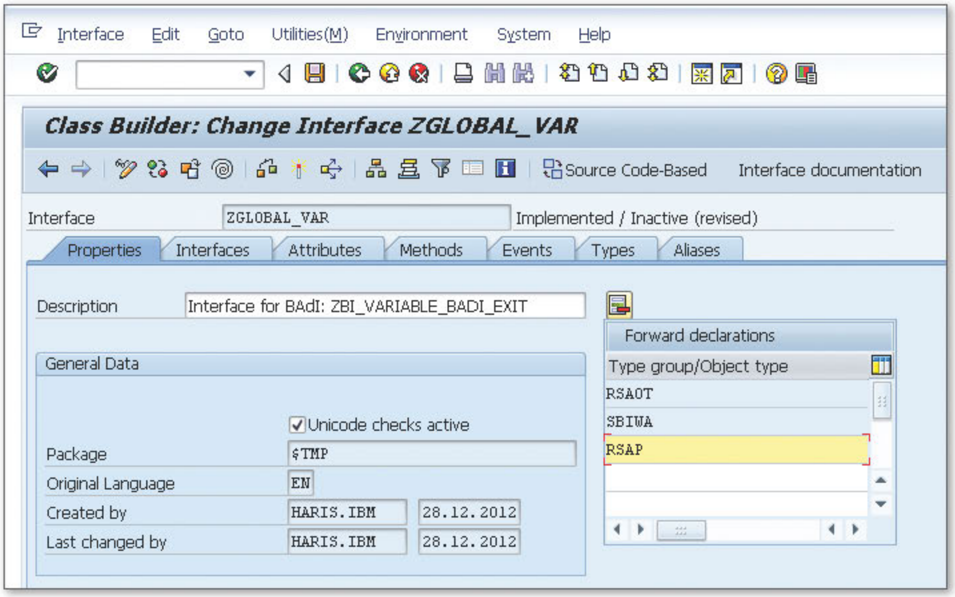
Name ZBI_VARIABLE_BADI_EXIT

Short Description Defination for global definition

Figure 1 Defining the Filter

Before creating the filter, go to Transaction SE11 to define dictionary object ZG_VAR. ZVAR_NAM as a domain object with data type as char and length of 30, which should be sufficient to hold the technical name of the variable.

4. Create the interface by entering a technical name and clicking on CHANGE.
5. Enter a method and description, and select INSTANCE METHOD for LEVEL.
6. Under the PROPERTIES tab, enter TYPE GROUP/OBJECT TYPE as needed and keep the other tabs unchanged (see Figure 2).



Class Builder: Change Interface ZGLOBAL_VAR

Interface ZGLOBAL_VAR Implemented / Inactive (revised)

Description: Interface for BAdI: ZBI_VARIABLE_BADI_EXIT

☑ Unicode checks active

Package	$TMP	
Original Language	EN	
Created by	HARIS.IBM	28.12.2012
Last changed by	HARIS.IBM	28.12.2012

Type group/Object type
RSAOT
SBIWA
RSAP

Figure 2 Create a BAdI for a Variable

Step 2

Create an individual BAdI implementation for each user-exit variable by following these steps:

1. Right-click on the IMPLEMENTATION option under the BAdI ZBI_VARIABLE_BADI_EXIT and click on CREATE BADI IMPLEMENTATION.
2. Enter the technical name and description of the user exit variable that requires the user exit logic. In this case, we'll use an already existing variable ZBI_LAST_DAY that gets the last day of the month in a query.

Step 3

After the enhancement implementation is created, the next step is to assign a customer exit variable name to the filter and add the code for the variable:

1. Click on FILTER VAL., and select a new combination.
2. Click on combination 1, and assign the technical name (ZFDAY) of the customer exit variable to the filter. Click on IMPLEMENTING CLASS, select the method on the right side, and choose YES.
3. In the include of `EXIT_SAPLRRS0_001` (enhancement project in CMOD), add the following code to call the BAdI:

```
DATA: badi_glob TYPE REF TO ZBW_VAR_BADI_EXIT
GET BADI badi_glob
  FILTERS
    zg_var = i_vnam.
CALL BADI badi_glob ->enhance
  EXPORTING
    flt_val       = i_vnam
    i_vnam        = i_vnam
    i_vartyp      = i_vartyp
    i_iobjnm      = i_iobjnm
    i_s_cob_pro   = i_s_cob_pro
    i_s_rkb1d     = i_s_rkb1d
    i_periv       = i_periv
    i_t_var_range = i_t_var_range
    i_step        = i_step
  CHANGING
    e_t_range     = e_t_range
    e_meeht       = e_meeht
    e_mefac       = e_mefac
    e_waers       = e_waers
    e_whfac       = e_whfac
    c_s_customer  = c_s_customer.
```

Using this approach, you can create one BAdI implementation for each customer exit variable and write code in the BAdI. This enables multiple developers to write the code for user exits at the same time.

Tip 25

Counting Using Exception Aggregation in BEx

You can meet specific reporting requirements by using the exception aggregation feature in BEx to count the occurrences of a calculated key figure.

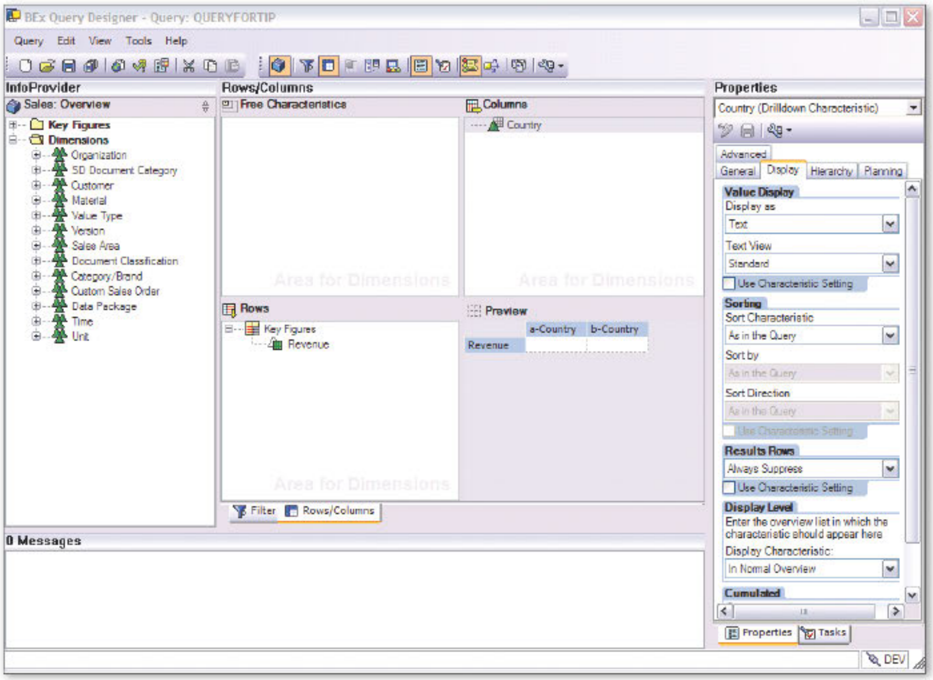
In SAP Business Explorer reporting, you'll likely need to determine counts based on common business questions, such as "How many customers have sales over $1 million?" or "How many purchase orders are over a certain amount?"

These scenarios can be built into your system using backend solutions by programming in the transformation. However, these calculations can be really complex sometimes. This tip describes an efficient and relatively simple method to count the occurrences of a calculated key figure using the exception aggregation feature in BEx.

✓ And Here's How ...

In this tip, we'll describe the exception aggregation feature by using a simple scenario, where the requirement is to group customers based on revenue and provide a count of customers by region based on the following revenues: less than 1 million, greater than 2 million, and greater than 3 million. Follow these steps:

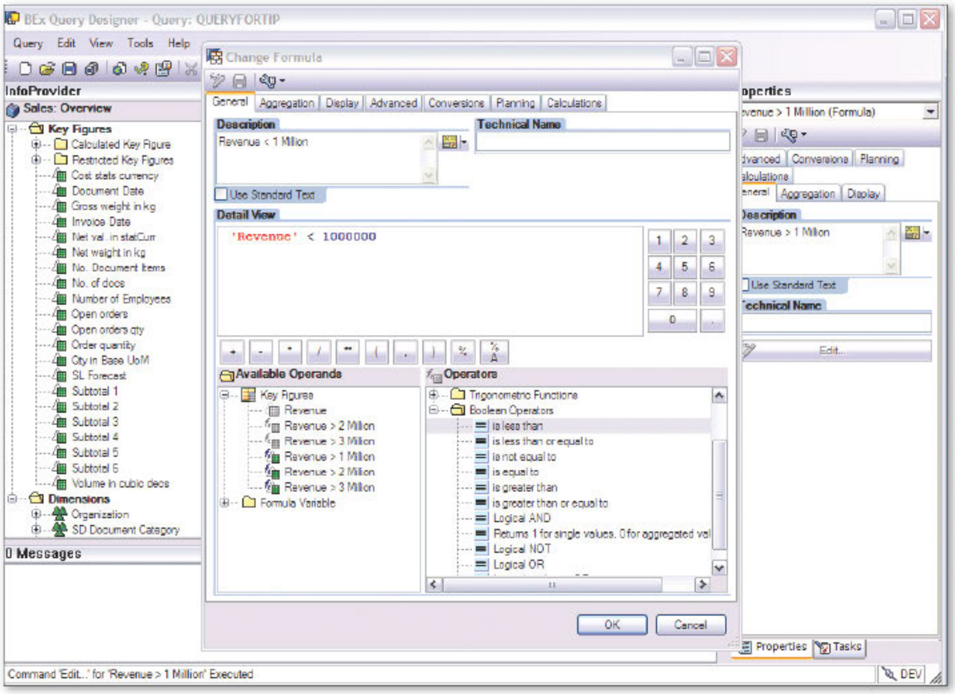
1. Create a query using an InfoProvider that contains the revenue key figure and customer and country characteristics. Place the revenue and country in Row/Columns as shown in Figure 1.



Value Display
Display as: Text
Text View: Standard
Use Characteristic Setting
Sorting
Sort Characteristic: As in the Query
Sort by: As in the Query
Sort Direction: As in the Query
Use Characteristic Setting
Results Rows: Always Suppress
Use Characteristic Setting
Display Level
Enter the overview list in which the characteristic should appear here
Display Characteristic: In Normal Overview
Cumulated

Figure 1 Query Definition

2. Create a formula with a condition for "revenue less than 1 million" by following these steps:
   - Right-click on the KEY FIGURES structure and choose NEW FORMULA. The NEW FORMULA entry is inserted, and the properties for the formula are displayed in the PROPERTIES screen area.
   - Make the basic settings on the GENERAL tab page by entering a description of the formula in the text field provided in the upper part of the screen.
3. Create the formula for "revenue less than 1 million" by following three steps (see Figure 2):
   - Drag and drop the REVENUE KEY FIGURE from the KEY FIGURES pane to the DETAIL VIEW pane.
   - On the right side of the screen, you see the functions that are available as operators. Choose the Boolean operator = IS LESS THAN, and insert it into the DETAIL VIEW field by double-clicking or dragging the operator.
   - Using the numeric pad, select the numerical values for this formula as 1000000.



Description: Revenue < 1 Million

Technical Name:

Use Standard Text

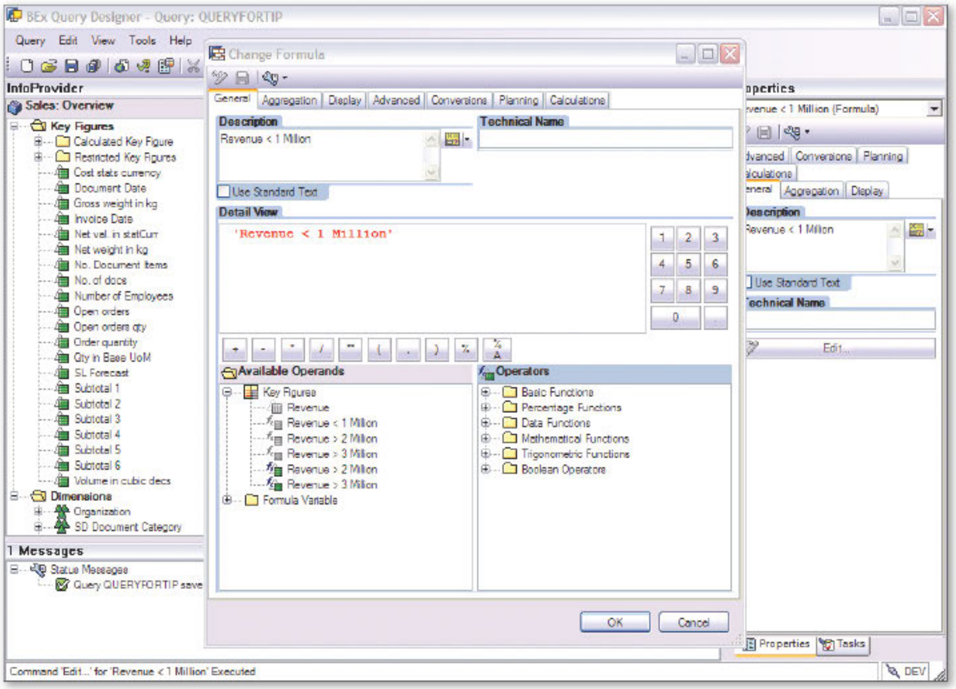
Detail View

```
'Revenue' < 1000000
```

Command 'Edit...' for 'Revenue > 1 Million' Executed

Figure 2 Creating a Formula in a BEx Query

4. Using the steps outlined previously, create two other formulas for "revenue greater than 2 but less than 3 million" and "revenue greater than 3 million." Select all key figures, go to the Display tab, and select Always Hide.
5. Create three additional formulas with the respective formula buckets (created in the previous steps) as shown in Figure 3.
6. Go to the Aggregation tab, and select the options as shown in Figure 4.
7. Save the query.
8. Upon execution of the query, the results shown in Figure 5 are displayed.



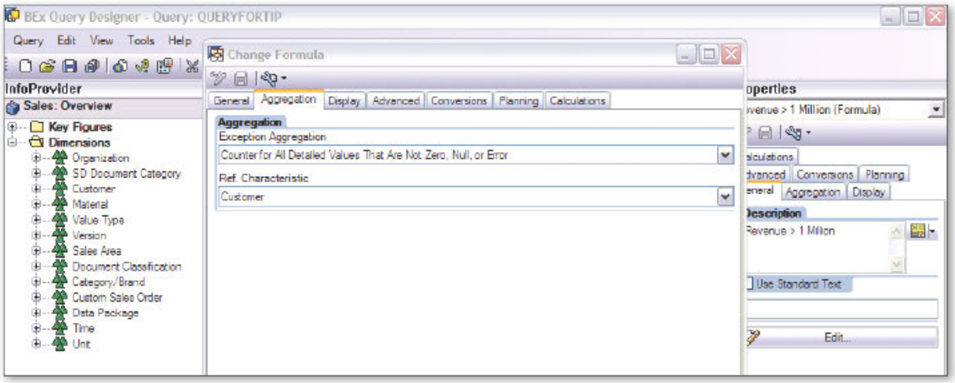
Description: Revenue < 1 Million

```
'Revenue < 1 Million'
```

Revenue < 1 Million (Formula)

Command 'Edit...' for 'Revenue < 1 Million' Executed

Figure 3 *Creating Additional Formulas*

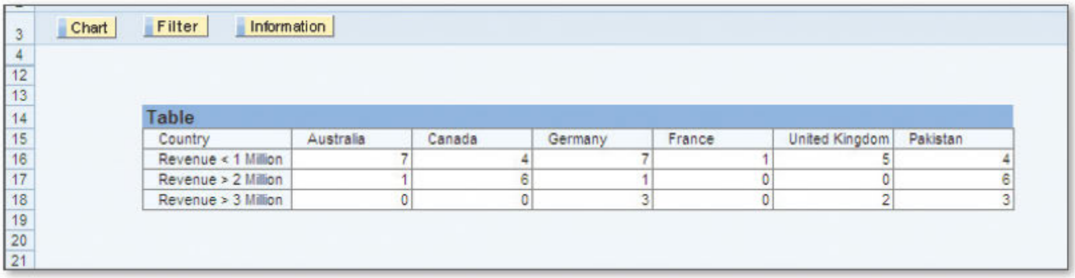


Aggregation
Exception Aggregation: Counter for All Detailed Values That Are Not Zero, Null, or Error
Ref. Characteristic: Customer

Revenue > 1 Million (Formula)

Description: Revenue > 1 Million

Figure 4 *Aggregation Tab Selections*



Table

Country	Australia	Canada	Germany	France	United Kingdom	Pakistan
Revenue < 1 Million	7	4	7	1	5	4
Revenue > 2 Million	1	6	1	0	0	6
Revenue > 3 Million	0	0	3	0	2	3

Figure 5 Query Results

The query results show three groups based on a range of revenue amounts and display the count of customers for each of the groups by region. Thus, we are able to count the occurrences of the key figure REVENUE. This method can be used to count occurrences for any condition of any key figure.

Tip 26

Copying and Creating Calculated and Restricted Key Figures to Use as a Template

To save time and effort, you can copy existing calculated and restricted key figures to use as a template for new key figures that have similar characteristics.

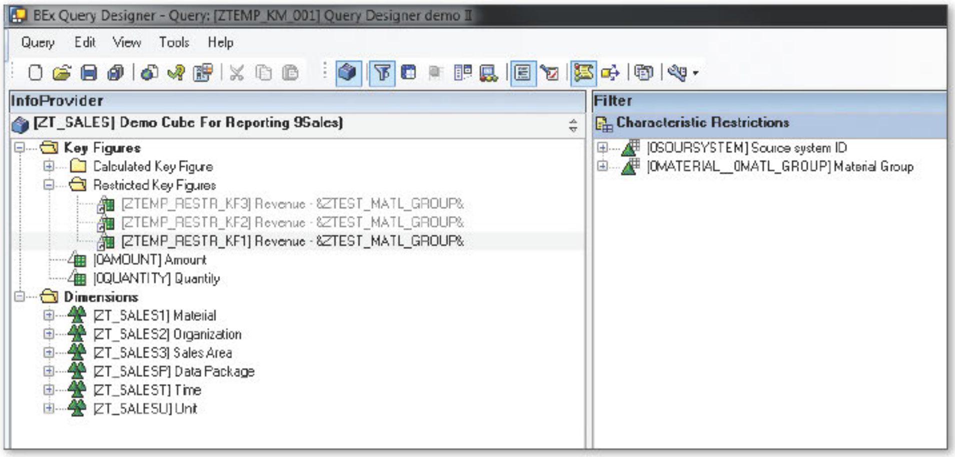
Often, project reporting scenarios require you to create a large amount of restricted and calculated key figures in the BEx Query Designer. In cases where many of these restricted/calculated key figures only have a slight variation based on restrictions on different characteristics, you can copy an existing restricted/calculated key figure to a new one instead of creating something from scratch.

This tip shows how to create new restricted and calculated key figures using existing restricted and calculated key figures as a template to save time.

✓ And Here's How ...

For this tip, we'll use an existing BEx query on a sales cube (ZT_SALES) that contains some restricted key figures. In the BEx query shown in Figure 1, we'll copy two of the three restricted key figures (ZTEMP_RESTR_KF1 and ZTEMP_RESTR_KF2) from cube ZTEMP_SALES as a template and create two new key figures. Once copied, we'll change the restrictions on the new key figures.

Figure 1 shows that there are three restricted key figures and two calculated key figures that already exist at the cube level.


BEx Query Designer - Query: [ZTEMP_KM_001] Query Designer demo II
Query
Edit
View
Tools
Help
InfoProvider
Filter
Characteristic Restrictions

[ZT_SALES] Demo Cube For Reporting 9Sales)

- Key Figures
  - Calculated Key Figure
  - Restricted Key Figures
    - [ZTEMP_RESTR_KF3] Revenue - &ZTEST_MATL_GROUP&
    - [ZTEMP_RESTR_KF2] Revenue - &ZTEST_MATL_GROUP&
    - [ZTEMP_RESTR_KF1] Revenue - &ZTEST_MATL_GROUP&
  - [0AMOUNT] Amount
  - [0QUANTITY] Quantity
- Dimensions
  - [ZT_SALES1] Material
  - [ZT_SALES2] Organization
  - [ZT_SALES3] Sales Area
  - [ZT_SALESP] Data Package
  - [ZT_SALEST] Time
  - [ZT_SALESU] Unit

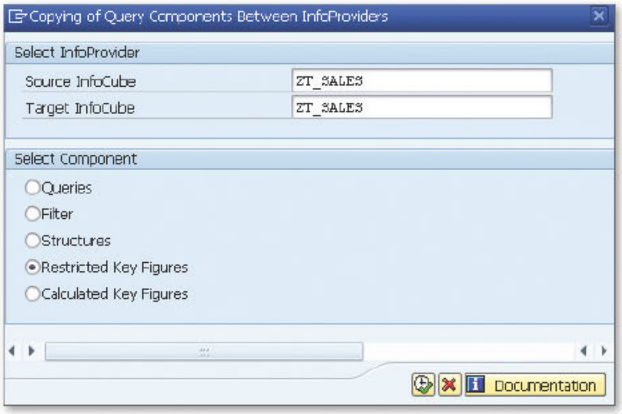
Filter – Characteristic Restrictions:

- [0SOURSYSTEM] Source system ID
- [0MATERIAL__0MATL_GROUP] Material Group

Figure 1 BEx Query Displaying the Calculated and Restricted Key Figures

Follow these steps:

1. Go to Transaction RSZC (see Figure 2). Enter the InfoCube or MultiProvider name for both Source InfoCube and the Target InfoCube.
2. Select the Restricted Key Figures or Calculated Key Figures radio button based on which object you need to create. Click Execute (F8).



Select InfoProvider

Source InfoCube	ZT_SALES
Target InfoCube	ZT_SALES

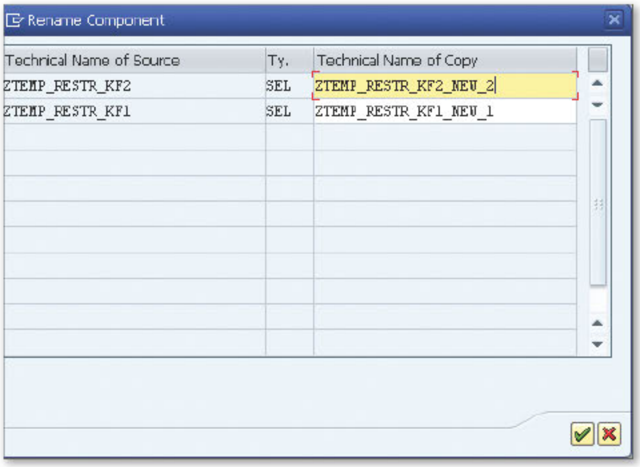
Select Component

○ Queries
○ Filter
○ Structures
● Restricted Key Figures
○ Calculated Key Figures

Figure 2 Transaction RSZC

3. In the Select Component screen, select the two existing restricted/calculated key figures that should be used as a template to create a new key figure, and click on Transfer selection.

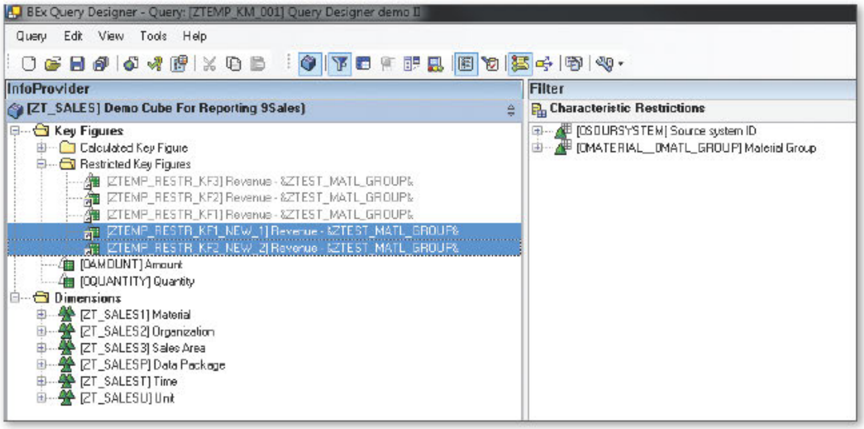
4. In the RENAME COMPONENT screen that appears (see Figure 3), enter new technical names for the key figures in the TECHNICAL NAME OF COPY column, and click OK. This step creates two new key figures at the InfoProvider level.



Technical Name of Source	Ty.	Technical Name of Copy
ZTEMP_RESTR_KF2	SEL	ZTEMP_RESTR_KF2_NEW_2
ZTEMP_RESTR_KF1	SEL	ZTEMP_RESTR_KF1_NEW_1

Figure 3 Rename Component Screen for Transaction RSZC

5. Reopen the query in the ZT_SALES cube using the BEx Query Designer, and you'll see two new restricted key figures created as shown in Figure 4.


BEx Query Designer - Query: [ZTEMP_KM_001] Query Designer demo II
Query
Edit
View
Tools
Help
InfoProvider
Filter
Characteristic Restrictions

```
[ZT_SALES] Demo Cube For Reporting 9Sales)
  Key Figures
    Calculated Key Figure
    Restricted Key Figures
      [ZTEMP_RESTR_KF3] Revenue - &ZTEST_MATL_GROUP&
      [ZTEMP_RESTR_KF2] Revenue - &ZTEST_MATL_GROUP&
      [ZTEMP_RESTR_KF1] Revenue - &ZTEST_MATL_GROUP&
      [ZTEMP_RESTR_KF1_NEW_1] Revenue - &ZTEST_MATL_GROUP&
      [ZTEMP_RESTR_KF2_NEW_2] Revenue - &ZTEST_MATL_GROUP&
    [0AMOUNT] Amount
    [0QUANTITY] Quantity
  Dimensions
    [ZT_SALES1] Material
    [ZT_SALES2] Organization
    [ZT_SALES3] Sales Area
    [ZT_SALESP] Data Package
    [ZT_SALEST] Time
    [ZT_SALESU] Unit

Characteristic Restrictions
  [0SOURSYSTEM] Source system ID
  [0MATERIAL__0MATL_GROUP] Material Group
```

Figure 4 BEx Query Designer Showing the New Calculated Key Figures

These key figures can be modified to fit the new definition of the key figure.

You can use the same steps to copy and create new calculated key figures using an existing key figure as a template by selecting CALCULATED KEY FIGURES as the option in step 2.

Tip 27

Changing a Variable Type en Masse

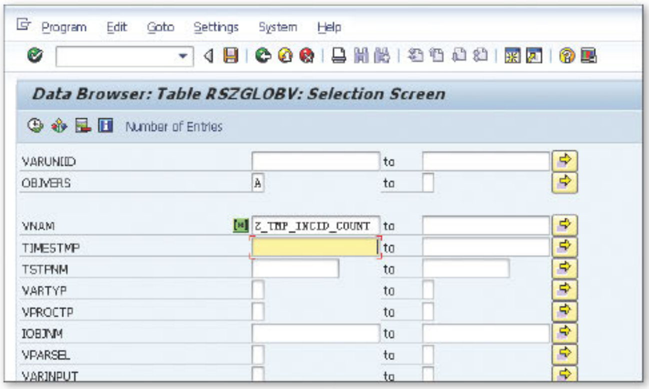
If you've defined a required variable that's used in several reports, you can change the variable en masse without removing it from the query.

Let's say that you've defined a BEx variable that appears as a required entry in several BEx queries. However, now you need to change the variable but don't want to take it out of all the BEx queries. Luckily, SAP provides a database table that allows you to change the variable without the need to remove it, change it, and reinsert it. This tip provides the instructions you need to use the table successfully.

✓ And Here's How ...

SAP provides database Table RSZGLOBV in the ABAP Data Dictionary to make mass changes to variables. To make the changes, follow these steps:

1. Go to Transaction SE11 in SAP NetWeaver BW and look for Table RSZGLOBV by clicking on the DISPLAY button.
2. In the resulting selection screen of the table, enter the technical name of the variable that has to be changed in the VNAM field as shown in Figure 1. Click EXECUTE.
3. In the following screen, select the table entry and click on the CHANGE button at the top of the screen or press F6 to open the next screen as shown in Figure 2.
4. Change the values of the variable properties as appropriate.



Data Browser: Table RSZGLOBV: Selection Screen

VARUNIID		to	
OBJVERS	A	to	
VNAM	Z_TMP_INCID_COUNT	to	
TIMESTMP		to	
TSTPNM		to	
VARTYP		to	
VPROCTP		to	
IOBJNM		to	
VPARSEL		to	
VARINPUT		to	

« *Figure 1 Search for Variable*

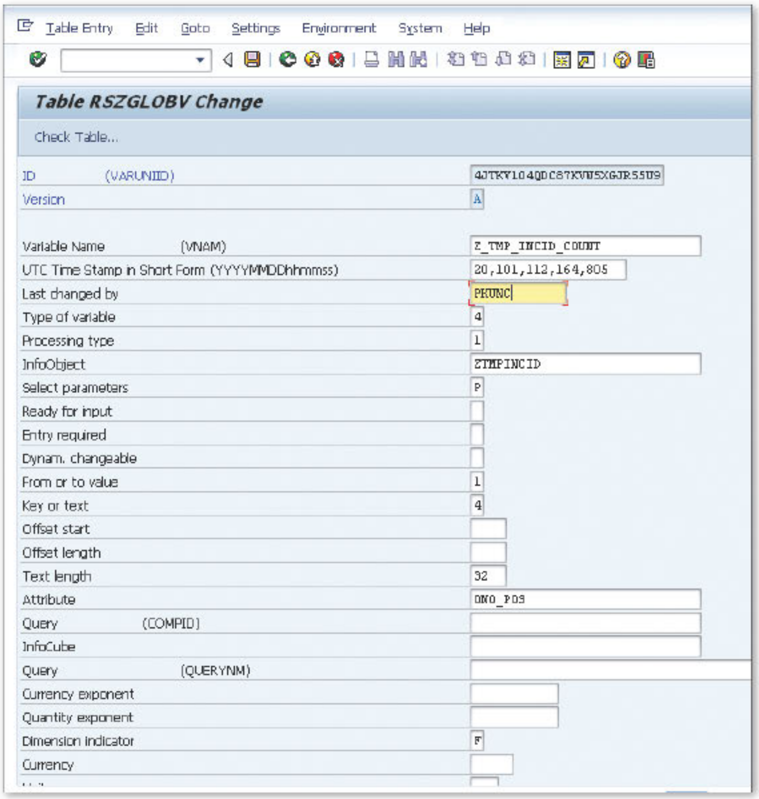


Table RSZGLOBV Change

ID (VARUNIID)	4JTKV104QDC87KVU5XGJR55U9
Version	A
Variable Name (VNAM)	Z_TMP_INCID_COUNT
UTC Time Stamp in Short Form (YYYYMMDDhhmmss)	20,101,112,164,805
Last changed by	PKUNC
Type of variable	4
Processing type	1
InfoObject	ZTMPINCID
Select parameters	P
Ready for input	
Entry required	
Dynam. changeable	
From or to value	1
Key or text	4
Offset start	
Offset length	
Text length	32
Attribute	0NO_POS
Query (COMPID)	
InfoCube	
Query (QUERYNM)	
Currency exponent	
Quantity exponent	
Dimension indicator	F
Currency	

Figure 2 Change Variable Properties

After you save your changes, the changes to the variable will automatically be populated in all of your reports.

Tip 28

Reviewing Key Figure Information without Using the BEx Query Designer

You can provide business end users with the ability to look up a BEx key figure formula and break down a calculated key figure with drilldown capabilities without using the BEx Query Designer.

Typically, BEx key figure formulas represent critical business rules, which can be viewed in the BEx Query Designer. In large SAP NetWeaver BW implementations, it's possible that only the developers or power users will have access to BEx Query Designer. However, it may be important for business end users to review the definition of the key figure and breakdown of a calculated key figure with drilldown capabilities, but they most likely won't have access to the BEx Query Designer. Even if the end users have BEx Query Designer access, they may not have the required training to look up the definition of a key figure.

In this tip, we explain how to look at the formula of the key figure without accessing the Query Designer.

✓ And Here's How ...

For this scenario, we'll use a simple BEx query with one key figure: gross profit. In this case, gross profit is an aggregation of different levels of a hierarchy. Follow these steps:

1. Execute a BEx report as shown in Figure 1, and right-click on the BEx key figure.

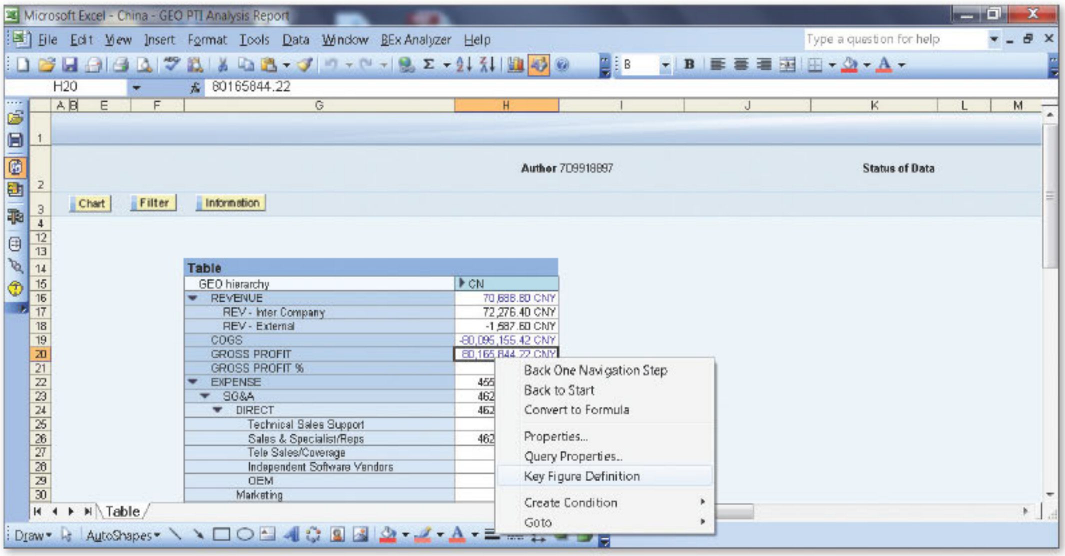


Table	
GEO hierarchy	CN
REVENUE	70,688.80 CNY
REV - Inter Company	72,276.40 CNY
REV - External	-1,587.60 CNY
COGS	-80,095,155.42 CNY
GROSS PROFIT	80,165,844.22 CNY
GROSS PROFIT %	
EXPENSE	455
SG&A	462
DIRECT	462
Technical Sales Support	
Sales & Specialist/Reps	462
Tele Sales/Coverage	
Independent Software Vendors	
OEM	
Marketing	

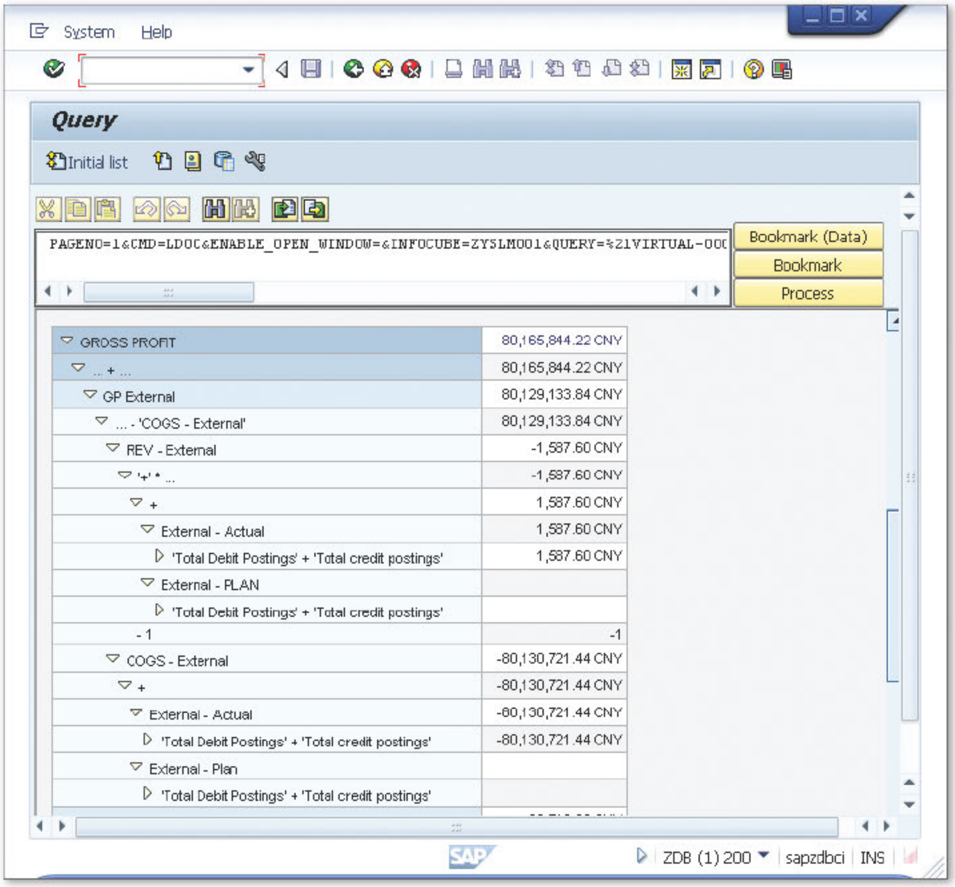
Figure 1 Right-Clicking on a BEx Key Figure

2. From the context-sensitive menu of the key figure, select Key Figure Definition. This launches the SAP NetWeaver Portal and displays the definition of the key figure (see Figure 2).

In this example, the business user is now able to see a detailed breakdown of GROSS PROFIT, including all of the calculations that are used to derive this number.

If the key figure contains a formula, you can see how the formula is constructed in the key figure definition. The formula is displayed in full with all of the operators and objects used in the formula.

If you want to see additional information for this key figure, choose Details from the key figure context menu.



```
PAGENO=1&CMD=LDOC&ENABLE_OPEN_WINDOW=&INFOCUBE=ZYSLM001&QUERY=%21VIRTUAL-00(
```

GROSS PROFIT	80,165,844.22 CNY
... + ...	80,165,844.22 CNY
GP External	80,129,133.84 CNY
... - 'COGS - External'	80,129,133.84 CNY
REV - External	-1,587.60 CNY
'+' * ...	-1,587.60 CNY
+	1,587.60 CNY
External - Actual	1,587.60 CNY
'Total Debit Postings' + 'Total credit postings'	1,587.60 CNY
External - PLAN	
'Total Debit Postings' + 'Total credit postings'	
- 1	-1
COGS - External	-80,130,721.44 CNY
+	-80,130,721.44 CNY
External - Actual	-80,130,721.44 CNY
'Total Debit Postings' + 'Total credit postings'	-80,130,721.44 CNY
External - Plan	
'Total Debit Postings' + 'Total credit postings'	

Figure 2 Key Figure Definition and Detailed Breakdown of GROSS PROFIT

Tip 29

Creating a Custom Hierarchical Display in BEx

You can pool together different key figures using BEx to create your own custom hierarchy structure for a specific reporting purpose.

When creating a BEx report, you can display the data in a hierarchical format by using an existing standard hierarchy either loaded from the SAP ERP system or created on the SAP NetWeaver BW side. However, in some scenarios, you may want the flexibility to report the data in a specific hierarchical format that is not supported by the standard available hierarchies. In this case, you need the ability to rapidly create your own hierarchical structure in the BEx Query Designer.

This tip discusses an intuitive approach you can use to create a custom hierarchy on the fly using the BEx Query Designer.

✓ And Here's How ...

In this example, we'll group together revenue categories and expense categories in a customized hierarchical format. Although standard hierarchies are available to display the revenue and expense categories on the BEx report, this tip shows how these revenue and cost categories can be regrouped on the fly to build a highly customized hierarchical display of these categories.

To create a custom hierarchy in a BEx report, follow these steps:

1. Create a new BEx report on an InfoCube or DSO that has all the revenue and cost data available. In this case, you'll create the report on Special Ledger data, which has all of the revenue and cost line items.

2. Create the key figures for different types of revenues such as REV - Inter Company, REV - External, and so on. These key figures can be created as a formula, calculated key figure, or restricted key figure. In this case, you'll use a characteristic such as G/L Account for identifying the different revenue types.
3. Create a formula type key figure and name it "Revenue".
4. Drag all of the different types of revenue key figures created in the earlier step to the Revenue key figure as shown in Figure 1. These key figures will now show up under the Total Revenue Key figure. Using this method, you can group the revenues in different ways depending on your reporting requirement and then place them under Revenue.

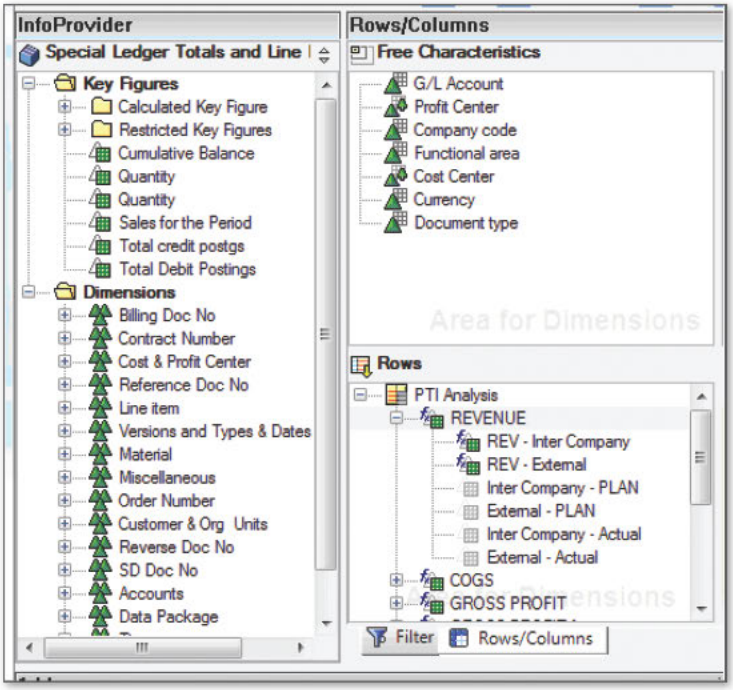
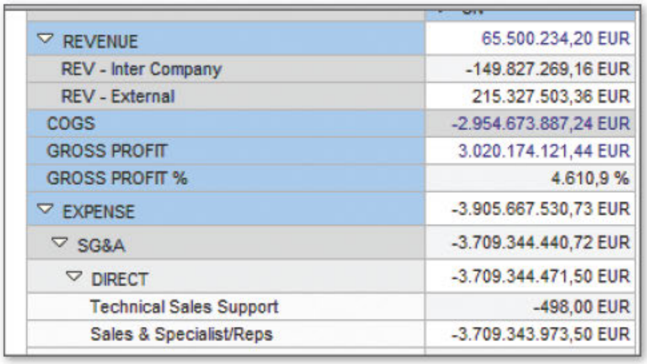


Figure 1 BEx Query Designer Showing Revenue Key Figures under Revenue

5. Create the required grouping for Cost of Goods Sold (COGS) by creating the formula type key figure and naming it "COGS".
6. Create all of the different types of COGS key figures such as COGS - Intercompany, COGS - External, and so on. Again, these key figures can be formulas,

restricted key figures, or calculated key figures. The groupings can be based on the G/L ACCOUNT object or any other characteristic.

7. Drag and drop all the COGS key figures under the COGS formula key figure. This way you have created a structure for displaying all the COGS data.
8. Similarly, you can create groupings for any other expenses or GROSS PROFIT.
9. Save and execute the query. Reports generated from this BEx will display the data in the custom hierarchical format as defined in the earlier steps (see Figure 2). It also enables drilldowns from the total or total expense to the breakdowns of these groupings.



▽ REVENUE	65.500.234,20 EUR
REV - Inter Company	-149.827.269,16 EUR
REV - External	215.327.503,36 EUR
COGS	-2.954.673.887,24 EUR
GROSS PROFIT	3.020.174.121,44 EUR
GROSS PROFIT %	4.610,9 %
▽ EXPENSE	-3.905.667.530,73 EUR
▽ SG&A	-3.709.344.440,72 EUR
▽ DIRECT	-3.709.344.471,50 EUR
Technical Sales Support	-498,00 EUR
Sales & Specialist/Reps	-3.709.343.973,50 EUR

Figure 2 Report Display Showing the Data in a Hierarchical Format

Using the simple steps outlined in the preceding list, you can now create a custom hierarchical format to display the data that is not supported by the standard hierarchies.

Tip 30

Integrating Custom Formulas into the Formula Builder for Reuse

You can integrate custom-created formulas into the Formula Builder for reuse across transformations without any coding.

In SAP NetWeaver BW, you'll sometimes have scenarios where you're creating formulas in transformations instead of doing ABAP coding. In these cases, you can integrate the custom formula into the Formula Builder that can be reused in multiple transformations. The transformation library, in collaboration with the Formula Builder, enables you to easily create formulas without using ABAP coding. This method leverages more standard transformation content as compared to implementing ABAP coding in individual transformations, and also enables reusability of the logic across transformations.

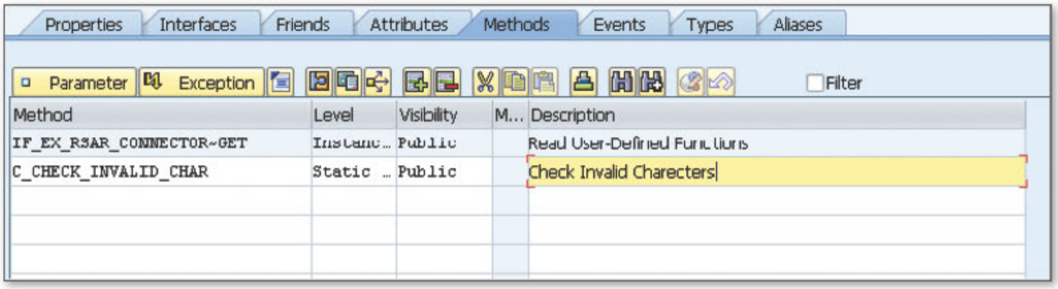
This tip describes how to integrate a custom-created formula into the Formula Builder.

✓ And Here's How ...

Custom formulas are integrated into the transformation library of the Formula Builder by means of BAdI `RSAR_CONNECTOR`. Go to Transaction SE19 and follow these steps:

1. In the Create Implementation section, change the radio button to Classic BAdI, enter "RSAR_CONNECTOR" as the BAdI name, and click on the Create Impl button.

2. On the resulting popup, enter the Implementation Name and click Continue.
3. In the resulting screen, make a note of the name of the implementation class, and activate the implementation by clicking the Activate button at the top of the screen.
4. Go to Transaction SE24. The Class Builder: Initial Screen will appear; enter "ZCL_IM_TEST_FORMULA" in the Object type field and click the Change button.
5. On the resulting screen, go to the Methods tab to implement the custom method. Enter the technical name of the method and declare it as Static and Public as shown in Figure 1. Declaring the method as Static ensures that there will only be one instance of the method during runtime.



Method	Level	Visibility	M...	Description
IF_EX_RSAR_CONNECTOR~GET	Instanc...	Public		Read User-Defined Functions
C_CHECK_INVALID_CHAR	Static ...	Public		Check Invalid Charecters

Figure 1 Defining the Method

6. Click on the Parameters button and define the parameters.
7. In the Methods tab, double-click on the Method name, enter the code, and activate the method.
8. Go to Transaction SE19. In the Edit Implementation section, change the radio button to Classic BAdI, enter Implementation name "ZTEST_FORMULA", and click on the Change button.

Tip 31

Seamlessly Binding all SAP NetWeaver BW Web Application Designer (WAD) Components

You can connect all WAD components to facilitate interactive filter and drill-down, similar to what you'll see with SAP BusinessObjects Dashboards functionality.

Similar to SAP BusinessObjects Dashboards (formerly Xcelsius), the SAP NetWeaver BW Web Application Designer (WAD) uses many objects such as tables, charts, and other widgets to display information. Typically, user actions such as drill, filter, or radio button selections impact one or more of these objects. In WAD, however, you need to configure the object properties to allow this behavior.

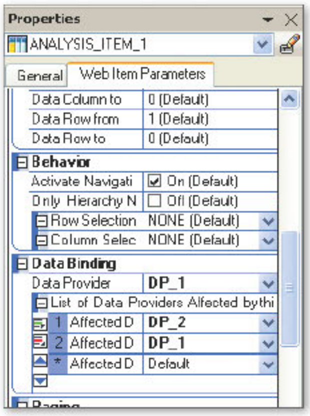
For example, during WAD navigation, we want every drill or filter action to influence all the WAD objects (tables, charts, etc.). If you have table 1 called DP_1 and table 2 called DP_2, with each table originating from the same InfoProvider but from a different query, we must connect/bind them together. With this tip, users can easily bind all WAD objects together to enable drill or filter actions, which in turn update all WAD components.

Note: you need to perform this action on every component in your WAD report, and make sure that the characteristic you want to affect will be defined in each query (since for each item, you can choose a different SAP NetWeaver BW query).

And Here's How ...

To bind WAD objects, follow these steps:

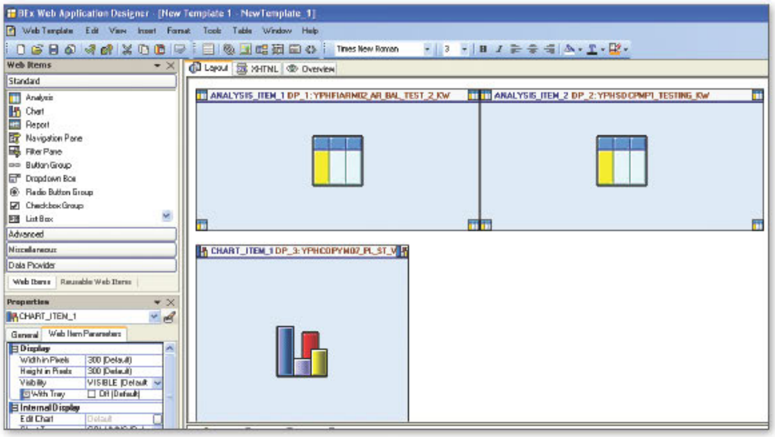
1. Access the SAP NetWeaver BW Web Application Designer, go to Web Item Properties, and then go to Web Item Parameters.
2. In the Data Binding section, specify that Data Provider1 (DP_1) will affect DP_1 and DP_2 objects as shown in Figure 1.



Data Column to	0 (Default)
Data Row from	1 (Default)
Data Row to	0 (Default)
Behavior	
Activate Navigati	☑ On (Default)
Only Hierarchy N	☐ Off (Default)
Row Selection	NONE (Default)
Column Selec	NONE (Default)
Data Binding	
Data Provider	**DP_1**
List of Data Providers Affected by thi	
1 Affected D	**DP_2**
2 Affected D	**DP_1**
* Affected D	Default

« *Figure 1 WAD Web Item Parameters Pane*

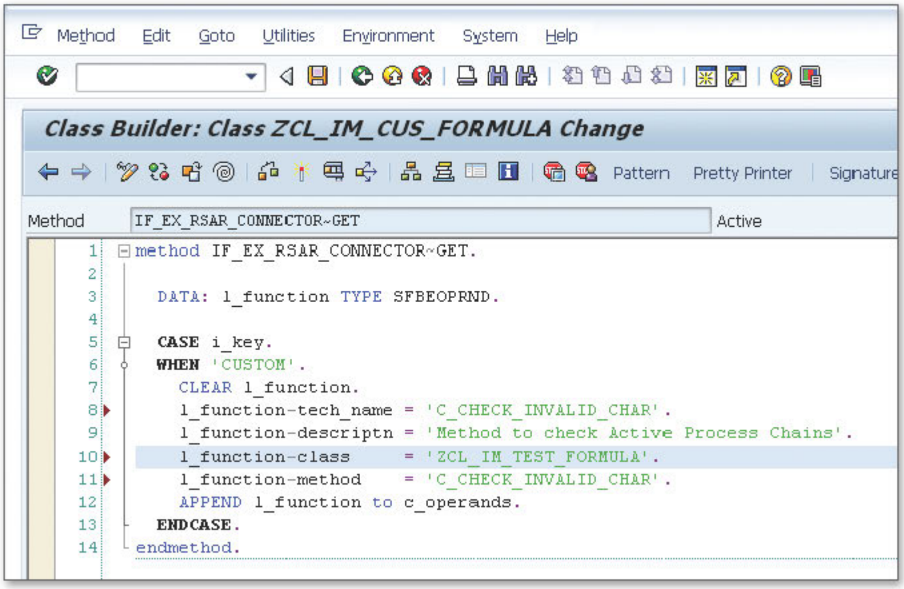
Now when you filter or drill, the action will affect both DP_1 and DP_2 objects. Figure 2 displays the multiple web panes that will be updated with content based on user selection via radio button, drop-down menu, filter menu, or checkbox.



Display	
Width in Pixels	300 (Default)
Height in Pixels	300 (Default)
Visibility	VISIBLE (Default
With Tray	☐ Off (Default)
Internal Display	
Edit Chart	Default

Figure 2 WAD Interactivity of Components: DP_1 and DP_2 Objects

3. Click on the INACTIVATE button at the top of the screen to implement your BAdI (you can't edit active implementations).
4. Double-click on the `GET` method on the INTERFACE tab to incorporate your custom methods as formulas (see Figure 3).

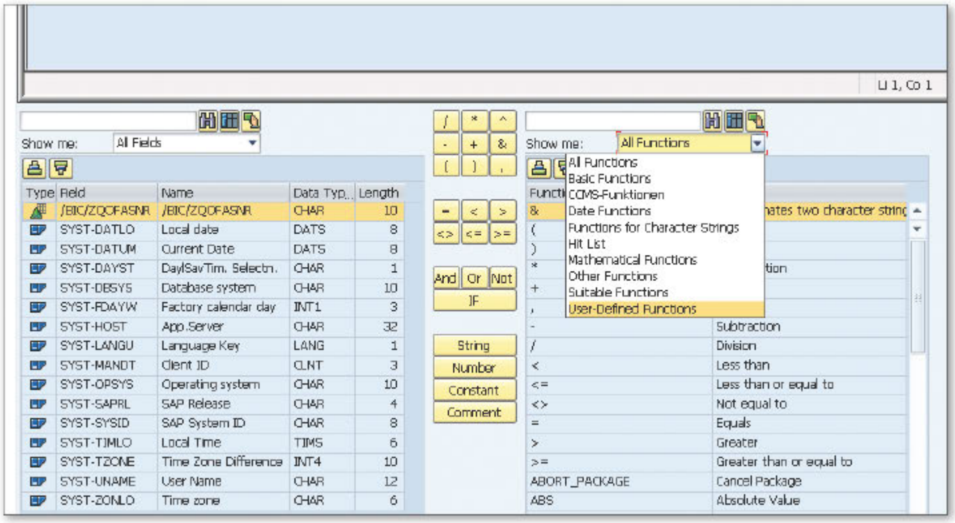


```
method IF_EX_RSAR_CONNECTOR~GET.

  DATA: l_function TYPE SFBEOPRND.

  CASE i_key.
  WHEN 'CUSTOM'.
    CLEAR l_function.
    l_function-tech_name = 'C_CHECK_INVALID_CHAR'.
    l_function-descriptn = 'Method to check Active Process Chains'.
    l_function-class     = 'ZCL_IM_TEST_FORMULA'.
    l_function-method    = 'C_CHECK_INVALID_CHAR'.
    APPEND l_function to c_operands.
  ENDCASE.
endmethod.
```

Figure 3 Code for the Class

5. Each new formula needs the following four fields populated:
   - TECH_NAME
     Technical name that will uniquely identify the formula in the list of formulas.
   - DESCRIPTION
     Description displayed in the Formula Builder.
   - CLASS
     ABAP class in which the formula method is implemented.
   - METHOD
     Method within the ABAP class mentioned earlier in which the formula is implemented.
6. Activate the `Get` method by clicking on the ACTIVATE button at the top of the screen, and then go back one screen and activate the implementation. You can now use the formula shown in Figure 4 in any of the transformations.

You can see this method under User-Defined Functions when creating a formula in a transformation as shown in Figure 4.



Type	Field	Name	Data Typ..	Length
	/BIC/ZQOFASNR	/BIC/ZQOFASNR	CHAR	10
	SYST-DATLO	Local date	DATS	8
	SYST-DATUM	Current Date	DATS	8
	SYST-DAYST	DaylSavTim. Selectn.	CHAR	1
	SYST-DBSYS	Database system	CHAR	10
	SYST-FDAYW	Factory calendar day	INT1	3
	SYST-HOST	App.Server	CHAR	32
	SYST-LANGU	Language Key	LANG	1
	SYST-MANDT	Client ID	CLNT	3
	SYST-OPSYS	Operating system	CHAR	10
	SYST-SAPRL	SAP Release	CHAR	4
	SYST-SYSID	SAP System ID	CHAR	8
	SYST-TIMLO	Local Time	TIMS	6
	SYST-TZONE	Time Zone Difference	INT4	10
	SYST-UNAME	User Name	CHAR	12
	SYST-ZONLO	Time zone	CHAR	6

Functi	
&	nates two character strinç
(	
)	
*	tion
+	
,	
-	Subtraction
/	Division
<	Less than
<=	Less than or equal to
<>	Not equal to
=	Equals
>	Greater
>=	Greater than or equal to
ABORT_PACKAGE	Cancel Package
ABS	Absolute Value

Figure 4 Creating a Formula in the Transformations

Tip 32

Implementing the NOT EXISTS Functionality in BEx Queries

You can use the "not exists" functionality in a BEx query to show entries that don't yet have any data but are needed for reporting purposes.

Some reporting scenarios require you to display master data that do not have any transactions associated with them. For instance, say you need a report that shows sales by region, and you are required to show sales for all regions and also regions that didn't have any sales for the period selected. One way to model this scenario is to combine the relevant InfoObject with the transaction data using a MultiProvider; however, this involves backend model changes that aren't easy to implement.

This tip describes a simple setting that can be used to model scenarios that require the "not exists" functionality. This method doesn't require any modeling changes on the MultiProvider, so it's very simple to implement.

✓ And Here's How ...

We'll use a simple BEx query that displays sales by region to illustrate this tip. It's assumed that the query is already created and available to execute. Follow these steps:

1. Execute the BEx query, and choose the characteristics you want to show the NO VALUES records for. In this tip, we're choosing the REGION characteristic. Figure 1 shows only two regions because the data exists only for those two regions.

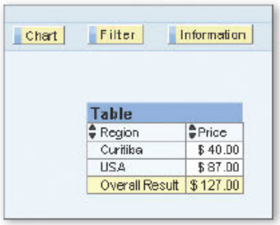
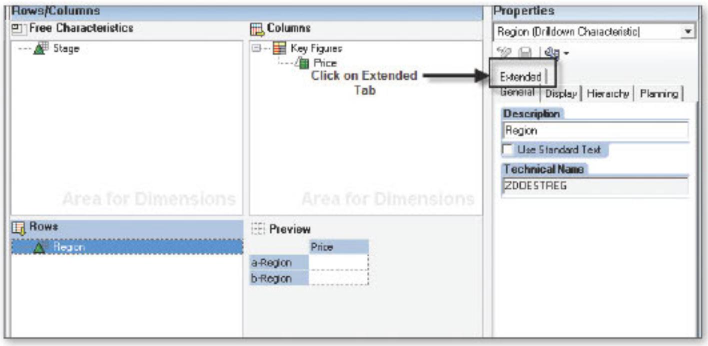


Table	
Region	Price
Curitiba	$ 40.00
USA	$ 87.00
Overall Result	$ 127.00

« *Figure 1 BEx Query Output Showing Regions with Sales*

2. Go into the BEx Analyzer Toolbox, and choose the Edit Query button.
3. Choose the characteristic Region where you want all of the master data values shown. In the Properties pane, choose the Extended option (see Figure 2).



Description: Region

Technical Name: ZDDESTREG

Figure 2 Properties Pane for the Characteristic REGION

4. In the Extended tab of the Properties section, choose the Master Data radio button in the Access Type for Result Values.
5. Save the query and run it; now you'll see that the report shows all the data values for the master data even if the transactional data does "not exist." So in the example, Region has more values than before even without the corresponding transactional records in the MultiProvider (see Figure 3).

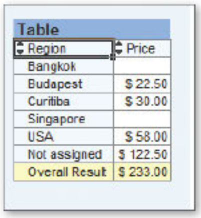


Table	
Region	Price
Bangkok	
Budapest	$ 22.50
Curitiba	$ 30.00
Singapore	
USA	$ 58.00
Not assigned	$ 122.50
Overall Result	$ 233.00

« *Figure 3 BEx Query Output Showing Regions with and without Sales Data*

This method can be used to implement the "not exists" functionality for any master data scenarios using this simple setting on the BEx query.

Tip 33

Dynamically Determining Multiple Time Periods for Reporting

You can bypass standard BEx limitations and create dynamic time ranges as selection criteria for BEx reporting.

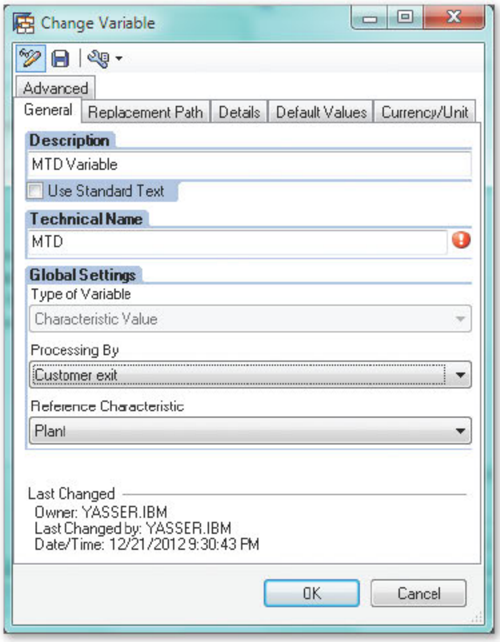
Dynamic time measures are critical in most financial and operational reporting and analysis. However, standard BEx functionality is limited with respect to this functionality. But by using the system date as a point of reference, dynamic time calculations are possible.

This tip describes the steps to create dynamic time ranges as a selection and use them to filter criteria in BEx queries.

✓ And Here's How ...

You can set up dynamic time ranges by using the user-exit variables and a few lines of ABAP code in the SAP NetWeaver BW system. Follow these steps:

1. Using BEx Query Designer, open the BEx query in change mode.
2. Define a variable using the variable editor in the BEx Query Designer, and set the PROCESSING BY dropdown to CUSTOMER EXIT.
3. In the REFERENCE CHARACTERISTIC field, select the InfoObject on which the variable is to be based.
4. Set other options as required as shown in Figure 1. This variable can be assigned as a filter in the BEx query.


Change Variable

Advanced | General | Replacement Path | Details | Default Values | Currency/Unit

Description: MTD Variable

Use Standard Text

Technical Name: MTD

Global Settings

Type of Variable: Characteristic Value

Processing By: Customer exit

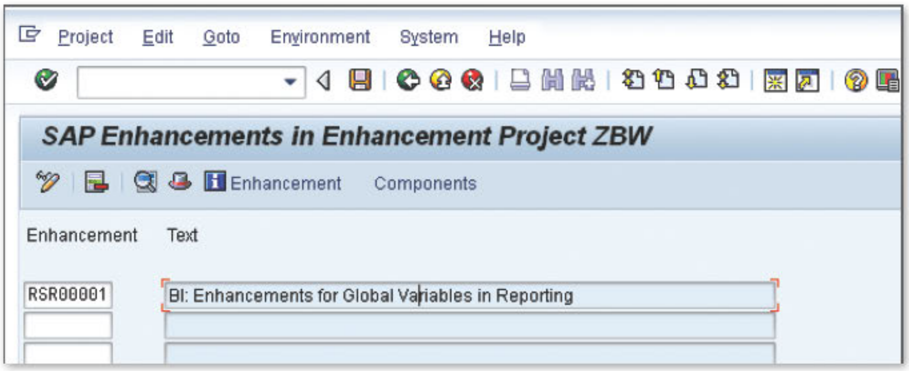
Reference Characteristic: Plant

Last Changed
Owner: YASSER.IBM
Last Changed by: YASSER.IBM
Date/Time: 12/21/2012 9:30:43 PM

OK | Cancel

« Figure 1 Variable Creation Screen in a BEx Query

5. Log in to the SAP NetWeaver BW system. Execute Transaction CMOD, and create an enhancement project. Name the enhancement project "ZBW".
6. Assign enhancement RSR00001 to the project as shown in Figure 2. This enhancement will include exit EXIT_SAPLRRS0_001. Click on the COMPONENTS option available on the header to access the function exit EXIT_SAPLRRS0_001.


Project Edit Goto Environment System Help

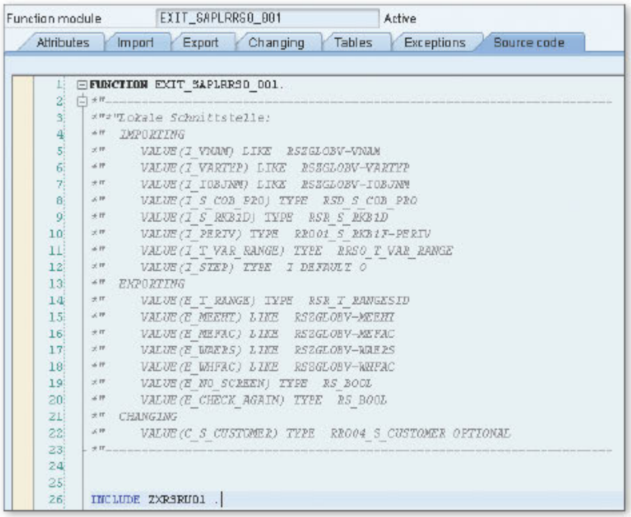
SAP Enhancements in Enhancement Project ZBW

Enhancement Components

Enhancement	Text
RSR00001	BI: Enhancements for Global Variables in Reporting

« Figure 2 Defining the Enhancement Project

7. In the next screen, double-click on EXIT_SAPLRRS0_001 and you will see the include ZXRSRU01 as shown in Figure 3.



```
EXIT_SAPLRRS0_001

FUNCTION EXIT_SAPLRRS0_001.
*"----------------------------------------------------------------------
*"*"Lokale Schnittstelle:
*"  IMPORTING
*"     VALUE(I_VNAM) LIKE  RSZGLOBV-VNAM
*"     VALUE(I_VARTYP) LIKE  RSZGLOBV-VARTYP
*"     VALUE(I_IOBJNM) LIKE  RSZGLOBV-IOBJNM
*"     VALUE(I_S_COB_PRO) TYPE  RSD_S_COB_PRO
*"     VALUE(I_S_RKB1D) TYPE  RSR_S_RKB1D
*"     VALUE(I_PERIV) TYPE  RRO01_S_RKB1F-PERIV
*"     VALUE(I_T_VAR_RANGE) TYPE  RRS0_T_VAR_RANGE
*"     VALUE(I_STEP) TYPE  I DEFAULT 0
*"  EXPORTING
*"     VALUE(E_T_RANGE) TYPE  RSR_T_RANGESID
*"     VALUE(E_MEEHT) LIKE  RSZGLOBV-MEEHT
*"     VALUE(E_MEFAC) LIKE  RSZGLOBV-MEFAC
*"     VALUE(E_WAERS) LIKE  RSZGLOBV-WAERS
*"     VALUE(E_WHFAC) LIKE  RSZGLOBV-WHFAC
*"     VALUE(E_NO_SCREEN) TYPE  RS_BOOL
*"     VALUE(E_CHECK_AGAIN) TYPE  RS_BOOL
*"  CHANGING
*"     VALUE(C_S_CUSTOMER) TYPE  RRO04_S_CUSTOMER OPTIONAL
*"----------------------------------------------------------------------


INCLUDE ZXRSRU01 .
```

Figure 3 Customer Exit Code

8. Double click on include ZXRSRU01, thereby creating a new include, and enter the following code to determine the dynamic current month to date range.

```
IF i_step = 1.

  CASE i_vnam.

    WHEN 'MTD'.
* date range from start of current month to current date
      l_s_range-sign = 'I'.
      l_s_range-opt = 'BT'.
      CONCATENATE sy-datum+0(6) '01'  INTO l_s_range-low.
      l_s_range-high = sy-datum+0

      APPEND l_s_range TO e_t_range.
   ENDCASE
ENDIF.
```

9. Execute Transaction RSRTU to use the variable in a BEx query and test it in the SAP NetWeaver BW system. This variable will now enable dynamic time periods.

Tip 34

Enabling Cascading Variable Prompt Values in a BEx Query

You can make a user's navigation of BEx query variables easier and quicker by using a BAdI implementation to enable cascading variable prompt values.

When you execute a query in the BEx Query Designer through SAP NetWeaver BW, prompts are defined as variables for entering selection prompts when you execute the query. This method, however, doesn't support cascading values for the variable prompt selection, which means that both independent and dependent variables are listed individually. Therefore, a user will see a long list of unwanted values, which can be confusing and waste a lot of time while selecting a right value from the list.

To bypass this issue, you can use a BAdI to implement cascading filters in BEx variables. This method restricts the list of values for a query variable prompt based on a previously entered variable value.

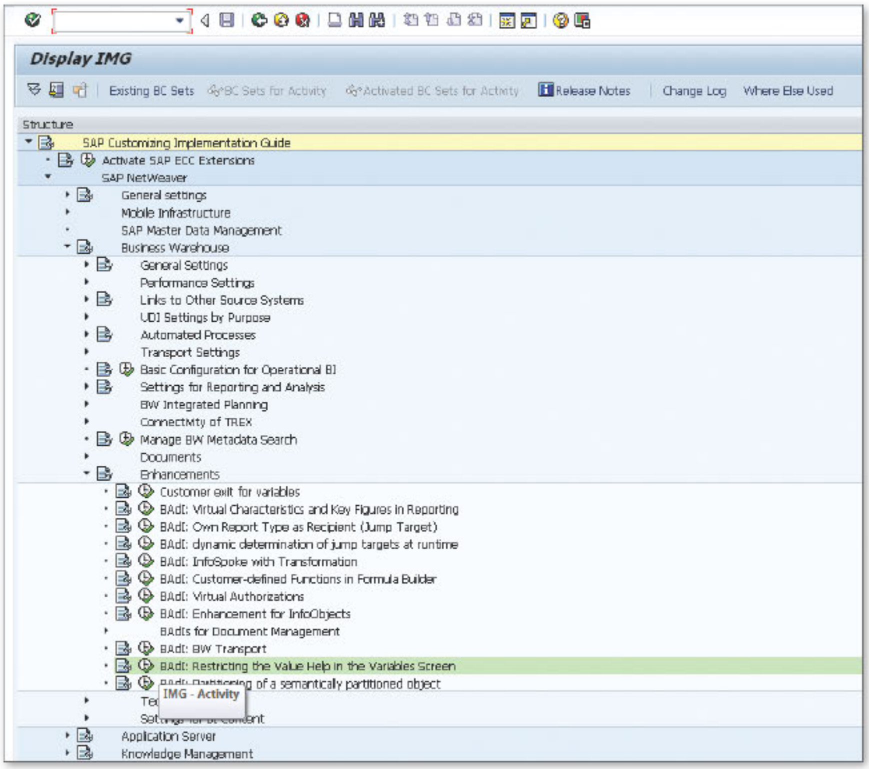
✓ And Here's How ...

To help illustrate the tip, we'll use a simple scenario of a BEx query with two variable prompts, PROFIT CENTER and COST CENTER. From a modeling perspective, it's assumed that the profit center is enabled as a navigational attribute of COST CENTER, which means that cost centers are created based on existing profit centers. In this scenario, when the user selects the value from the first variable prompt (that is, PROFIT CENTER), the list of values for the second variable (that is, COST CENTER), should be restricted for the PROFIT CENTER that is selected previously.

We'll use BAdI RSR_VARIABLE_F4_RESTRICT_BADI that is available with support package SAPKW70109 to restrict the list of values of input-ready variables for Cost Center in the BEx variables screen.

This tip assumes that the reader has some basic knowledge of BAdI implementation and also can create the required BEx query to complete this implementation. Follow these steps:

1. Log in to the SAP NetWeaver BW system and execute Transaction SPRO.
2. Click on SAP Reference IMG. Search for the BAdI: Restricting the Value Help in the Variables Screen item in the Structure, and click Continue.
3. In the resulting screen shown in Figure 1, click the Execute button near BAdI: Restricting the Value Help in the Variables Screen.



Display IMG

```
Structure
SAP Customizing Implementation Guide
  Activate SAP ECC Extensions
  SAP NetWeaver
    General settings
    Mobile Infrastructure
    SAP Master Data Management
    Business Warehouse
      General Settings
      Performance Settings
      Links to Other Source Systems
      UDI Settings by Purpose
      Automated Processes
      Transport Settings
      Basic Configuration for Operational BI
      Settings for Reporting and Analysis
      BW Integrated Planning
      Connectivity of TREX
      Manage BW Metadata Search
      Documents
      Enhancements
        Customer exit for variables
        BAdI: Virtual Characteristics and Key Figures in Reporting
        BAdI: Own Report Type as Recipient (Jump Target)
        BAdI: dynamic determination of jump targets at runtime
        BAdI: InfoSpoke with Transformation
        BAdI: Customer-defined Functions in Formula Builder
        BAdI: Virtual Authorizations
        BAdI: Enhancement for InfoObjects
        BAdIs for Document Management
        BAdI: BW Transport
        BAdI: Restricting the Value Help in the Variables Screen
        BAdI: Partitioning of a semantically partitioned object
      Te[illegible]
      Set[illegible]ent
    Application Server
    Knowledge Management
```

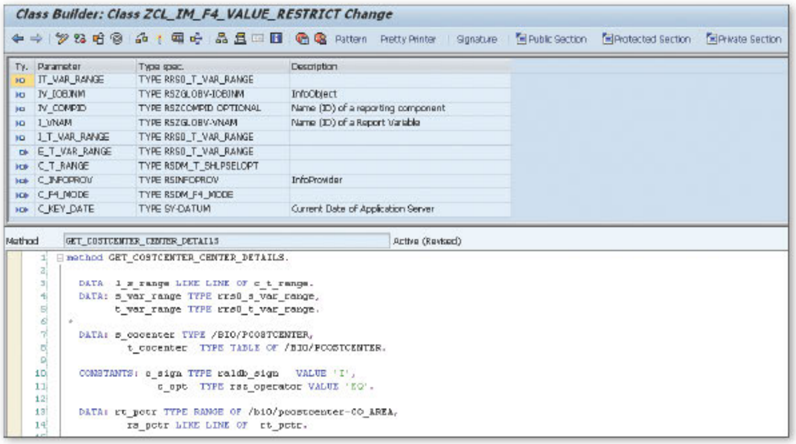
Figure 1 SAP Reference IMG Screen with the BAdI Selected

4. In the resulting screen, enter the name "ZX_IM_F4_HELP" for the enhancement implementation. Also add a description for the enhancement. Click Continue.

5. In the next screen, associate the enhancement implementation to the BAdI that you're planning to use. In the BADI IMPLEMENTATION column, enter "RSR_VARIABLE_F4_RESTRICT_BADI". Click CONTINUE.
6. When the CREATE IMPLEMENTATION CLASS popup appears, click on the EMPTY CLASS button.
7. Double-click on FILTER VAL on the left side of the screen. Click on the COMBINATION button to enter the InfoObject names on which you need cascading. In the CHANGE FILTER VALUE popup, enter the InfoObject name "0COSTCENTER" in the VALUE 1 field on which you need cascading, and click CONTINUE.
8. Save and activate the implementation.

Now you need to enhance the implementation class that you just created.

1. Go to Transaction SE24, enter the CLASS name "ZCL_IM_F4_VALUE_RESTRICT", and click on CHANGE.
2. Create a new method for the class by entering the name "GET_COSTCENTER_DETAILS". This method will have the logic for deriving the cost center based on profit center values selected by the user.
3. Enter the relevant parameters for the method, and then save and activate.
4. Double-click on the method GET_COSTCENTER. The next screen allows you to enter code for the method (see Figure 2). You can also use some of the parameters for the method to restrict this logic to specific InfoProviders, queries, or variables. This ensures that the logic applies only where required and not in other cases.



Parameter	Type spec.	Description
IT_VAR_RANGE	TYPE RRS0_T_VAR_RANGE	
IV_IOBJNM	TYPE RSZGLOBV-IOBJNM	InfoObject
IV_COMPID	TYPE RSZCOMPID OPTIONAL	Name (ID) of a reporting component
I_VNAM	TYPE RSZGLOBV-VNAM	Name (ID) of a Report Variable
I_T_VAR_RANGE	TYPE RRS0_T_VAR_RANGE	
E_T_VAR_RANGE	TYPE RRS0_T_VAR_RANGE	
C_T_RANGE	TYPE RSDM_T_SHLPSELOPT	
C_INFOPROV	TYPE RSINFOPROV	InfoProvider
C_F4_MODE	TYPE RSDM_F4_MODE	
C_KEY_DATE	TYPE SY-DATUM	Current Date of Application Server

Method GET_COSTCENTER_CENTER_DETAILS Active (Revised)

```
method GET_COSTCENTER_CENTER_DETAILS.

  DATA  l_s_range LIKE LINE OF c_t_range.
  DATA: s_var_range TYPE rrs0_s_var_range,
        t_var_range TYPE rrs0_t_var_range.
*
  DATA: s_cocenter TYPE /BIO/PCOSTCENTER,
          t_cocenter  TYPE TABLE OF /BIO/PCOSTCENTER.

  CONSTANTS: c_sign TYPE raldb_sign   VALUE 'I',
                 c_opt  TYPE rss_operator VALUE 'EQ'.

  DATA: rt_pctr TYPE RANGE OF /bi0/pcostcenter-CO_AREA,
          rs_pctr LIKE LINE OF  rt_pctr.
```

Figure 2 Entering Code for the Method to Derive the Cost Center Values

The input parameters of this method will have the values of the profit center entered by the user. You need to code the logic in the method to look up the relevant cost centers for the profit center selected by the user, as shown in the following sample code:

```
method GET_COSTCENTER_CENTER_DETAILS.
DATA  l_s_range LIKE LINE OF c_t_range.
  DATA: s_var_range TYPE rrs0_s_var_range,
        t_var_range TYPE rrs0_t_var_range.
DATA: s_cocenter TYPE /BI0/PCOSTCENTER,
        t_cocenter  TYPE TABLE OF /BI0/PCOSTCENTER.
CONSTANTS: c_sign TYPE raldb_sign   VALUE 'I',
           c_opt  TYPE rsz_operator VALUE 'EQ'.
DATA: rt_pctr TYPE RANGE OF /bi0/pcostcenter-CO_AREA,
        rs_pctr LIKE LINE OF  rt_pctr.
  CHECK i_t_var_range IS NOT INITIAL.
   t_var_range[] = i_t_var_range[].
   SELECT co_area FROM /BI0/PCOSTCENTER into rs_pctr.
     APPEND rs_pctr TO rt_pctr.
SELECT * FROM /bi0/pcostcenter INTO TABLE t_cocenter
          WHERE CO_AREA IN rt_pctr.
  IF sy-subrc IS NOT INITIAL.   " No Records?
    s_cocenter-costcenter = ''.
    APPEND s_cocenter TO t_cocenter.
  ENDIF.
  CLEAR l_s_range.
  LOOP AT t_cocenter INTO s_cocenter.
    l_s_range-iobjnm = iv_iobjnm.
    l_s_range-sign   = c_sign.
    l_s_range-option = c_opt.
    l_s_range-low    =  s_cocenter-costcenter.
    APPEND l_s_range TO c_t_range.
  ENDLOOP.
ENDselect.
endmethod.
```

With the preceding implementation, you should be able to use profit center and cost center in a BEx query and implement cascading filter functionality for the cost center.

Tip 35

Enabling Dynamic Calculations in BEx by Using the Master Data Attribute Value in the Formula Variable

You can use the master data attribute value in the formula variable to derive calculated key figures in BEx.

Certain business scenarios require dynamic calculations in a BEx query based on the attribute value of the master data. Usually, you can do this by including the attribute value in the InfoProvider and using that value for the calculation. However, in scenarios where the attribute value changes very often, using the value from the InfoProvider won't work because the value of the attribute will be stored in the InfoProvider and won't reflect any changes unless the InfoProvider is reloaded with data. Instead, you can use the attribute from the master data because master data is typically loaded more frequently and should always have the most recent value.

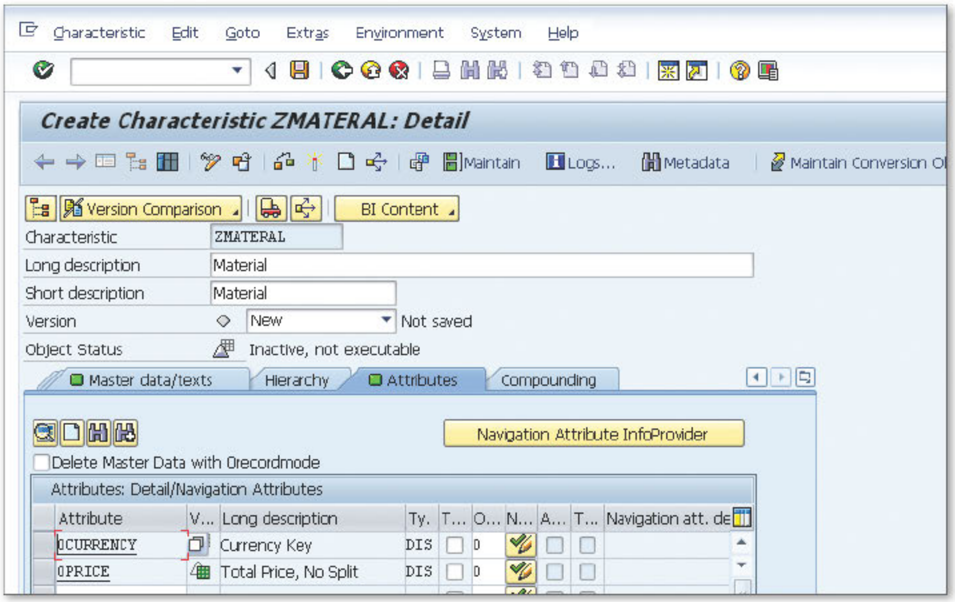
This tip describes how to create a formula variable using a master data attribute value as the replacement path and using the formula variable for calculating key figures in BEx.

✓ And Here's How ...

In this tip, we'll explain how to dynamically calculate the total cost of a material using the value of price in the material master. In this case, the total cost of material will not be calculated in the transformations. The material price will not be part of the InfoProvider but will instead be used in the formula variable in BEx from the attribute value of the material.

Follow these steps to set up the scenario:

1. Log in to SAP NetWeaver BW and use Transaction RSD1 to create a new master data object named "ZMATERIAL". Go to the ATTRIBUTES tab and add 0PRICE as an attribute (see Figure 1).
2. Save and activate the InfoObject. The attribute 0PRICE will hold the value for the material price.



Create Characteristic ZMATERAL: Detail

Characteristic	ZMATERAL
Long description	Material
Short description	Material
Version	New Not saved
Object Status	Inactive, not executable

Delete Master Data with 0recordmode

Attributes: Detail/Navigation Attributes

Attribute	V...	Long description	Ty.	T...	O...	N...	A...	T...	Navigation att. de
0CURRENCY		Currency Key	DIS	☐	0	✓	☐	☐	
0PRICE		Total Price, No Split	DIS	☐	0	✓	☐	☐	

Figure 1 Creating ZMATERIAL and Adding 0PRICE as an Attribute

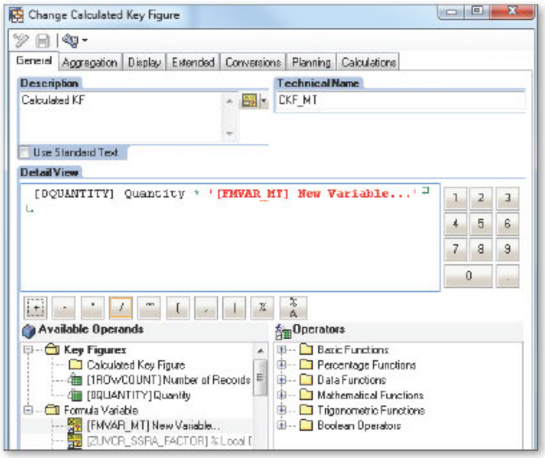
3. Create a test DSO ZTEST_MT, and include ZMATERIAL and 0QUANTITY in the DSO. Load data for the DSO and ZMATERIAL using the regular ETL process.
4. Using the BEx Query Designer, create a simple BEx query on the DSO using sales document number & ZMATERIAL as characteristics and 0QUANITY as the key figure.

Your next step is to create a calculated key figure based on the master data attribute. Follow these steps:

1. Open the BEx query in change mode, and create a new calculated key figure named CKF_MT. In the screen for Calculated Key Figure, right-click on the Formula Variable option under Available Operands, and select the New Variable option.
2. In the next screen, name the variable "FMVAR_MT". In the Processing By dropdown box, select Replacement Path. As soon as you select Replacement Path as the processing type, the screen will show another field for entering the reference characteristic. Select ZMATERIAL as the reference material.

You have now created a calculated key figure CKF_MT, which is based on a formula variable derived from an attribute value 0PRICE of ZMATERIAL master data object. This key figure will now calculate the material cost based on the value of 0PRICE in the ZMATERIAL InfoObject. Follow these steps:

1. Go to the Replacement Path tab on the Change Variable screen. In the Replace With field, select the Attribute Value option. Also in the field attribute, select 0PRICE from the dropdown box. Click on OK, and you will be back in the Change Calculated Key Figure screen.
2. In the Change Calculate Key figure screen, drag and drop both 0Quantity from the Key Figures section and the newly created formula variable FMVAR_MT to the Detail View box. From the Operators section, use the multiplication operator to calculate the material cost as shown in Figure 2.



Change Calculated Key Figure

Description: Calculated KF

Technical Name: CKF_MT

Use Standard Text

Detail View:

```
[0QUANTITY] Quantity * '[FMVAR_MT] New Variable...'
```

Available Operands
- Key Figures
  - Calculated Key Figure
  - [1ROWCOUNT] Number of Records
  - [0QUANTITY] Quantity
- Formula Variable
  - [FMVAR_MT] New Variable...
  - [ZUVCR_SSRA_FACTOR] % Local [

Operators
- Basic Functions
- Percentage Functions
- Data Functions
- Mathematical Functions
- Trigonometric Functions
- Boolean Operators

Figure 2 Defining the Calculated Key Figure Using the Formula Variable

Tip 36

Setting Up the Report-Report Interface for Additional Drill-Through in BEx Queries

You can set up the Report-Report Interface to allow users to drill through from one query to another and discover detailed information in an intuitive manner.

Typically, BEx queries can contain free characteristics, in addition to the main characteristics and key figures used for reporting and analysis. However, users cannot drill-through to understand more granular details regarding a transaction. For example, in a BEx query, a user may view net invoice amount by sold-to-party. With the Report-Report Interface, users will gain new ability by drilling-through to additional billing details such as sold-to-party name. In addition to providing more detailed information, the Report-Report Interface provides a more customized and seamless user experience by eliminating the need to build additional queries or to modify queries during analysis.

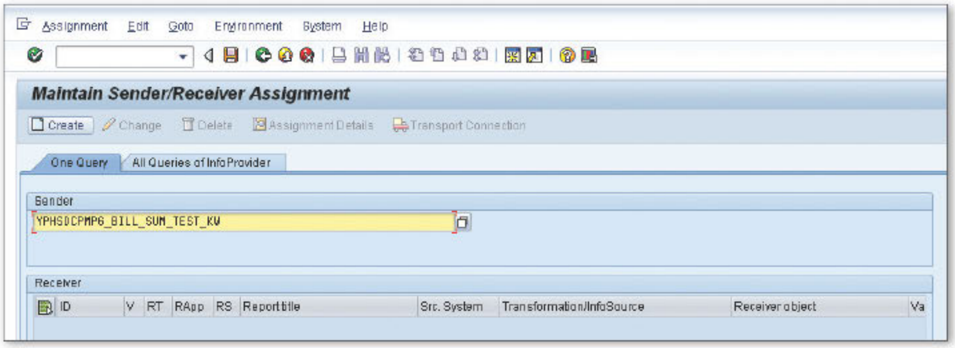
In this tip, we'll show you how to set up the Report-Report Interface to seamlessly enable drill-through capability to more detailed information.

✓ And Here's How ...

Note that Report-Report Interface or query jump can be mentioned to describe the same functionality.

Enter Transaction RSBBS and follow these steps to set up the interface for the scenario described above:

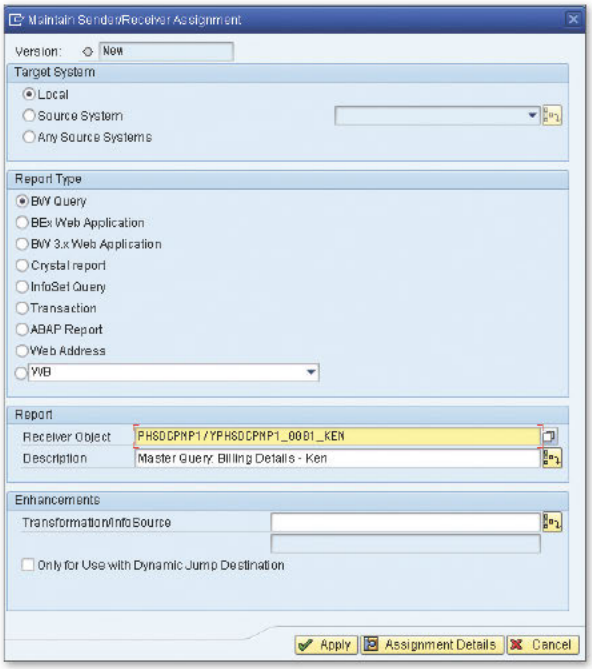
1. In the Maintain Sender/Receiver Assignment screen (Figure 1), enter the BEx query where the report link will be set up in the Sender field. Click Create.


Assignment Edit Goto Environment System Help
Maintain Sender/Receiver Assignment
Create Change Delete Assignment Details Transport Connection
One Query All Queries of InfoProvider
Receiver
ID V RT RApp RS Report title Src. System Transformation/InfoSource Receiver object Va

Sender: YPHSDCPMP6_BILL_SUM_TEST_KV

Figure 1 Enter BEx Query in the Sender Field

2. The screen shown in Figure 2 will be displayed. Select Local for the target system and BW Query for the report type. Then select the target or receiver report that will be displayed via the drill-through action to display more granular information. You can fill in the Description field to display the drill-through option for the user. Finally, click on Apply.

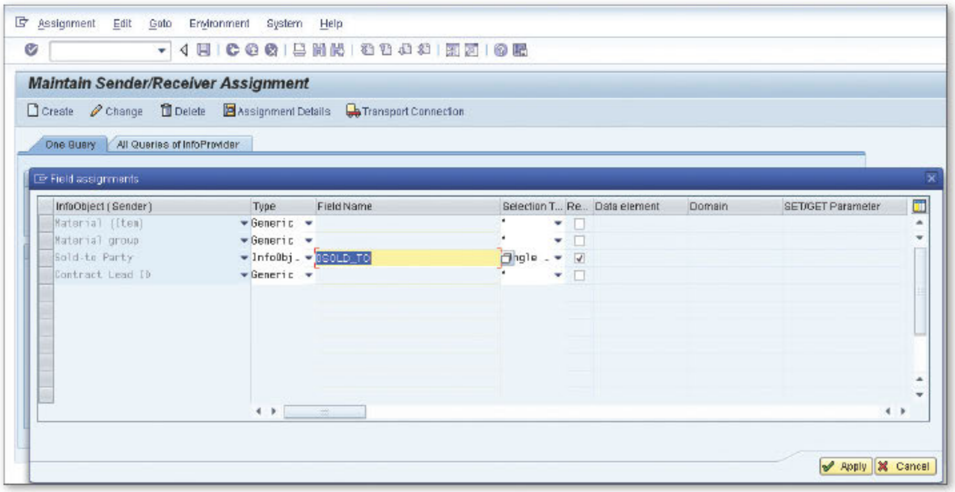


Version: New
Target System: Local
Report Type: BW Query
Receiver Object: PHSDCPMP1/YPHSDCPMP1_0001_KEN
Description: Master Query: Billing Details - Ken

Figure 2 Enter Fields for Maintain Sender/Receiver Assignment

3. Next, click on the Assignment Details button to set the fields to be assigned from sender to receiver.
4. Next, as shown in Figure 3, set the field assignments for sold-to party including Type (InfoObject), Field Name (0SOLD_TO), and Selection Type (Single Value).

The Report-Report Interface has now been configured.



InfoObject (Sender)	Type	Field Name	Selection T...	Re...	Data element	Domain	SET/GET Parameter
Material (Item)	Generic		*	☐			
Material group	Generic		*	☐			
Sold-to Party	InfoObj...	0SOLD_TO	...ngle ...	✓			
Contract Lead ID	Generic		*	☐			

Figure 3 Set Field Assignments

To test the Report-Report Interface or query jump, open the sender BEx query. Select Sold-to Party and use the context menu option GoTo • Master Query: Billing Details – Ken. The receiver BEx query will be displayed with granular billing details.

Tip 37

Using Virtual Characteristics and Key Figures in SAP NetWeaver BW for Easy Ad Hoc Analysis

You can easily leverage virtual characteristics and key figures to add dynamic analysis capabilities to your SAP NetWeaver BW BEx reports.

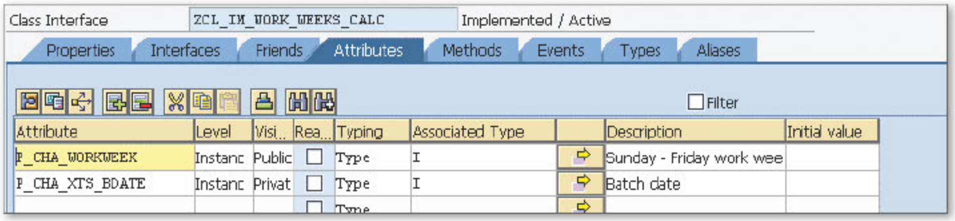
In contrast to traditional data load processing, virtual characteristics and key figures are calculated on the fly or during query runtime. Virtual characteristics and key figures provide the benefit of additional ad hoc analysis without having to rerun and reprocess the entire query, including the data.

For example, if you need to calculate a custom work week (i.e., Sunday through Saturday), you can use virtual characteristics. To accomplish this, you will add a characteristic to your cube (i.e., ZZZZZZ) and it will not be filled in with any transformations. The new virtual characteristic will be calculated by field YYYYYY, which is also in the cube, and populated with data. In this tip, we'll show you how to create and configure the SAP NetWeaver BW BAdI program to enable virtual characteristics in your BEx query. After reading the tip, you'll be able to create on-line manipulations on the data without having to model them first in SAP NetWeaver BW.

✓ And Here's How ...

In this tip, we'll use the example mentioned previously of calculating a custom work week. To use virtual characteristics, follow these steps:

1. Create a new BAdI based on RSR_OLAP_BADI through Transaction SE19. Enter the implementation name and implementation description in the IMPLEMENTATION SHORT TEXT field on the following screen.
2. On the next screen, go to the ATTRIBUTES tab and define the relevant InfoCubes as shown in Figure 1. In this example we're using a Z customer InfoCube, but you can use any InfoCube that has 0CALDAY in it.
3. Now switch to the INTERFACE tab and enter the implementation class. Add the required attributes.

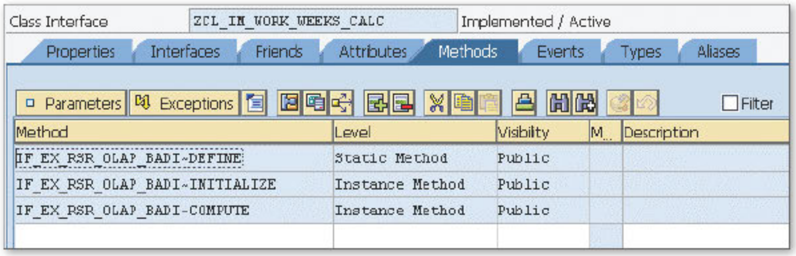


Class Interface ZCL_IM_WORK_WEEKS_CALC Implemented / Active

Attribute	Level	Visi..	Rea...	Typing	Associated Type		Description	Initial value
P_CHA_WORKWEEK	Instanc	Public	☐	Type	I	⇨	Sunday - Friday work wee	
P_CHA_XTS_BDATE	Instanc	Privat	☐	Type	I	⇨	Batch date	
			☐	Type		⇨		

Figure 1 Add the Required Attribute

4. You now need to add code to methods to select your new work week as shown in Figure 2. Access the already created table/DSO DB_Table and assign it to the ZZZZZZ field. We've included an example here where we assume that you have an SAP NetWeaver BW table named DB_TABLE, that has the mapping between fields YYYYYY and ZZZZZZ and uses select single to find the match.



Class Interface ZCL_IM_WORK_WEEKS_CALC Implemented / Active

Method	Level	Visibility	M..	Description
IF_EX_RSR_OLAP_BADI~DEFINE	Static Method	Public		
IF_EX_RSR_OLAP_BADI~INITIALIZE	Instance Method	Public		
IF_EX_RSR_OLAP_BADI~COMPUTE	Instance Method	Public		

Figure 2 Add Code to Methods

Note that virtual characteristics should start with p_cha and virtual key figures should start with p_kyf.

```
method IF_EX_RSR_OLAP_BADI~COMPUTE.

  FIELD-SYMBOLS <fs_XTS_BDATE> TYPE ANY.
  FIELD-SYMBOLS <fs_WORKWEEK> TYPE ANY.
```

```
 DATA: l_WORKWEEK TYPE I.

  ASSIGN COMPONENT P_CHA_YYYYYY OF STRUCTURE c_s_data TO <fs_
YYYYYY>.
  ASSIGN COMPONENT P_CHA_ZZZZZZ OF STRUCTURE c_s_data TO <fs_
ZZZZZZ>.

select SINGLE ZZZZZZ FROM DB_TABLE
  INTO <fs_ZZZZZZ>
  where CALDAY = <fs_YYYYYY>.

endmethod.
```

5. Finally, add the virtual characteristics/key figures to the query and it will be populated during run time through the BAdI; you don't have to add the fields that the calculation is based upon.

Tip 38

Using the Enhanced Menu Option in BEx Reports to Obtain More Information about Key Figures

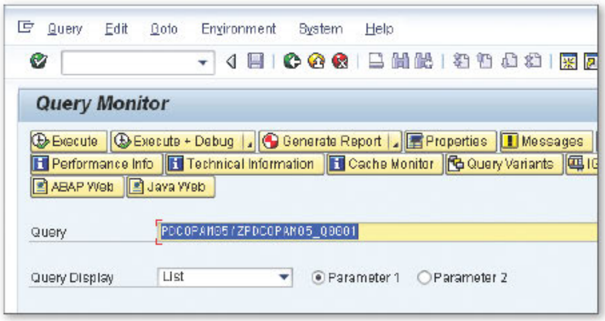
You can use the enhanced menu option to obtain additional key figure metadata though the SAP NetWeaver BW interface.

In BEx, there are many instances where a user may want to know additional details about a key figure. Details regarding the definition of a key figure can be easily found via the ENHANCED MENU option. Without this option, users can't quickly understand the metadata associated with the key figure. In contrast, users may have to generate additional metadata reports via BEx to obtain the same information. This is an unnecessary step when the answer is available with just a few button clicks in the SAP NetWeaver BW interface; we'll show you how in the following tip.

✓ And Here's How ...

Log in to the SAP NetWeaver BW environment and follow these steps:

1. Go to QUERY MONITOR, enter the technical name of the BEx query in the QUERY field, and then click on the EXECUTE button (Figure 1).

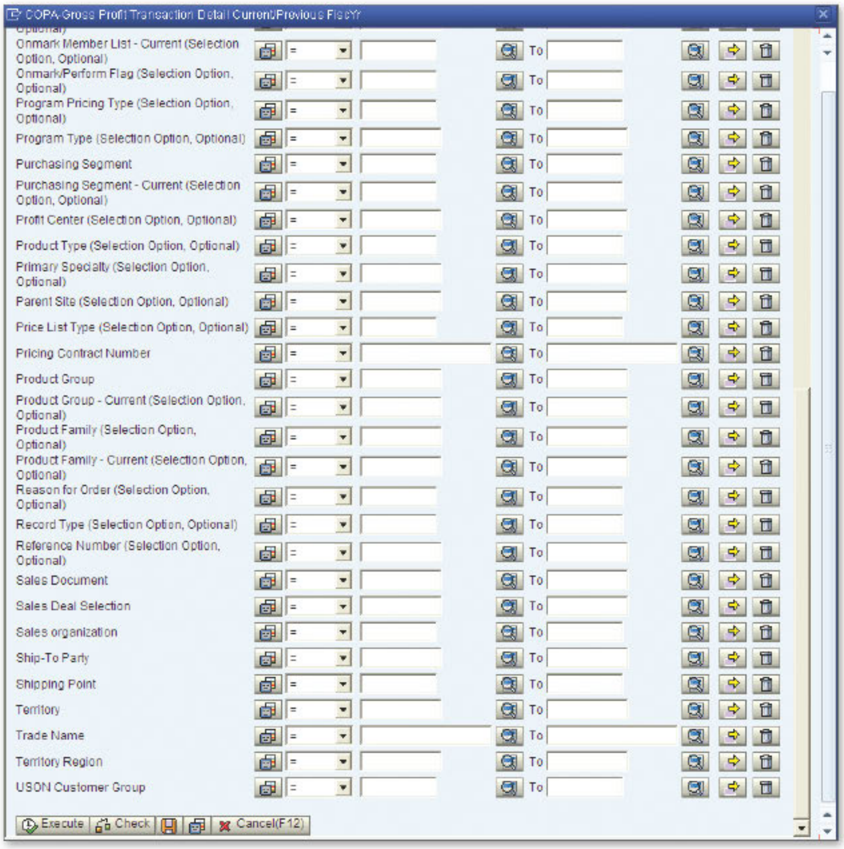


Query PDCOPAM05/ZPDCOPAM05_Q0001

Query Display List [✓ Parameter 1] Parameter 2

Figure 1 Enter the Technical Name of the BEx Query in the Query Monitor

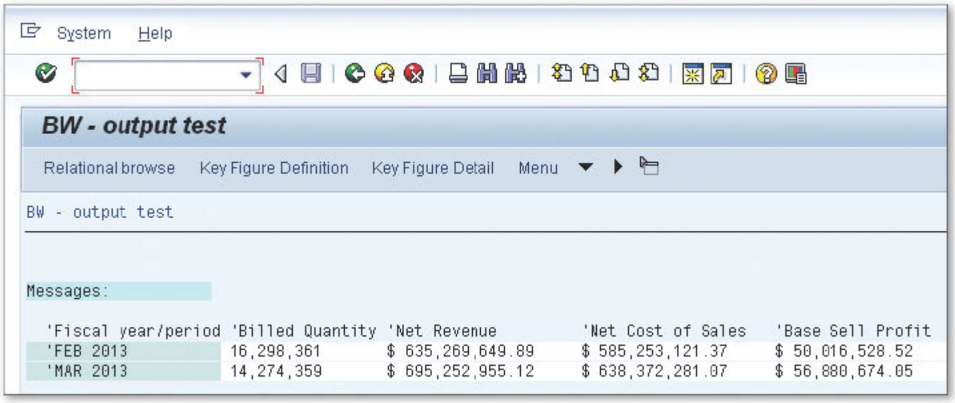
2. Enter any parameters for which you'd like to find out more information (see Figure 2). In this example, we've chosen the Fiscal Year Selection Period.


COPA-Gross Profit Transaction Detail Current/Previous FiscYr
Execute
Check
Cancel(F12)

Onmark Member List - Current (Selection Option, Optional)	=		To	
Onmark/Perform Flag (Selection Option, Optional)	=		To	
Program Pricing Type (Selection Option, Optional)	=		To	
Program Type (Selection Option, Optional)	=		To	
Purchasing Segment	=		To	
Purchasing Segment - Current (Selection Option, Optional)	=		To	
Profit Center (Selection Option, Optional)	=		To	
Product Type (Selection Option, Optional)	=		To	
Primary Specialty (Selection Option, Optional)	=		To	
Parent Site (Selection Option, Optional)	=		To	
Price List Type (Selection Option, Optional)	=		To	
Pricing Contract Number	=		To	
Product Group	=		To	
Product Group - Current (Selection Option, Optional)	=		To	
Product Family (Selection Option, Optional)	=		To	
Product Family - Current (Selection Option, Optional)	=		To	
Reason for Order (Selection Option, Optional)	=		To	
Record Type (Selection Option, Optional)	=		To	
Reference Number (Selection Option, Optional)	=		To	
Sales Document	=		To	
Sales Deal Selection	=		To	
Sales organization	=		To	
Ship-To Party	=		To	
Shipping Point	=		To	
Territory	=		To	
Trade Name	=		To	
Territory Region	=		To	
USON Customer Group	=		To	

Figure 2 Entering Parameters for the BEx Query

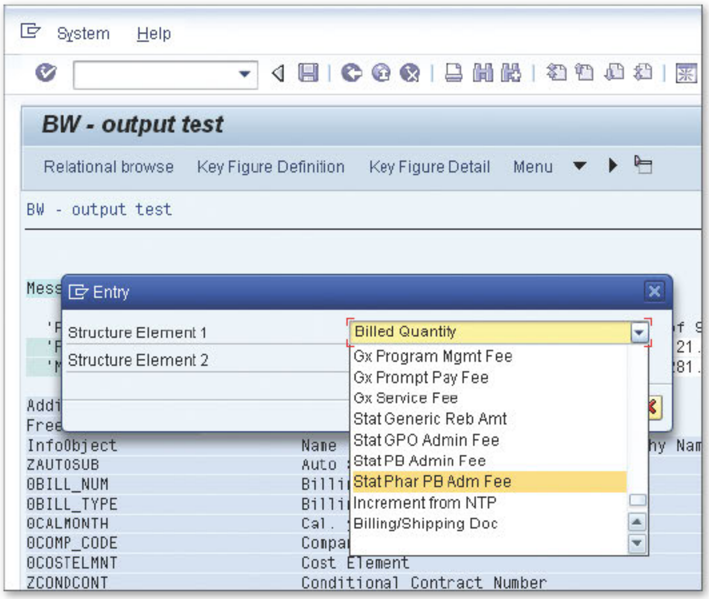
3. The results of the BEx query are returned in the SAP NetWeaver BW interface as shown in Figure 3. Click on the KEY FIGURE DEFINITION button in the report header.



```
BW - output test

Messages:

 'Fiscal year/period 'Billed Quantity 'Net Revenue         'Net Cost of Sales    'Base Sell Profit
 'FEB 2013           16,298,361        $ 635,269,649.89     $ 585,253,121.37      $ 50,016,528.52
 'MAR 2013           14,274,359        $ 695,252,955.12     $ 638,372,281.07      $ 56,880,674.05
```

Figure 3 Results of the BEx Query in SAP NetWeaver BW

4. Select the desired key figure to view its detailed metadata (Figure 4).



```
BW - output test

InfoObject                    Name
ZAUTOSUB                      Auto
0BILL_NUM                     Billi
0BILL_TYPE                    Billi
0CALMONTH                     Cal.
0COMP_CODE                    Compa
0COSTELMNT                    Cost Element
ZCONDCONT                     Conditional Contract Number
```

Figure 4 Select Desired Key Figure to View Its Detailed Metadata

Tip 39

Customizing a BEx Workbook with Excel VBA for Greater Control

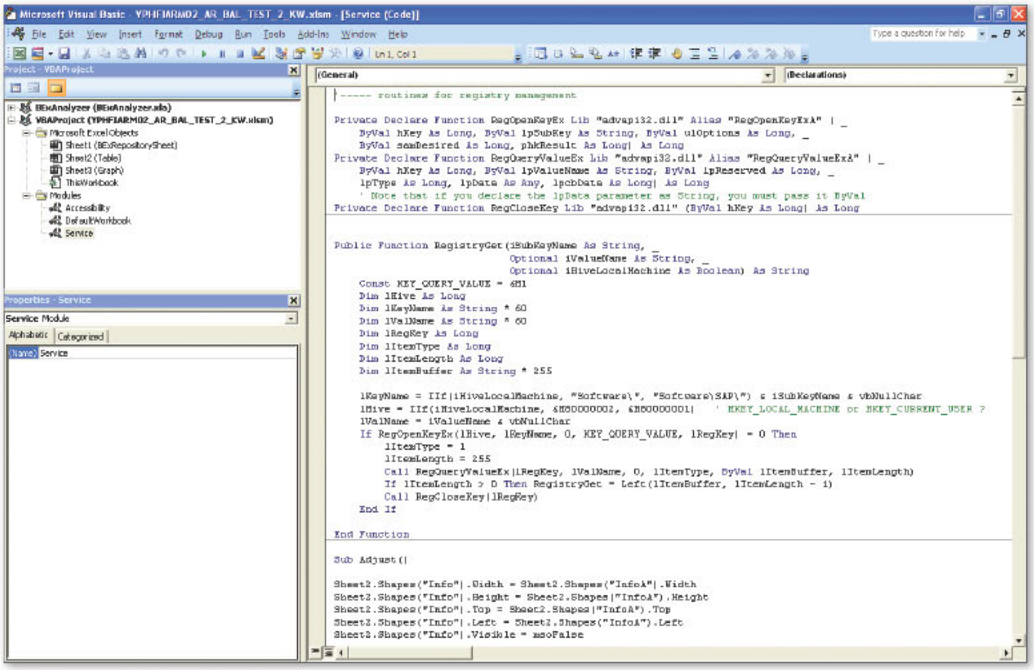
You can use Excel Visual Basic for Applications to control query processing and additional processing functionality in a BEx workbook.

Although BEx gives you the ability to create vertical, horizontal, and cross tab tables, there are times when you need to perform additional processing to accommodate formatting, merging, or cleaning up data. BEx provides the flexibility to embed additional processing in the workbook through the Excel VBA (Visual Basic for Applications) using coding. The SAP-delivered On Refresh capability will be executed after each query pass; so, if there are two queries, this macro would run twice, and if there are three queries, it would run three times. When you are creating your code, you have to take that into consideration. Using Excel VBA enables more precise control/additional processing in BEx Analyzer.

✓ And Here's How ...

To customize a BEx workbook with Excel VBA to control precise query processing requirements, follow these steps:

1. Open up an existing BEx query in Analyzer, and press `Alt` + `F11` to go into the VBA editor as shown in Figure 1.



```
'----- routines for registry management

Private Declare Function RegOpenKeyEx Lib "advapi32.dll" Alias "RegOpenKeyExA" (_
    ByVal hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, _
    ByVal samDesired As Long, phkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32.dll" Alias "RegQueryValueExA" (_
    ByVal hKey As Long, ByVal lpValueName As String, ByVal lpReserved As Long, _
    lpType As Long, lpData As Any, lpcbData As Long) As Long
    ' Note that if you declare the lpData parameter as String, you must pass it ByVal
Private Declare Function RegCloseKey Lib "advapi32.dll" (ByVal hKey As Long) As Long


Public Function RegistryGet(iSubKeyName As String, _
                            Optional iValueName As String, _
                            Optional iHiveLocalMachine As Boolean) As String
    Const KEY_QUERY_VALUE = &H1
    Dim lHive As Long
    Dim lKeyName As String * 60
    Dim lValName As String * 60
    Dim lRegKey As Long
    Dim lItemType As Long
    Dim lItemLength As Long
    Dim lItemBuffer As String * 255

    lKeyName = IIf(iHiveLocalMachine, "Software\", "Software\SAP\") & iSubKeyName & vbNullChar
    lHive = IIf(iHiveLocalMachine, &H80000002, &H80000001)   ' HKEY_LOCAL_MACHINE or HKEY_CURRENT_USER ?
    lValName = iValueName & vbNullChar
    If RegOpenKeyEx(lHive, lKeyName, 0, KEY_QUERY_VALUE, lRegKey) = 0 Then
        lItemType = 1
        lItemLength = 255
        Call RegQueryValueEx(lRegKey, lValName, 0, lItemType, ByVal lItemBuffer, lItemLength)
        If lItemLength > 0 Then RegistryGet = Left(lItemBuffer, lItemLength - 1)
        Call RegCloseKey(lRegKey)
    End If

End Function

Sub Adjust()

Sheet2.Shapes("Info").Width = Sheet2.Shapes("InfoA").Width
Sheet2.Shapes("Info").Height = Sheet2.Shapes("InfoA").Height
Sheet2.Shapes("Info").Top = Sheet2.Shapes("InfoA").Top
Sheet2.Shapes("Info").Left = Sheet2.Shapes("InfoA").Left
Sheet2.Shapes("Info").Visible = msoFalse
```

Figure 1 Opening the VBA Editor

2. On the left side of the VBA Editor, you'll see the PROJECT browser and the SAP NetWeaver BW workbook you are in under VBAPROJECT.
3. At this point, you can either insert a module or write your VB code directly in the table sheet. To insert a module, right-click MODULES, and choose INSERT • MODULE.

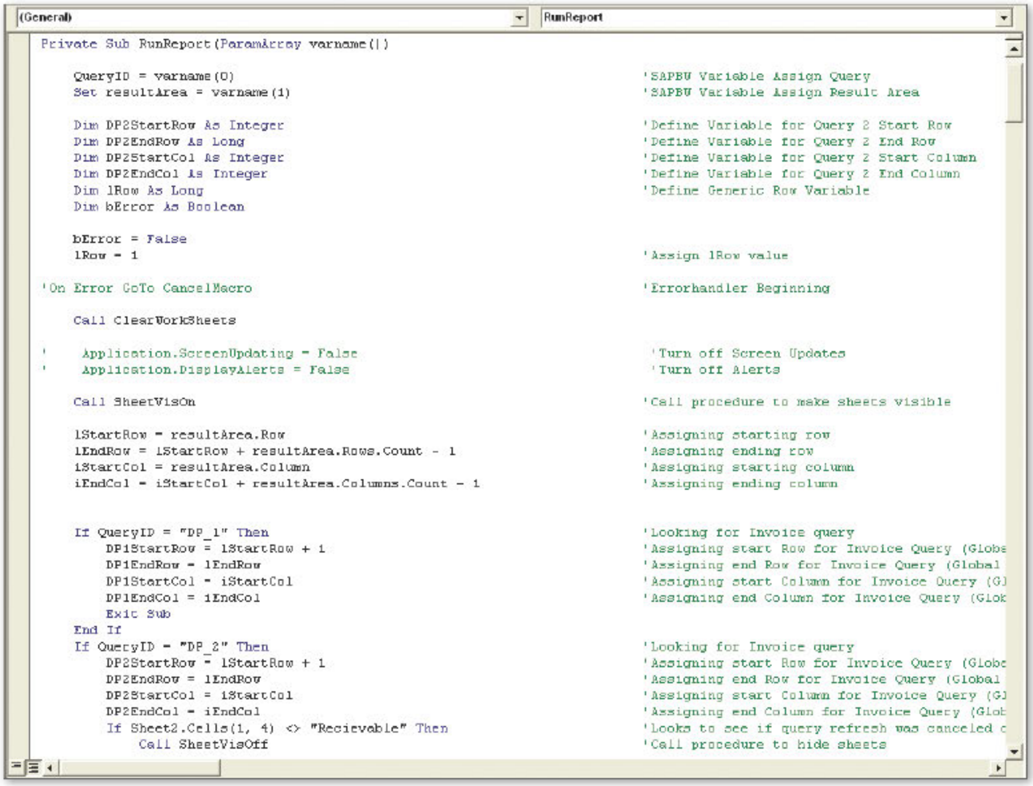
Depending on the version of SAP NetWeaver BW, there are two methods to coding this function:

- 3.x version
  - `Public Sub SAPBEXonRefresh(queryID As String, resultArea As Range)`
- 7.x version
  - `Private Sub RunReport(ParamArray varname())`
  - `QueryID = varname(0) 'SAPBW Variable Assign Query Set resultArea = varname(1) 'SAPBW Variable Assign Result Area`
    - `QueryID` is the data provider name (3.x, SAPBEXq0002/7.x, DP1).
    - `ResultArea` is the row and column range of the table (Dataset).

4. An example of the VB module code block is as follows (Figure 2):

```
lStartRow = resultArea.Row
'Assigning starting row
    lEndRow = lStartRow + resultArea.Rows.Count - 1
'Assigning ending row
    iStartCol = resultArea.Column
'Assigning starting column
     iEndCol = iStartCol + resultArea.Columns.Count - 1
'Assigning ending column
```

This code shows how to assign the row and column number of the first and last row and column of the data table returned from the BEx Query Refresh.



```
Private Sub RunReport(ParamArray varname())

    QueryID = varname(0)                                                    'SAPBW Variable Assign Query
    Set resultArea = varname(1)                                             'SAPBW Variable Assign Result Area

    Dim DP2StartRow As Integer                                              'Define Variable for Query 2 Start Row
    Dim DP2EndRow As Long                                                   'Define Variable for Query 2 End Row
    Dim DP2StartCol As Integer                                              'Define Variable for Query 2 Start Column
    Dim DP2EndCol As Integer                                                'Define Variable for Query 2 End Column
    Dim lRow As Long                                                        'Define Generic Row Variable
    Dim bError As Boolean

    bError = False
    lRow = 1                                                                'Assign lRow value

'On Error GoTo CancelMacro                                                  'Errorhandler Beginning

    Call ClearWorkSheets

'    Application.ScreenUpdating = False                                      'Turn off Screen Updates
'    Application.DisplayAlerts = False                                       'Turn off Alerts

    Call SheetVisOn                                                         'Call procedure to make sheets visible

    lStartRow = resultArea.Row                                              'Assigning starting row
    lEndRow = lStartRow + resultArea.Rows.Count - 1                         'Assigning ending row
    iStartCol = resultArea.Column                                           'Assigning starting column
    iEndCol = iStartCol + resultArea.Columns.Count - 1                      'Assigning ending column


    If QueryID = "DP_1" Then                                                'Looking for Invoice query
        DP1StartRow = lStartRow + 1                                         'Assigning start Row for Invoice Query (Globe
        DP1EndRow = lEndRow                                                 'Assigning end Row for Invoice Query (Global
        DP1StartCol = iStartCol                                             'Assigning start Column for Invoice Query (Gl
        DP1EndCol = iEndCol                                                 'Assigning end Column for Invoice Query (Glob
        Exit Sub
    End If
    If QueryID = "DP_2" Then                                                'Looking for Invoice query
        DP2StartRow = lStartRow + 1                                         'Assigning start Row for Invoice Query (Globe
        DP2EndRow = lEndRow                                                 'Assigning end Row for Invoice Query (Global
        DP2StartCol = iStartCol                                             'Assigning start Column for Invoice Query (Gl
        DP2EndCol = iEndCol                                                 'Assigning end Column for Invoice Query (Glob
        If Sheet2.Cells(1, 4) <> "Recievable" Then                          'Looks to see if query refresh was canceled c
            Call SheetVisOff                                                'Call procedure to hide sheets
```

Figure 2 VB Module Code Example

Tip 40

Creating a Condition to Filter a Key Figure in BEx

You can create a condition to get around the common limitation of filtering on a key figure.

Creating filters on a BEx query is a pretty straightforward process, and SAP NetWeaver BW offers plenty of options you can use to create your variables against a characteristic. However, there is no way to create variables on a key figure. There are times when you don't need to see all of the key figure values from your selection criteria. For instance, you might be running an AR query and only want to see all the credits your organization gave to a customer over $20,000. You can run a query to pull all credits based on document type but you can't just see the credit where the amounts meet your criteria without a filter on the measure. To get around this issue, you have to create a condition. After the condition is created on the key figure, the condition is applied after the query is processed and the data is returned to the client; hence, it's not a runtime filter.

This tip guides you through the step-by-step process for creating a filter on a key figure in BEx.

✓ And Here's How ...

In this example, we'll run an accounts receivables query that pulls all documents for a specified customer, and we'll create a condition to only show the debit/credit amounts that are over $1,000.

To create a new condition, log into the BEx environment and follow these steps:

1. Click on the conditions tab on the tool bar of a BEx query. Now right-click and select NEW CONDITION.
2. The NEW CONDITION tab opens as shown in Figure 1. Enter the DESCRIPTION for the condition "Greater than 1000", and then click on the EDIT button.

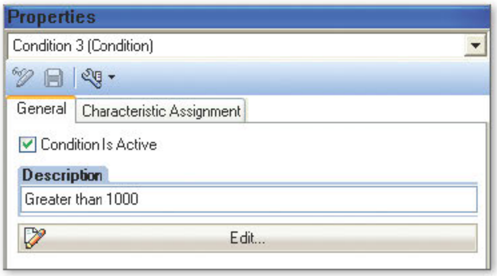


Figure 1 New Condition Tab

3. When the CHANGE CONDITION tab opens, click on the NEW button.
4. Define the condition parameters as shown in Figure 2:
   - Select KEY FIGURE under the STRUCTURE column.
   - Select GREATER THAN under the OPERATOR column.
   - Select 1,000 under the VALUES column.
5. After you've finalized the DEFINE CONDITION PARAMETERS tab, click on the now activated TRANSFER button to complete the condition creation process (Figure 2).

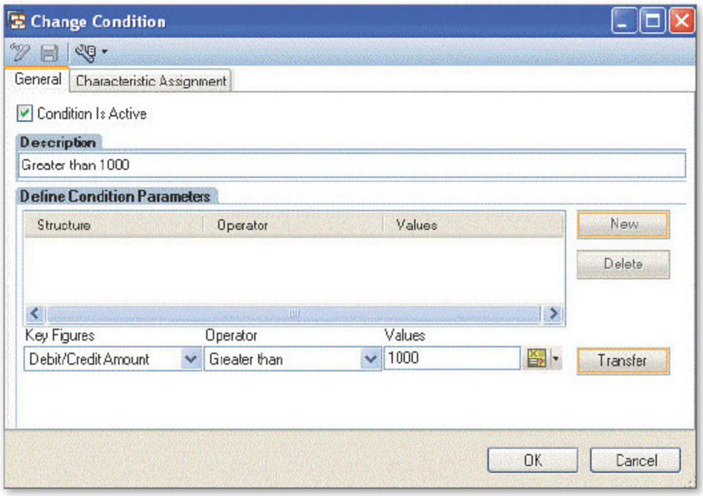
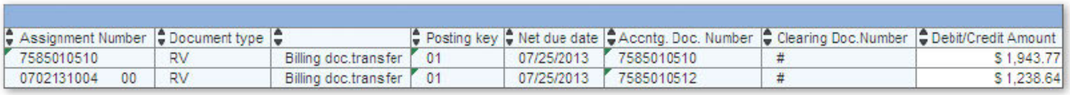


Figure 2 Finalize the Create Condition by Clicking on the Transfer Button

6. After the condition has been created, run a BEx query with the key figure referenced in the condition (i.e., DEBIT/CREDIT AMOUNT). Notice that every record has a DEBIT/CREDIT AMOUNT exceeding $1,000 as shown in Figure 3.



Assignment Number	Document type		Posting key	Net due date	Accntg. Doc. Number	Clearing Doc.Number	Debit/Credit Amount
7585010510	RV	Billing doc.transfer	01	07/25/2013	7585010510	#	$ 1,943.77
0702131004 00	RV	Billing doc.transfer	01	07/25/2013	7585010512	#	$ 1,238.64

Figure 3 BEx Query Results after Condition Has Been Enforced

Tip 41

Suppressing Records with Zero Values in BEx Query Designer

You can clean up end users' reporting output by suppressing records with zero values in the BEx Query Designer.

When reporting against an SAP NetWeaver BW InfoCube, there are instances when cube compression hasn't occurred yet, and you may return rows of data that have values of zero. From an end-user reporting perspective, this may present some confusing data when master data has changed, but records show up with zero values. To overcome this situation, there is a setting within the BEx Query Designer that suppresses rows with zero values from showing up in the query results. We'll show you how to adjust this setting in this tip.

✓ And Here's How ...

Open the BEx Query Designer, and open an existing BEx query. Then follow these steps:

1. In the open query, click on the PROPERTIES button on the toolbar (, see Figure 1).

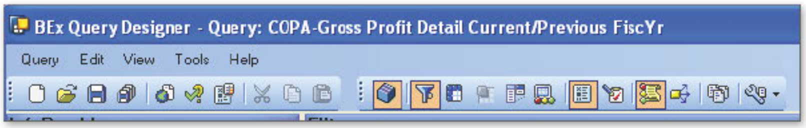
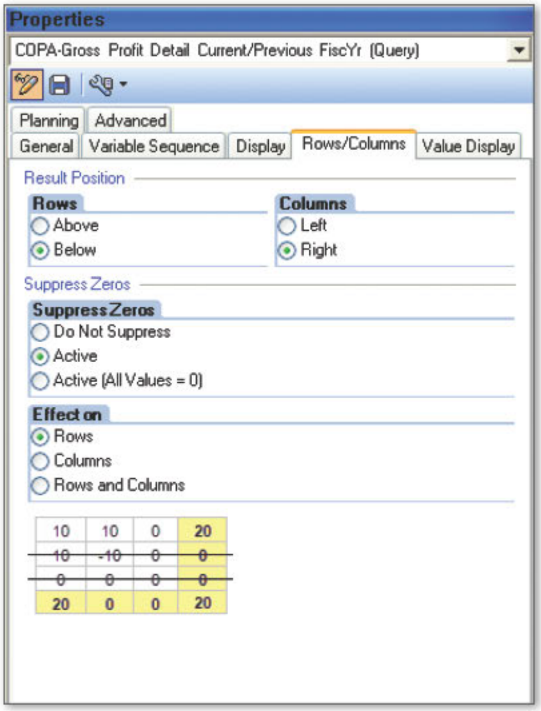

BEx Query Designer - Query: COPA-Gross Profit Detail Current/Previous FiscYr
Query
Edit
View
Tools
Help

Figure 1 BEx Query Designer Toolbar

2. From the PROPERTIES menu, click on the ROW/COLUMNS tab.
3. Select the ACTIVE radio button under the SUPPRESS ZEROS subheader, and then close the PROPERTIES menu and execute the query to view the results (Figure 2).



COPA-Gross Profit Detail Current/Previous FiscYr (Query)

Result Position

Rows: Above [✓ Below]

Columns: Left [✓ Right]

Suppress Zeros

Suppress Zeros: Do Not Suppress [✓ Active] Active (All Values = 0)

Effect on: [✓ Rows] Columns Rows and Columns

10	10	0	20
~~10~~	~~-10~~	~~0~~	~~0~~
~~0~~	~~0~~	~~0~~	~~0~~
20	0	0	20

Figure 2 Suppress Zeros Active Selection

The data set that is returned will not display any records where key figures have a zero balance selection.

Tip 42

Making Multiple Changes to a Query Seamlessly in BEx Analyzer

You can save time by minimizing the query refresh processing that occurs after each change you make to a BEx Analyzer query.

Often in BEx Analyzer, it's very easy to click and drag, and then add or remove fields. If you need to make multiple changes at once and don't want to wait for the report to refresh on every change, the following tip facilitates this process. The obvious benefit is to customize the query by including row, columns, and conditions from a single user interface before refreshing the query.

With the help of this tip, we'll show you how to easily and quickly make multiple changes from a single BEx Analyzer menu.

✓ And Here's How ...

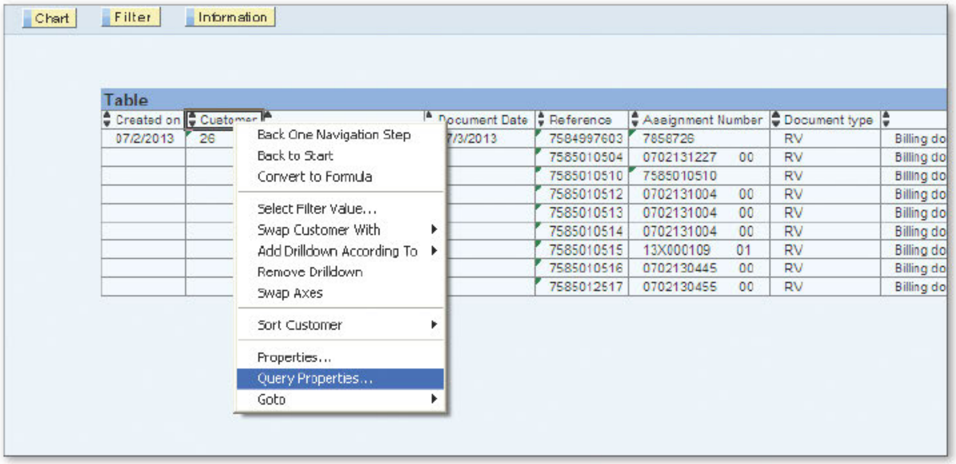
To make the required changes through the BEx Analyzer, open the BEx Analyzer and execute an existing BEx query. After the query is open, follow these steps:

1. Right-click on the result section, and select QUERY PROPERTIES (Figure 1).

   The pop up window opens with three main sections where you can make all the required changes to the query at one time and have the system refresh at once (see Figure 2). From left to right, you'll see the following sections:

   ▸ COLUMNS
   Features characteristic/key figure in the report that are being displayed in the columns.

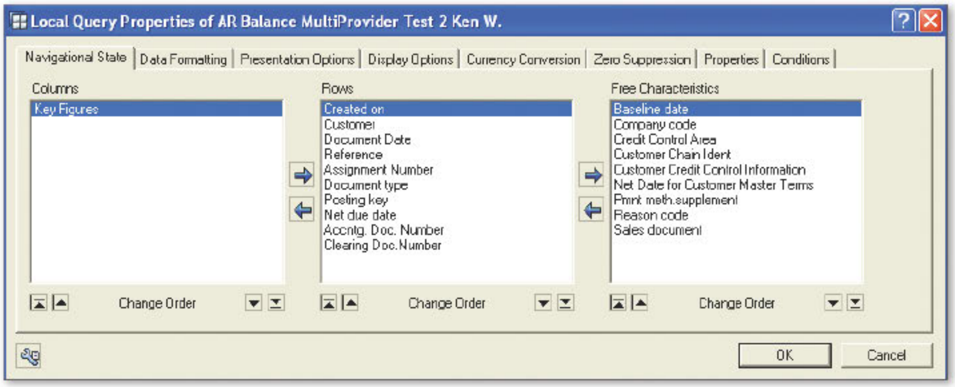
- Rows
  Features characteristics and key figures in the report that are being displayed in the rows.
- Free Characteristics
  Features free characteristics that provide the user with additional filtering options.



Table

Created on	Customer	Document Date	Reference	Assignment Number		Document type	
07/2/2013	26	7/3/2013	7584997603	7858726		RV	Billing do
			7585010504	0702131227	00	RV	Billing do
			7585010510	7585010510		RV	Billing do
			7585010512	0702131004	00	RV	Billing do
			7585010513	0702131004	00	RV	Billing do
			7585010514	0702131004	00	RV	Billing do
			7585010515	13X000109	01	RV	Billing do
			7585010516	0702130445	00	RV	Billing do
			7585012517	0702130455	00	RV	Billing do

Figure 1 Selecting the Query Properties Option from the Context Menu



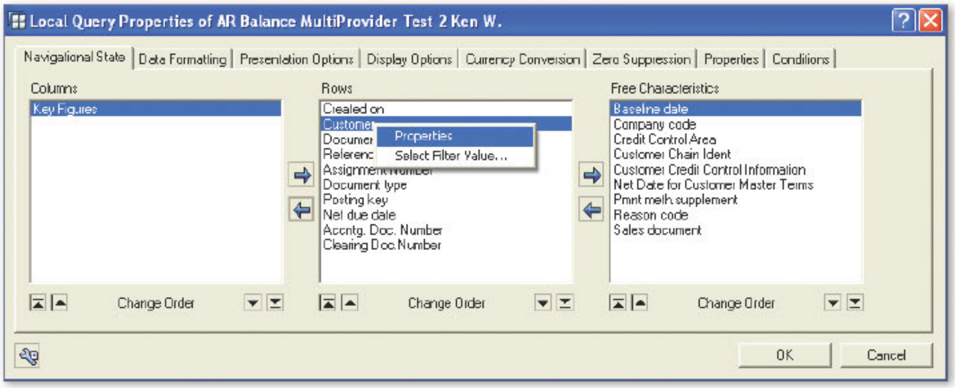
Local Query Properties of AR Balance MultiProvider Test 2 Ken W.

Columns	Rows	Free Characteristics
Key Figures	Created on	Baseline date
	Customer	Company code
	Document Date	Credit Control Area
	Reference	Customer Chain Ident
	Assignment Number	Customer Credit Control Information
	Document type	Net Date for Customer Master Terms
	Posting key	Pmnt meth.supplement
	Net due date	Reason code
	Accntg. Doc. Number	Sales document
	Clearing Doc.Number	

Figure 2 Popup Window with Columns, Rows, and Free Characteristics

2. Right-click on a key figure and choose Select Filter Value to select additional key figures that may be hidden.

3. Right-click on an InfoObject to then select or deselect attributes or apply report level filters (Figure 3).



Columns
Key Figures

Rows
Created on
Customer
Documen
Referenc
Assignment Number
Document type
Posting key
Net due date
Accntg. Doc. Number
Clearing Doc.Number

Free Characteristics
Baseline date
Company code
Credit Control Area
Customer Chain Ident
Customer Credit Control Information
Net Date for Customer Master Terms
Pmnt meth.supplement
Reason code
Sales document

Figure 3 Filtering on an InfoObject with the Select Filter Value Option

Exceptions and conditions can be activated or deactivated from this screen as well.

Tip 43

Combining Query Results by Setting Up Subqueries with Linked Variables

To include missing attributes and results for a more comprehensive view of a query run, you can pass the results from one query to another in BEx.

Sometimes, you may try to run a query from an InfoProvider that doesn't contain all of the characteristic attributes you need. Additionally, the inclusion of navigational attributes is prohibitive due to additional development time or adverse performance. To bypass this issue and get all of the information you need, you can set up a variable to seamlessly pass the results from one query to another (note that there must be a common characteristic between the two queries). Query 1 is the summary view, while Query 2 contains additional details; by filtering on a customer attribute in Query 1, you can essentially drill down into Query 2 to get additional details about the customer.

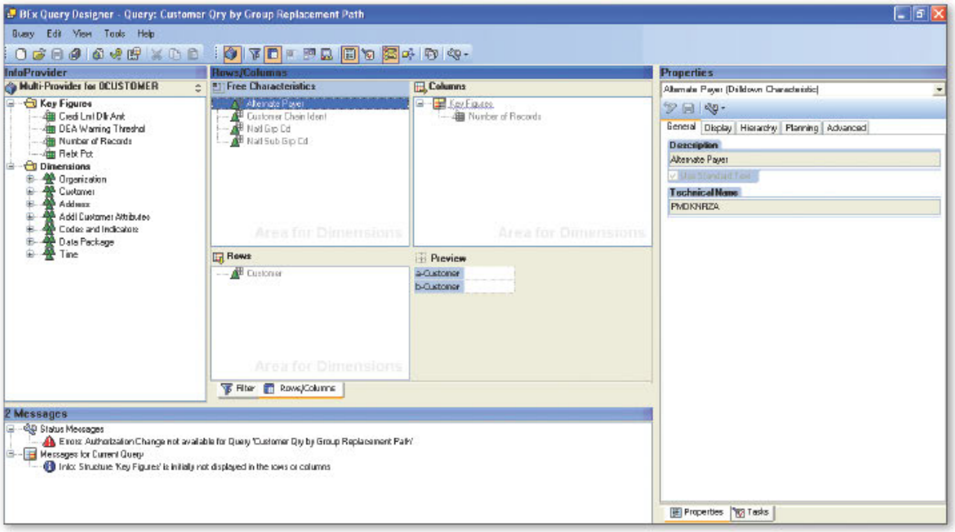
In this tip, we'll show you how to create a variable that is associated with the main query and returns a data set that will be used in another query. Ultimately, this process will teach you to use dynamic subqueries in SAP NetWeaver BW.

✓ And Here's How ...

When there's a common variable between two queries, follow these steps to set up the BEx query to pass the results from one query to another:

1. Create a BEx query that will provide the necessary selections and variable prompts to return the required fields and data. For example, Query 1 (Figure 1)

will return a count of customers based on the selected filters or characteristic restrictions (Figure 2).


BEx Query Designer - Query: Customer Qry by Group Replacement Path

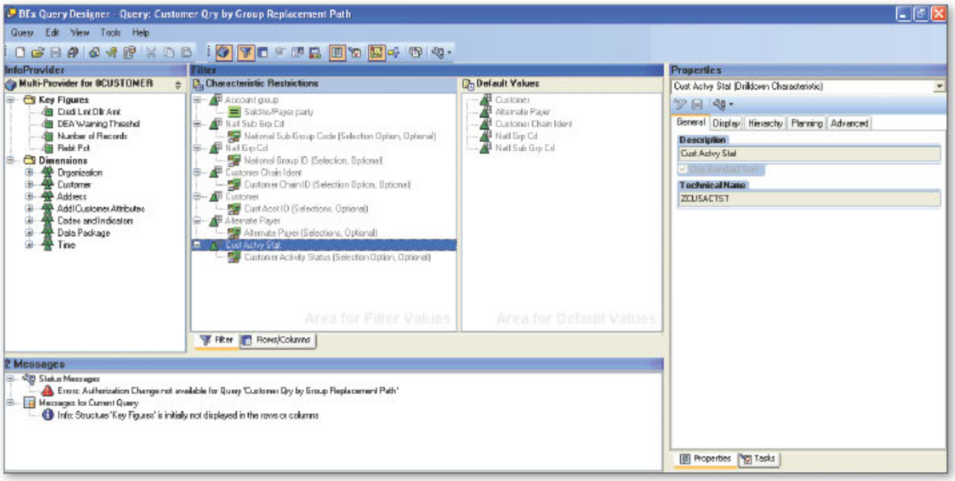
InfoProvider: Multi-Provider for 0CUSTOMER
- Key Figures: Credit Lmt Dlr Amt; DEA Warning Threshd; Number of Records; Rebt Pct
- Dimensions: Organization; Customer; Address; Addl Customer Attributes; Codes and Indicators; Data Package; Time

Rows/Columns
- Free Characteristics: Alternate Payer; Customer Chain Ident; Natl Grp Cd; Natl Sub Grp Cd
- Columns: Key Figures – Number of Records
- Rows: Customer
- Preview: a-Customer; b-Customer

Properties: Alternate Payer (Drilldown Characteristic)
- Description: Alternate Payer
- Technical Name: PMDKNRZA

2 Messages
- Status Messages
  - Error: Authorization Change not available for Query 'Customer Qry by Group Replacement Path'
- Messages for Current Query
  - Info: Structure 'Key Figures' is initially not displayed in the rows or columns

Figure 1 Query 1 with Customers in Rows and Number of Records in Columns


BEx Query Designer - Query: Customer Qry by Group Replacement Path

InfoProvider: Multi-Provider for 0CUSTOMER
- Key Figures: Credit Lmt Dlr Amt; DEA Warning Threshd; Number of Records; Rebt Pct
- Dimensions: Organization; Customer; Address; Addl Customer Attributes; Codes and Indicators; Data Package; Time

Filter
- Characteristic Restrictions:
  - Account group – Soldto/Payer party
  - Natl Sub Grp Cd – National Sub-Group Code (Selection Option, Optional)
  - Natl Grp Cd – National Group ID (Selection, Optional)
  - Customer Chain Ident – Customer Chain ID (Selection Option, Optional)
  - Customer – Cust Acct ID (Selections, Optional)
  - Alternate Payer – Alternate Payer (Selections, Optional)
  - Cust Actvy Stat – Customer Activity Status (Selection Option, Optional)
- Default Values: Customer; Alternate Payer; Customer Chain Ident; Natl Grp Cd; Natl Sub Grp Cd

Properties: Cust Actvy Stat (Drilldown Characteristic)
- Description: Cust Actvy Stat
- Technical Name: ZCUSACTST

2 Messages
- Status Messages
  - Error: Authorization Change not available for Query 'Customer Qry by Group Replacement Path'
- Messages for Current Query
  - Info: Structure 'Key Figures' is initially not displayed in the rows or columns

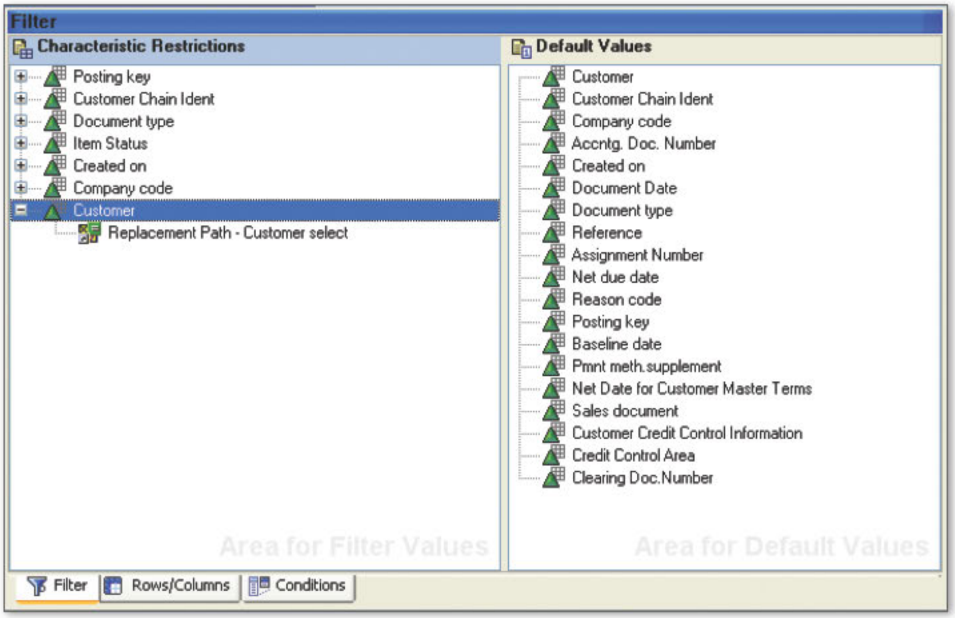
Figure 2 Query 1 with the Required Filter: Customer

The variables that are assigned to Query 1 can be optional or mandatory.

2. Create a new variable for a customer characteristic that will reference Query 1. Fill in the description (here we use "Replacement Path – Customer select") and

technical name ("ZCUSTSELT"). On the GLOBAL SETTINGS, assign the PROCESSING BY as "replacement path" and the REFERENCE CHARACTERISTIC as "Customer" or the characteristic you are passing the results to.

3. Click on the REPLACEMENT PATH tab, then select QUERY as the REPLACE VARIABLE WITH, and assign QUERY 1 to it.
4. Create the second BEx query—Query 2—with additional characteristics and key figures (subreport).
5. Assign the newly created variable to CUSTOMER in Query 2 (Figure 3). The filters or characteristic restrictions that are passed from Query 1 do not exist in Query 2.



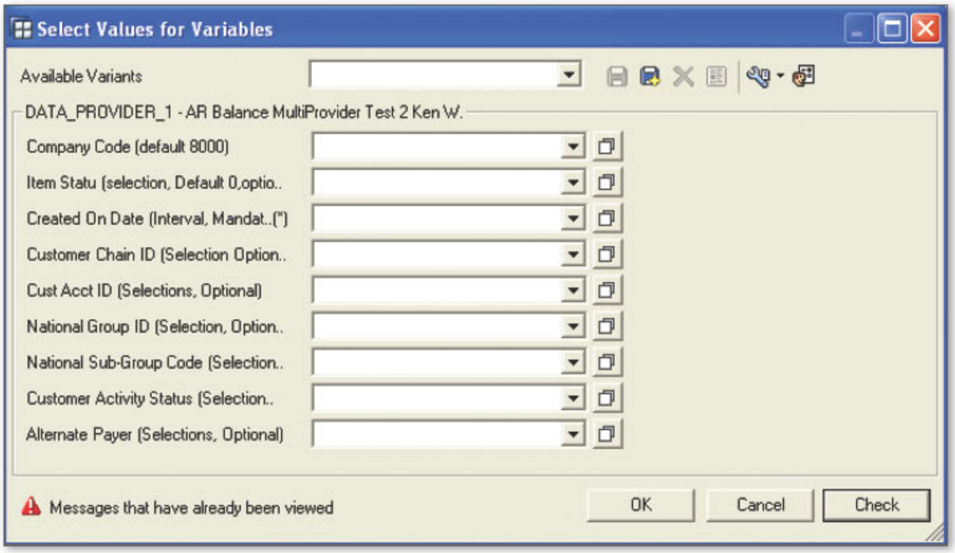
Characteristic Restrictions:
- Posting key
- Customer Chain Ident
- Document type
- Item Status
- Created on
- Company code
- Customer
  - Replacement Path - Customer select

Default Values:
- Customer
- Customer Chain Ident
- Company code
- Accntg. Doc. Number
- Created on
- Document Date
- Document type
- Reference
- Assignment Number
- Net due date
- Reason code
- Posting key
- Baseline date
- Pmnt meth.supplement
- Net Date for Customer Master Terms
- Sales document
- Customer Credit Control Information
- Credit Control Area
- Clearing Doc.Number

Figure 3 Customer Variable Assigned to Customer in Query 2

6. Finally, execute Query 2; the variable from Query 1 will now be passed to Query 2.

In summary, you've filtered on a customer attribute in Query 1 that passes the CUSTOMER ACCT ID to the second query to return additional details not present in the first query. For example, CUSTOMER ACTIVITY STATUS variable from Query 1 passes the CUSTOMER ACCT ID to the CUST ACCT ID variable in Query 2 (see Figure 4).



DATA_PROVIDER_1 - AR Balance MultiProvider Test 2 Ken W.

Company Code (default 8000)
Item Statu (selection, Default 0,optio..
Created On Date (Interval, Mandat..(*)
Customer Chain ID (Selection Option..
Cust Acct ID (Selections, Optional)
National Group ID (Selection, Option..
National Sub-Group Code (Selection..
Customer Activity Status (Selection..
Alternate Payer (Selections, Optional)

Messages that have already been viewed

Figure 4 Customer Activity Status Variable from Query 1 Passing the Customer Acct ID to the Cust Acct ID Variable in Query 2

Tip 44

Leveraging Existing Data Range References in the BEx Analyzer for Data Manipulation

You can reuse existing data range references in BEx Analyzer and use them for future data analysis to save time.

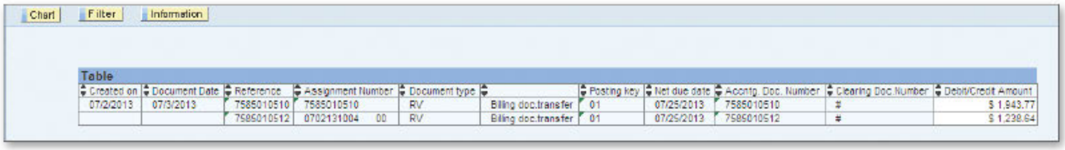
In SAP NetWeaver BW 7.x, BEx Analyzer creates a range reference in the FORMULAS section of Excel. This range comes in handy when you're performing additional data manipulation to your results from SAP NetWeaver BW. For example, you can use this formula range in a vlookup (`vlookup(A1,DF_GRID_1,2,False)`). When the query is refreshed, workbook macros run and update the formula references, which are reflected in the Excel formula. This prevents the need to create a large range of cells that the formula would have to read or create a data range that is too small to encompass the entire table. The end result for the end user is the reusability of data sets via range references in BEx Analyzer.

Through this tip, you can leverage data range references for use in other data-related tasks in the BEx Analyzer. This is a frontend tool in the BEx Analyzer that is used by developers and end users. If a support person is developing a report that requires combining two queries, this range setup can automatically update when the BEx workbook is refreshed.

And Here's How ...

To view the existing references and use them for further data analysis, follow these steps:

1. Open an existing BEx query in Analyzer and refresh it (Figure 1).



Table

Created on	Document Date	Reference	Assignment Number	Document type		Posting key	Net due date	Acctg. Doc. Number	Clearing Doc.Number	Debit/Credit Amount
07/2/2013	07/3/2013	7585010510	7585010510	RV	Billing doc.transfer	01	07/25/2013	7585010510	#	$ 1,943.77
		7585010512	0702131004 00	RV	Billing doc.transfer	01	07/25/2013	7585010512	#	$ 1,238.64

Figure 1 *Existing BEx Query*

2. When the refresh is complete, click on the FORMULAS tab in the toolbar in BEx Analyzer in Office 2007 as shown in Figure 2.

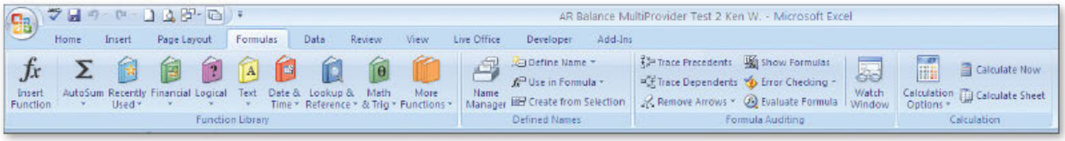
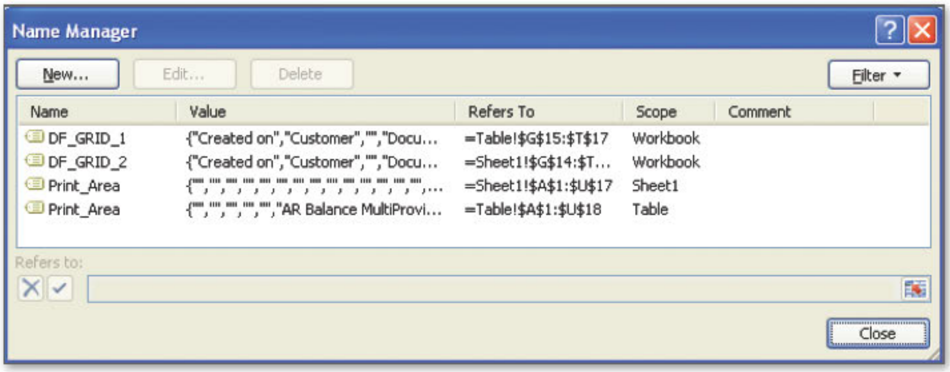


Figure 2 *Toolbar with Name Manager Link*

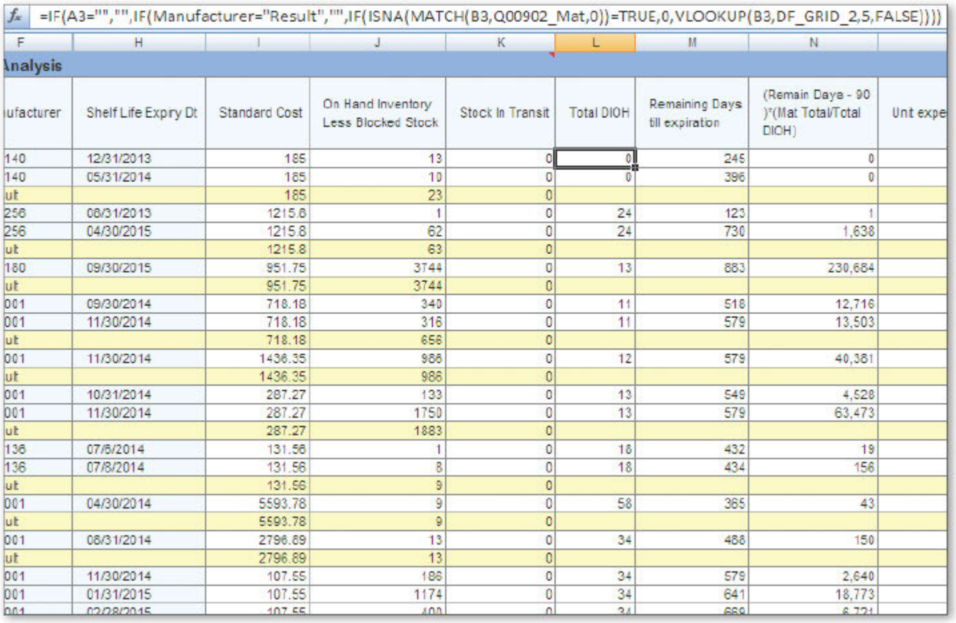
3. The list of available data range names is provided. These data range names are created by the BEx Analyzer based on the number of data providers in your workbook and are updated on each refresh based on the size of the data set that is returned. If more than one data provider exists on your workbook, then options similar to Figure 3 will be displayed.



Name	Value	Refers To	Scope	Comment
DF_GRID_1	{"Created on","Customer","","Docu...	=Table!$G$15:$T$17	Workbook	
DF_GRID_2	{"Created on","Customer","","Docu...	=Sheet1!$G$14:$T...	Workbook	
Print_Area	{"","","","","","","","","","","","","","",...	=Sheet1!$A$1:$U$17	Sheet1	
Print_Area	{"","","","","","AR Balance MultiProvi...	=Table!$A$1:$U$18	Table	

Figure 3 *Available Data Range References When More Than One Data Provider Exists on Your Workbook*

4. The following view (Figure 4) demonstrates how the data range reference can be used. The highlighted cell is populated via the statement function, which includes a reference to the `DF_GRID` data range which is normally coded in Excel as Table!$G$3:$K$545.



```
=IF(A3="","",IF(Manufacturer="Result","",IF(ISNA(MATCH(B3,Q00902_Mat,0))=TRUE,0,VLOOKUP(B3,DF_GRID_2,5,FALSE))))
```

F	H	I	J	K	L	M	N	
Analysis								
ufacturer	Shelf Life Expiry Dt	Standard Cost	On Hand Inventory Less Blocked Stock	Stock In Transit	Total DIOH	Remaining Days till expiration	(Remain Days - 90)*(Mat Total/Total DIOH)	Unt expe
140	12/31/2013	185	13	0	0	245	0	
140	05/31/2014	185	10	0	0	396	0	
ult		185	23	0				
256	08/31/2013	1215.8	1	0	24	123	1	
256	04/30/2015	1215.8	62	0	24	730	1,638	
ult		1215.8	63	0				
180	09/30/2015	951.75	3744	0	13	883	230,684	
ult		951.75	3744	0				
001	09/30/2014	718.18	340	0	11	518	12,716	
001	11/30/2014	718.18	316	0	11	579	13,503	
ult		718.18	656	0				
001	11/30/2014	1436.35	986	0	12	579	40,381	
ult		1436.35	986	0				
001	10/31/2014	287.27	133	0	13	549	4,528	
001	11/30/2014	287.27	1750	0	13	579	63,473	
ult		287.27	1883	0				
136	07/6/2014	131.56	1	0	18	432	19	
136	07/8/2014	131.56	8	0	18	434	156	
ult		131.56	9	0				
001	04/30/2014	5593.78	9	0	58	365	43	
ult		5593.78	9	0				
001	08/31/2014	2796.89	13	0	34	488	150	
ult		2796.89	13	0				
001	11/30/2014	107.55	186	0	34	579	2,640	
001	01/31/2015	107.55	1174	0	34	641	18,773	
001	02/28/2015	107.55	400	0	34	669	6,721	

Figure 4 The If Function Leveraging Data Range Reference DF_GRID

Tip 45

Returning a List of BEx Queries that Use Specified SAP NetWeaver BW BEx Variables

If you make a change to a commonly used BEx variable, you can generate a list of BEx queries that consume the specified variable to know which queries have changed.

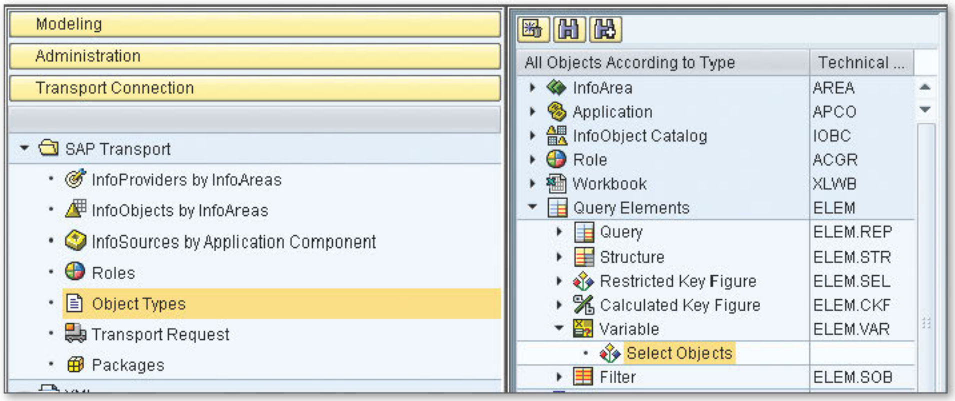
Because BEx variables can be re-used for multiple BEx queries, any changes to the variable may affect many queries and ultimately, the end users who consume these reports. If the dependent queries are not updated, users may have inconsistent and contradictory query results. Similar to metadata management capabilities, this tip quickly enables the user to perform data lineage and traceability analysis on BEx variables to view all BEx queries that consume it.

For example, a variable property may need to be changed from *Mandatory* to *Optional*. In that case, we need to update all dependant queries with the change. In order to find out all dependent queries, you can use this simple tip.

✓ And Here's How ...

To return a list of queries that are dependent upon a changed BEx variable, execute SAP NetWeaver BW Transaction RSA1 and follow these steps:

1. In the resulting screen, select Transport Connection (see the left pane in Figure 1) and then select Variable/Select Object. In this example, we've chosen object 0I_DAYIN as the variable for which we'd like to return the list.

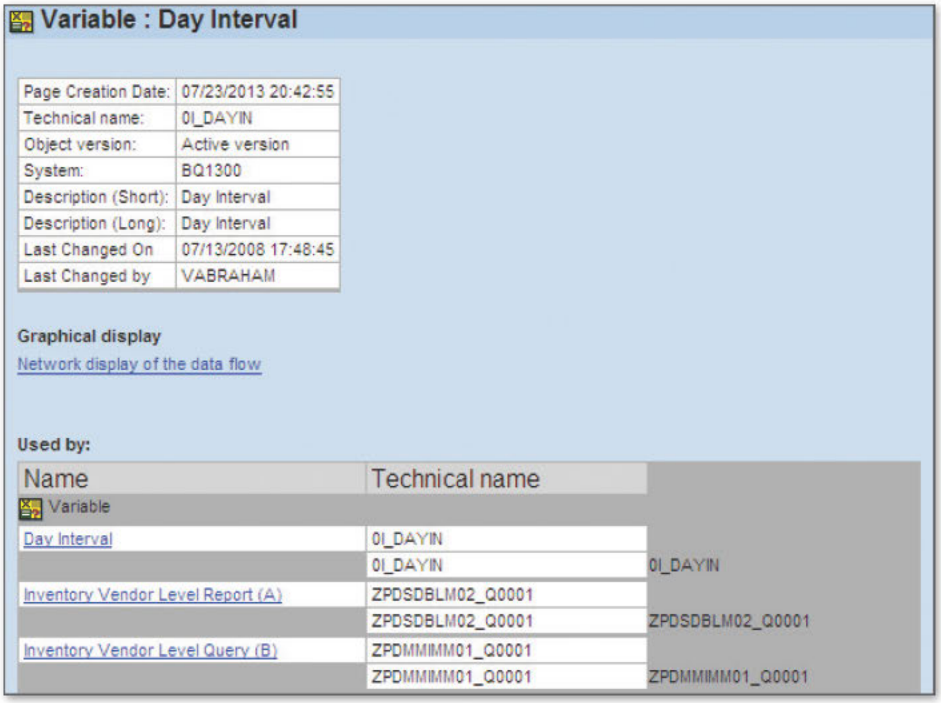


All Objects According to Type	Technical ...
InfoArea	AREA
Application	APCO
InfoObject Catalog	IOBC
Role	ACGR
Workbook	XLWB
Query Elements	ELEM
Query	ELEM.REP
Structure	ELEM.STR
Restricted Key Figure	ELEM.SEL
Calculated Key Figure	ELEM.CKF
Variable	ELEM.VAR
Select Objects	
Filter	ELEM.SOB

Figure 1 Selection of BEx Variable 0I_DAYIN

2. Right-click on the variable and choose Display description.

The results are displayed as shown in Figure 2, listing of all BEx queries that use the BEx variable 0I_DAYIN.



Variable : Day Interval

Page Creation Date:	07/23/2013 20:42:55
Technical name:	0I_DAYIN
Object version:	Active version
System:	BQ1300
Description (Short):	Day Interval
Description (Long):	Day Interval
Last Changed On	07/13/2008 17:48:45
Last Changed by	VABRAHAM

Graphical display

Network display of the data flow

Used by:

Name	Technical name	
Variable		
Day Interval	0I_DAYIN	
	0I_DAYIN	0I_DAYIN
Inventory Vendor Level Report (A)	ZPDSDBLM02_Q0001	
	ZPDSDBLM02_Q0001	ZPDSDBLM02_Q0001
Inventory Vendor Level Query (B)	ZPDMMIMM01_Q0001	
	ZPDMMIMM01_Q0001	ZPDMMIMM01_Q0001

Figure 2 Listing of all BEx Queries that use BEx Variable 0I_DAYIN

Part 3

SAP NetWeaver BW Data Flow

Things You‘ll Learn in this Section

SAP NetWeaver BW data flow deals with the path the data takes through the various data warehouse layers until it is ready for reporting. Data flow includes the entire process of extracting the data from multiple source systems, transforming the data (by implementing technical cleanup logic as well as business rules), and loading the data in InfoProviders within SAP NetWeaver BW. This section will provide insight into some key topics in the extraction, transformation, and loading (ETL) process.

A process chain in SAP NetWeaver BW is a critical element of the ETL process that's used to automate the complex schedules in SAP NetWeaver BW with the help of the event-controlled processing, visual schedules, and centrally controlled and monitored processes. We will cover some key topics that have to do with the creation and management of process chains.

We'll also discuss some tips that will help you to build effective transformations in the data flow. Additionally, we provide exclusive insights to the performance of the data loads.

Tip 46

Analyzing Process Chains for Troubleshooting or Optimizing Data Load Processes

You can use an SAP-delivered standard program to perform extensive analysis of process chains.

In an SAP NetWeaver BW support environment, it's very important to be aware of statistics and runtime information about scheduled process chains. This information can be used to optimize the data-loading process for the SAP NetWeaver BW system or to troubleshoot long-running data load processes. The common method to access the process chain log is by using Transaction RSPC; however, this transaction doesn't provide you with the capability to analyze and compare process chain steps.

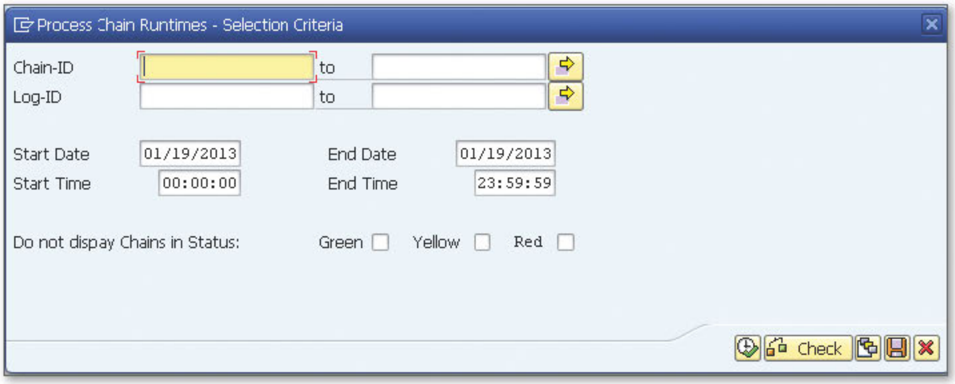
This tip discusses how to use a standard SAP-delivered program, /SSA/BWT, to analyze process chains in detail. The results from this analysis can be used to identify bottlenecks in the data-loading process and refine the steps in the process chain process.

✓ And Here's How ...

Program /SSA/BWT provides advanced analytical capabilities and can be very useful in a support mode. It helps monitor the daily loads for any failures and compares the runtime of specific process chain steps over a period of time in case of unusual changes to the load time. Follow these steps to execute the program:

1. Call up Transaction SE38. Select Program /SSA/BWT and click on EXECUTE. In the resulting SAP NetWeaver BW tools screen, you'll see the following options:
   - PROCESS CHAIN ANALYSIS
     Allows for analysis of a process chain either at the process chain level or process type level.
   - DETAILED REQUEST ANALYSIS
     Allows for analysis by a specific request. The technical name of the request ID is needed. This can be found from the InfoProvider/DSO management screen.
   - AGGREGATE TOOLSET
     Allows for analysis of the aggregates associated with a particular InfoCube.
   - TEMPLATE REPORT ANALYSIS
     Allows for analysis of specific workbooks/queries.
   - INFOPROVIDER ANALYSIS
     Provides details of InfoCube (information related to InfoCube design, SID tables and NRIV information, databases indexes, statistics, etc.).
2. The default selection is PROCESS CHAIN RUNTIME ANALYSIS. Click on EXECUTE. In the next screen, two options are presented:
   - ANALYSIS BY PROCESS CHAINS
     Specify one or more process chain IDs that you want to analyze.
   - ANALYSIS BY PROCESS TYPES
     Specify process types as the selection for your analysis.

   Depending on your requirement, you may use either one of the options. If you choose the ANALYSIS BY PROCESS CHAINS option, the screen in Figure 1 is presented.

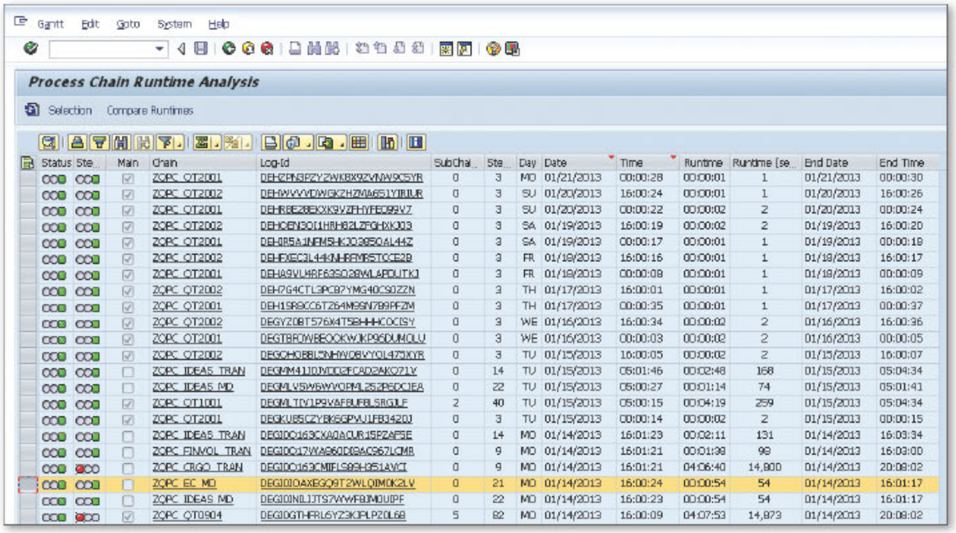

Process Chain Runtimes - Selection Criteria

Chain-ID		to	
Log-ID		to	
Start Date	01/19/2013	End Date	01/19/2013
Start Time	00:00:00	End Time	23:59:59

Do not dispay Chains in Status: Green ☐ Yellow ☐ Red ☐

Figure 1 Process Chain Runtime Analysis Selection Entry

3. Specify the selection criteria as desired. If you want to monitor the loads for a specific date, just specify the START DATE and END DATE. If you want to monitor the runtime of a particular process chain over several days, specify the CHAIN-ID (technical name of the process chain) and specify the START DATE/END DATE to cover that date range.
4. After you click on EXECUTE, the screen shown in Figure 2 appears. At a summary level report, this will display the following: number of steps in the chain, run day, and start and end date/time of process chain run. It displays the process chain run time in hours, minutes, and seconds (HH:MM:SS), and displays the runtime in seconds. Here you can see all of the process chains that fulfill the criteria you specified earlier.
5. You can also use this report to navigate into a specific run of the process chain by clicking the hyperlink on the CHAIN name to display more details about the steps (in a hierarchical display view) in the chain, such as start date/time, step run time, number of records processed, InfoProvider, or DataSource if relevant for the step. You can add more information fields to add to the display besides the default fields available on the summary report.

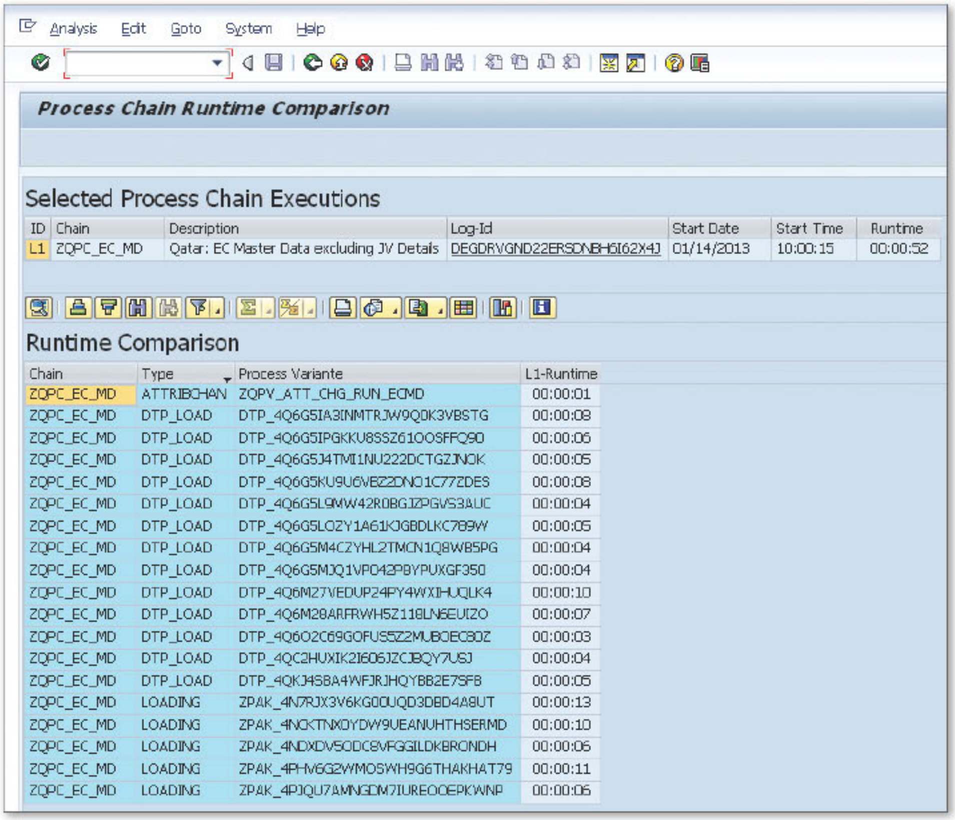


Status	Ste...	Main	Chain	Log-Id	SubChai...	Ste...	Day	Date	Time	Runtime	Runtime [se...	End Date	End Time
🟩	🟩	✓	ZQPC_QT2001	DEHZPNGPZY2WK8X9ZVNW9C5YR	0	3	MO	01/21/2013	00:00:28	00:00:01	1	01/21/2013	00:00:30
🟩	🟩	✓	ZQPC_QT2002	DEHWVVVDWGKZHZMA651YIRIUR	0	3	SU	01/20/2013	16:00:24	00:00:01	1	01/20/2013	16:00:26
🟩	🟩	✓	ZQPC_QT2001	DEHRBE28EKXK9VZFHYFEO99V7	0	3	SU	01/20/2013	00:00:22	00:00:02	2	01/20/2013	00:00:24
🟩	🟩	✓	ZQPC_QT2002	DEHOEN3OI1HRH82LZFGHXKJ03	0	3	SA	01/19/2013	16:00:19	00:00:02	2	01/19/2013	16:00:20
🟩	🟩	✓	ZQPC_QT2001	DEH0R5A1NFM5HKJO3850AL44Z	0	3	SA	01/19/2013	00:00:17	00:00:01	1	01/19/2013	00:00:18
🟩	🟩	✓	ZQPC_QT2002	DEHFXEC3L44KNHBRMRSTCCE2B	0	3	FR	01/18/2013	16:00:16	00:00:01	1	01/18/2013	16:00:17
🟩	🟩	✓	ZQPC_QT2001	DEHA9VU4RF63SO28WLAPDUTKJ	0	3	FR	01/18/2013	00:00:08	00:00:01	1	01/18/2013	00:00:09
🟩	🟩	✓	ZQPC_QT2002	DEH7G4CTL3PCB7YMG40CS0ZZN	0	3	TH	01/17/2013	16:00:01	00:00:01	1	01/17/2013	16:00:02
🟩	🟩	✓	ZQPC_QT2001	DEH1SR8CC6T264M9GN7B9PFZM	0	3	TH	01/17/2013	00:00:35	00:00:01	1	01/17/2013	00:00:37
🟩	🟩	✓	ZQPC_QT2002	DEGYZ0BT576X4T5BHHHCOCISY	0	3	WE	01/16/2013	16:00:34	00:00:02	2	01/16/2013	16:00:36
🟩	🟩	✓	ZQPC_QT2001	DEGTBROWBEOOKWJKP96DUMOLU	0	3	WE	01/16/2013	00:00:03	00:00:02	2	01/16/2013	00:00:05
🟩	🟩	✓	ZQPC_QT2002	DEGOHOBBL5NHWO8VYOL475XYR	0	3	TU	01/15/2013	16:00:05	00:00:02	2	01/15/2013	16:00:07
🟩	🟩		ZQPC_IDEAS_TRAN	DEGMM41IDJVDD2FCAD2AKO71V	0	14	TU	01/15/2013	05:01:46	00:02:48	168	01/15/2013	05:04:34
🟩	🟩		ZQPC_IDEAS_MD	DEGMLV5W6WVOPML252P6DCJEA	0	22	TU	01/15/2013	05:00:27	00:01:14	74	01/15/2013	05:01:41
🟩	🟩	✓	ZQPC_QT1001	DEGMLTIV1P9VAFBUFBLSRGJLF	2	40	TU	01/15/2013	05:00:15	00:04:19	259	01/15/2013	05:04:34
🟩	🟩	✓	ZQPC_QT2001	DEGKUB5CZYBK6GPVU1FB3420J	0	3	TU	01/15/2013	00:00:14	00:00:02	2	01/15/2013	00:00:15
🟩	🟩		ZQPC_IDEAS_TRAN	DEGIDO163CXA0ACUR15PZAF5E	0	14	MO	01/14/2013	16:01:23	00:02:11	131	01/14/2013	16:03:34
🟩	🟩		ZQPC_FINVOL_TRAN	DEGIDO17WA960DI9AC967LCMR	0	9	MO	01/14/2013	16:01:21	00:01:38	98	01/14/2013	16:03:00
🟩	🟥		ZQPC_CRGO_TRAN	DEGIDO163CMIFLS89H351AVCI	0	9	MO	01/14/2013	16:01:21	04:06:40	14,800	01/14/2013	20:08:02
🟩	🟩		ZQPC_EC_MD	DEGIDIOAXEGQ9T2WLQIMDK2LV	0	21	MO	01/14/2013	16:00:24	00:00:54	54	01/14/2013	16:01:17
🟩	🟩		ZQPC_IDEAS_MD	DEGIDINLJJTS7WWFBJMOUIPF	0	22	MO	01/14/2013	16:00:23	00:00:54	54	01/14/2013	16:01:17
🟩	🟥	✓	ZQPC_QT0904	DEGIDGTHFRL6YZ3KJPLPZ0L68	5	82	MO	01/14/2013	16:00:09	04:07:53	14,873	01/14/2013	20:08:02

Figure 2 Results of the Process Chain Runtime Analysis

6. By clicking on the SELECTION button in Figure 2, you can filter the displayed list using CHAIN-ID, LOG-ID, START AND END DATES, and START AND END TIMES.

7. By clicking on COMPARE RUNTIMES, the system will take you to the details of the process chain steps and the time spent on each step (see Figure 3).



Process Chain Runtime Comparison

Selected Process Chain Executions

ID	Chain	Description	Log-Id	Start Date	Start Time	Runtime
L1	ZQPC_EC_MD	Qatar: EC Master Data excluding JV Details	DEGDRVGND22ERSDNBH6I62X4J	01/14/2013	10:00:15	00:00:52

Runtime Comparison

Chain	Type	Process Variante	L1-Runtime
ZQPC_EC_MD	ATTRIBCHAN	ZQPY_ATT_CHG_RUN_ECMD	00:00:01
ZQPC_EC_MD	DTP_LOAD	DTP_4Q6G5IA3INMTRJW9QDK3VBSTG	00:00:08
ZQPC_EC_MD	DTP_LOAD	DTP_4Q6G5IPGKKU8SSZ61OOSFFQ90	00:00:06
ZQPC_EC_MD	DTP_LOAD	DTP_4Q6G5J4TMI1NU222DCTGZJNOK	00:00:05
ZQPC_EC_MD	DTP_LOAD	DTP_4Q6G5KU9U6VBZ2DNO1C77ZDES	00:00:08
ZQPC_EC_MD	DTP_LOAD	DTP_4Q6G5L9MW42R0BGIZPGVS3AUC	00:00:04
ZQPC_EC_MD	DTP_LOAD	DTP_4Q6G5LOZY1A61KJGBDLKC789W	00:00:05
ZQPC_EC_MD	DTP_LOAD	DTP_4Q6G5M4CZYHL2TMCN1Q8WB5PG	00:00:04
ZQPC_EC_MD	DTP_LOAD	DTP_4Q6G5MJQ1VP042PBYPUXGF350	00:00:04
ZQPC_EC_MD	DTP_LOAD	DTP_4Q6M27VEDUP24PY4WXIHUQLK4	00:00:10
ZQPC_EC_MD	DTP_LOAD	DTP_4Q6M28ARFRWH5Z118LN6EUIZO	00:00:07
ZQPC_EC_MD	DTP_LOAD	DTP_4Q6O2C69GOFUS5Z2MUBOEC8OZ	00:00:03
ZQPC_EC_MD	DTP_LOAD	DTP_4QC2HUXIK2I606JZCJBQY7USJ	00:00:04
ZQPC_EC_MD	DTP_LOAD	DTP_4QKJ4SBA4WFJRJHQYBB2E7SFB	00:00:05
ZQPC_EC_MD	LOADING	ZPAK_4N7RJX3V6KG0OUQD3DBD4A8UT	00:00:13
ZQPC_EC_MD	LOADING	ZPAK_4NCKTNXOYDW9UEANUHTHSERMD	00:00:10
ZQPC_EC_MD	LOADING	ZPAK_4NDXDV5ODC8VFGGILDKBRONDH	00:00:06
ZQPC_EC_MD	LOADING	ZPAK_4PHV6G2WMOSWH9G6THAKHAT79	00:00:11
ZQPC_EC_MD	LOADING	ZPAK_4PJQU7AMNGDM7IUREOOEPKWNP	00:00:06

Figure 3 Process Chain Runtime Comparison

Alternately, you may choose the ANALYSIS BY PROCESS TYPES option shown earlier in Figure 1.

8. Specify the selection criteria as desired, and click on EXECUTE. The resulting screen will show the process chains that meet the selection criteria.

You can perform further analysis for a specific process chain from this screen by clicking on the CHAIN-ID or LOG-ID.

Tip 47

Populating Internal Tables That Are Available across Multiple Data Packages

You can easily improve data load performance by using a static attribute to store table content that's used across multiple data packages during SAP NetWeaver BW data loads.

Data load performance is very important in SAP NetWeaver BW; it helps deliver report content on time and can reduce lead time for reporting. One controllable factor that affects load performance is how SAP NetWeaver BW custom transformations are implemented for data enrichment, data massaging, and calculations. In typical custom transformation routines, several table lookups (configuration data, master data, transaction data, etc.) are used for data enrichment or calculations. Lookup table content is specific to a data package and not visible across data packages so these lookup tables need to be populated for each data package resulting in additional overhead in terms of database processing time.

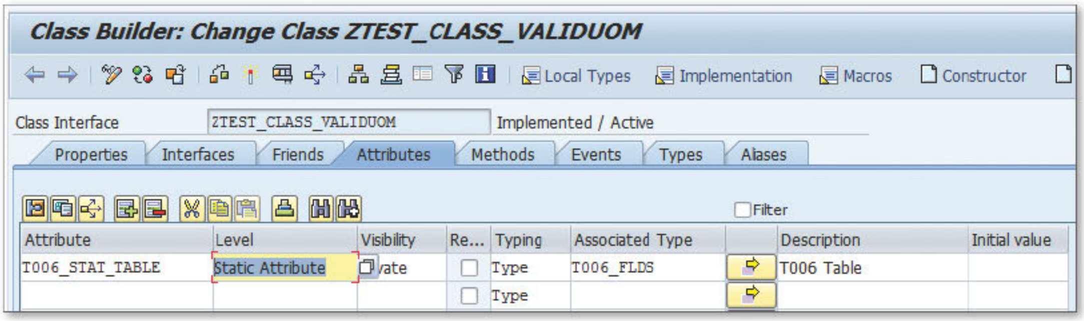
In SAP NetWeaver BW transformations, there are several scenarios in which the same table content is needed across multiple data packages for processing, which can be achieved using *static attributes*. This will reduce the total number of lookups needed for each data load; instead of one lookup per data package for the same table content, only one lookup is used for the entire load.

This tip provides details on how to enable table content in transformation that's visible across multiple data packages to reduce database processing time.

And Here's How ...

In this tip, we'll show you an example of how to use static attributes in transformation by using the class method used in a transformation routine. In this transformation routine, we want to validate incoming data units of measure (UOMs) and replace invalid UOMs with a space. Follow these steps:

1. Create a class using Class Builder Transaction SE24. Enter the name "ZTEST_CLASS_VALIDUOM" in the Object type field.
2. Create a static attribute internal table to store the table content of the T006 UOM table using the Attributes tab of the class (see Figure 1).



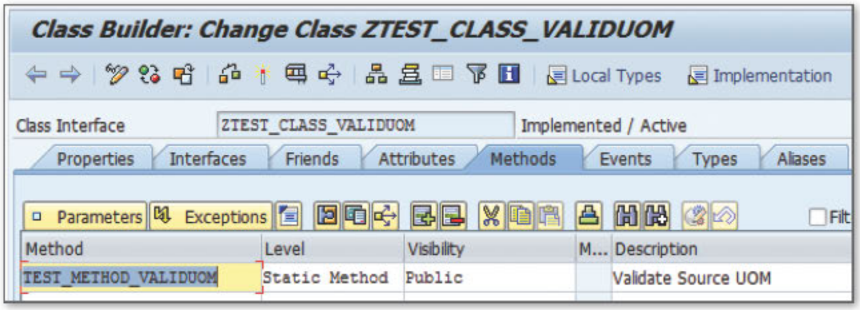
Class Builder: Change Class ZTEST_CLASS_VALIDUOM

Class Interface ZTEST_CLASS_VALIDUOM Implemented / Active

Attribute	Level	Visibility	Re...	Typing	Associated Type		Description	Initial value
T006_STAT_TABLE	Static Attribute	Private	☐	Type	T006_FLDS	⇨	T006 Table	
			☐	Type		⇨		

Figure 1 Creating a Static Attribute Internal Table in the Class Builder

3. Now move to the Methods tab (see Figure 2), and create a class method named "TEST_METHOD_VALIDUOM" to implement lookup of the T006 table, store T006 table content in a static internal table, and store UOM validation logic in the method. In the method implementation, the T006 table lookup only executes if there is no content in the static T006 internal table, which will be filled during the first time of method execution.



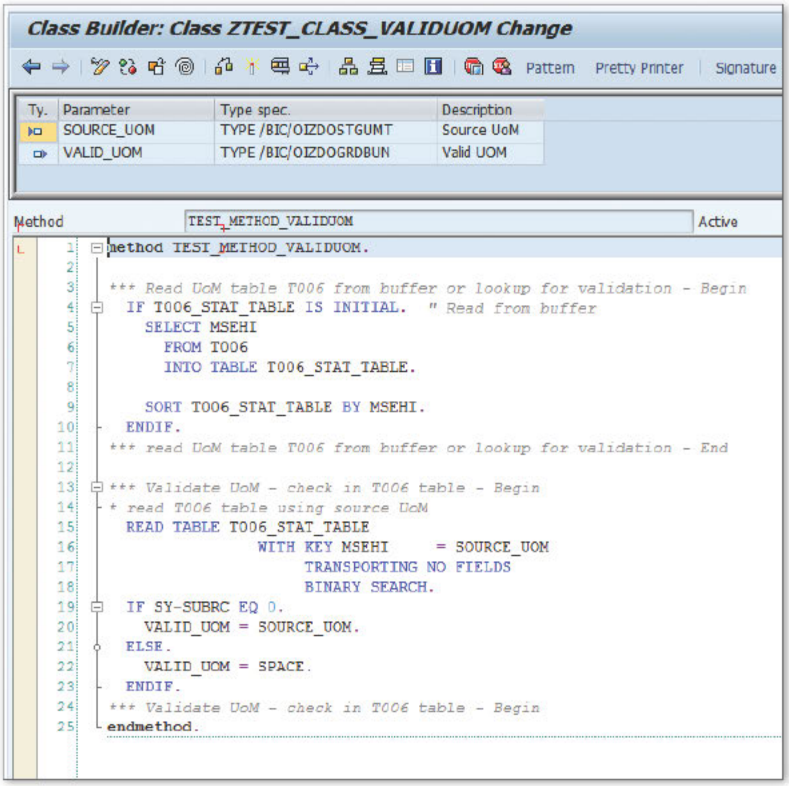
Class Builder: Change Class ZTEST_CLASS_VALIDUOM

Class Interface ZTEST_CLASS_VALIDUOM Implemented / Active

Method	Level	Visibility	M...	Description
TEST_METHOD_VALIDUOM	Static Method	Public		Validate Source UOM

Figure 2 Creating a Method for the Class

4. Double-click on the method TEST_METHOD_VALIDUOM, and enter code for the method in the resulting screen (see Figure 3). The code should populate the internal table in this method. In this scenario, we're reading the UOM table T006 and saving the results into the static attribute table T006_STAT_TABLE that is defined for the class in the earlier step.



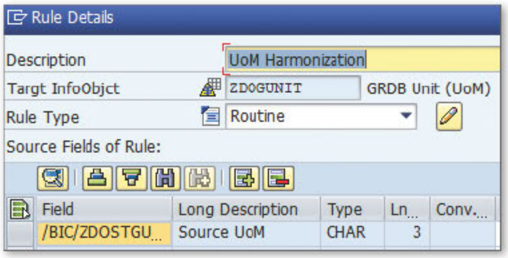
Ty.	Parameter	Type spec.	Description
	SOURCE_UOM	TYPE /BIC/OIZDOSTGUMT	Source UoM
	VALID_UOM	TYPE /BIC/OIZDOGRDBUN	Valid UOM

```
method TEST_METHOD_VALIDUOM.

*** Read UoM table T006 from buffer or lookup for validation - Begin
  IF T006_STAT_TABLE IS INITIAL.  " Read from buffer
    SELECT MSEHI
      FROM T006
      INTO TABLE T006_STAT_TABLE.

    SORT T006_STAT_TABLE BY MSEHI.
  ENDIF.
*** read UoM table T006 from buffer or lookup for validation - End

*** Validate UoM - check in T006 table - Begin
* read T006 table using source UoM
  READ TABLE T006_STAT_TABLE
                WITH KEY MSEHI     = SOURCE_UOM
                     TRANSPORTING NO FIELDS
                     BINARY SEARCH.
  IF SY-SUBRC EQ 0.
    VALID_UOM = SOURCE_UOM.
  ELSE.
    VALID_UOM = SPACE.
  ENDIF.
*** Validate UoM - check in T006 table - Begin
endmethod.
```

Figure 3 Coding the Method to Lookup UOM and Save Results to the Static Attribute Internal Table

5. Now call the class method in the transformation routine for validation. Go to the transformations of the DataStore Object (DSO) or InfoCube where this logic needs to be implemented. For our scenario, we'll use mapping rules for an InfoObject ZDOGUNIT as an example to illustrate how to call this method in the transformation rules.

   In the transformations, go to the Rule Details option for the InfoObject ZDOGUNIT, as shown in Figure 4. Select Rule Type Routine, and click on the Change button.



Rule Details

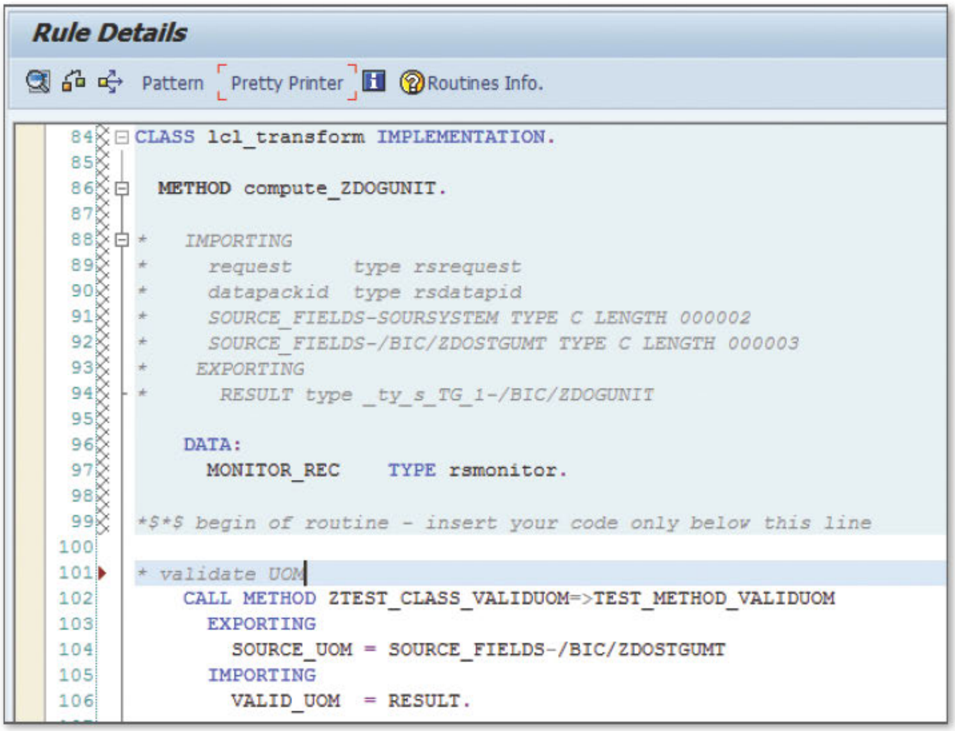
Description	UoM Harmonization		
Targt InfoObjct	ZDOGUNIT	GRDB Unit (UoM)	
Rule Type	Routine		

Source Fields of Rule:

Field	Long Description	Type	Ln...	Conv...
/BIC/ZDOSTGU...	Source UoM	CHAR	3	

Figure 4 Transformation Rules for ZDOGUNIT

6. Figure 5 will be displayed where you can include code to call the class method TEST_METHOD_VALIDUOM that was created in the earlier step.



```
CLASS lcl_transform IMPLEMENTATION.

  METHOD compute_ZDOGUNIT.

*   IMPORTING
*     request     type rsrequest
*     datapackid  type rsdatapid
*     SOURCE_FIELDS-SOURSYSTEM TYPE C LENGTH 000002
*     SOURCE_FIELDS-/BIC/ZDOSTGUMT TYPE C LENGTH 000003
*    EXPORTING
*      RESULT type _ty_s_TG_1-/BIC/ZDOGUNIT

    DATA:
      MONITOR_REC    TYPE rsmonitor.

*$*$ begin of routine - insert your code only below this line

* validate UOM
    CALL METHOD ZTEST_CLASS_VALIDUOM=>TEST_METHOD_VALIDUOM
      EXPORTING
        SOURCE_UOM = SOURCE_FIELDS-/BIC/ZDOSTGUMT
      IMPORTING
        VALID_UOM  = RESULT.
```

Figure 5 Coding the Rule Details to Call the Class Method TEST_METHOD_VALIDUOM

This method can be called in any transformation. Based on the logic, the internal table will be populated once, and the contents of the internal table are available across multiple data packages. During load, the T006 static internal table will be filled during first data package processing and will be used for all subsequent data package processing.

Tip 48

Correctly Populating the Process Keys for Logistics Extractors

With some additional steps in the SAP ERP source system, you can make sure that the process keys for logistics extractors are populated correctly when transferring data to SAP NetWeaver BW.

For modeling Logistics data in SAP NetWeaver BW, the concept of *process keys* is very important. Process keys are used to interpret MM document information that is transferred from SAP ERP to SAP NetWeaver BW. After activating the Logistics extractors, you need to follow a few steps in the source SAP ERP system to get the process key field populated correctly.

Fields BWGEO, BWGEOO, BWGVP, BWGVO, BWNETWR, and BWMNG in extractors 2LIS_02_SCL, 2LIS_02_ITM, 2LIS_03_BF, 2LIS_03_UM, and 2LIS_40_REVAL are not filled and updated in SAP NetWeaver BW correctly when you activate the DataSource and enable extraction of data using the Logistics Cockpit. To ensure these key figures are updated correctly in SAP NetWeaver BW, the BWVORG field needs to be filled in SAP ERP and mapped to the 0PROCESSKEY InfoObject in SAP NetWeaver BW.

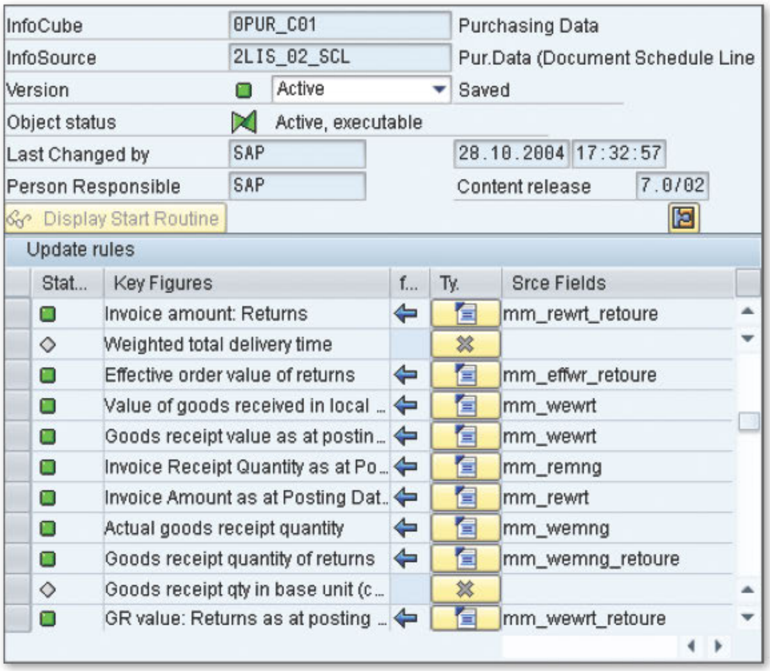
This tip shows you how to activate transaction/process keys for Logistics area extractors such as Purchasing and Inventory.

✓ And Here's How ...

As a first step, you *must* activate the determination of the process key via Transaction MCB_ in the SAP ERP system. In the first screen of the transaction, choose the industry sector that matches the SAP ERP implementation. STANDARD and

Consumer Products are for SAP ERP standard customers, whereas Retail is intended for customers with retail only.

After you perform this step, you can view the process keys available per application name (IS-R for IS Retail, MM for Materials Management, etc.) as shown in Figure 1 by using Transaction MCB0 or by navigating via the SAP IMG (Transaction SPRO by searching using the search string "Transaction Key Maintenance for SAP BW").



InfoCube	0PUR_C01	Purchasing Data
InfoSource	2LIS_02_SCL	Pur.Data (Document Schedule Line
Version	🟩 Active	Saved
Object status	Active, executable	
Last Changed by	SAP	28.10.2004 17:32:57
Person Responsible	SAP	Content release 7.0/02

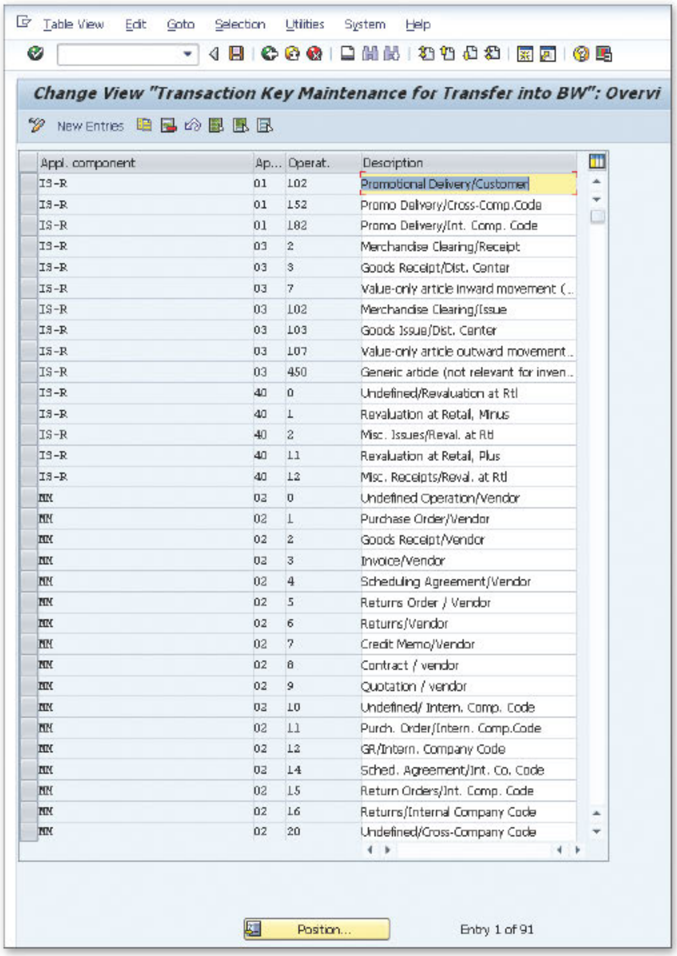
Display Start Routine

Update rules

Stat...	Key Figures	f...	Ty.	Srce Fields
🟩	Invoice amount: Returns	⬅	▤	mm_rewrt_retoure
◇	Weighted total delivery time		✖	
🟩	Effective order value of returns	⬅	▤	mm_effwr_retoure
🟩	Value of goods received in local ...	⬅	▤	mm_wewrt
🟩	Goods receipt value as at postin...	⬅	▤	mm_wewrt
🟩	Invoice Receipt Quantity as at Po...	⬅	▤	mm_remng
🟩	Invoice Amount as at Posting Dat...	⬅	▤	mm_rewrt
🟩	Actual goods receipt quantity	⬅	▤	mm_wemng
🟩	Goods receipt quantity of returns	⬅	▤	mm_wemng_retoure
◇	Goods receipt qty in base unit (c...		✖	
🟩	GR value: Returns as at posting ...	⬅	▤	mm_wewrt_retoure

Figure 1 Transaction Key Maintenance for SAP NetWeaver BW

Now navigate to your SAP NetWeaver BW system, and ensure that the BWVORG field is mapped to 0PROCESSKEY in the transformation as shown in Figure 2. After these fields are mapped, follow the LO-Cockpit extraction steps (initialization of data load using setup tables and enabling delta load using V3 delta) in the Wizard to update the data in SAP NetWeaver BW.



Change View "Transaction Key Maintenance for Transfer into BW": Overvi

Appl. component	Ap...	Operat.	Description
IS-R	01	102	Promotional Delivery/Customer
IS-R	01	152	Promo Delivery/Cross-Comp.Code
IS-R	01	182	Promo Delivery/Int. Comp. Code
IS-R	03	2	Merchandise Clearing/Receipt
IS-R	03	3	Goods Receipt/Dist. Center
IS-R	03	7	Value-only article inward movement (..
IS-R	03	102	Merchandise Clearing/Issue
IS-R	03	103	Goods Issue/Dist. Center
IS-R	03	107	Value-only article outward movement..
IS-R	03	450	Generic article (not relevant for inven..
IS-R	40	0	Undefined/Revaluation at Rtl
IS-R	40	1	Revaluation at Retail, Minus
IS-R	40	2	Misc. Issues/Reval. at Rtl
IS-R	40	11	Revaluation at Retail, Plus
IS-R	40	12	Misc. Receipts/Reval. at Rtl
MM	02	0	Undefined Operation/Vendor
MM	02	1	Purchase Order/Vendor
MM	02	2	Goods Receipt/Vendor
MM	02	3	Invoice/Vendor
MM	02	4	Scheduling Agreement/Vendor
MM	02	5	Returns Order / Vendor
MM	02	6	Returns/Vendor
MM	02	7	Credit Memo/Vendor
MM	02	8	Contract / vendor
MM	02	9	Quotation / vendor
MM	02	10	Undefined/ Intern. Comp. Code
MM	02	11	Purch. Order/Intern. Comp.Code
MM	02	12	GR/Intern. Company Code
MM	02	14	Sched. Agreement/Int. Co. Code
MM	02	15	Return Orders/Int. Comp. Code
MM	02	16	Returns/Internal Company Code
MM	02	20	Undefined/Cross-Company Code

Entry 1 of 91

Figure 2 Mapping of 0PROCESSSKEY

If these steps don't correctly populate the Process key field, set the application indicator BW to active using Transaction BF11 in the SAP ERP system.

The following SAP Notes provide information on how to use process keys for Inventory Management and Purchasing DataSources:

- Note 352344: Process Key + Reversals in Inventory Management
- Note 492828: Determining the Transaction Key for 2LIS_03_BF + 2LIS_03_UM
- Note 684465: BWVORG for Purchasing DataSources

Tip 49

Activating Transformations in an SAP NetWeaver BW System en Masse

You can perform a mass activation of inactive transformations in an SAP NetWeaver BW system without allowing changes to the system.

When you perform certain activities such as a system copy or upgrades, it's likely that some or all of the transformations will become inactive. The task of activating transformations one by one can be very tedious and time-consuming. Also, activating a transformation in a production system requires the system to be open for changes, which adds the risk of undesired changes to the system.

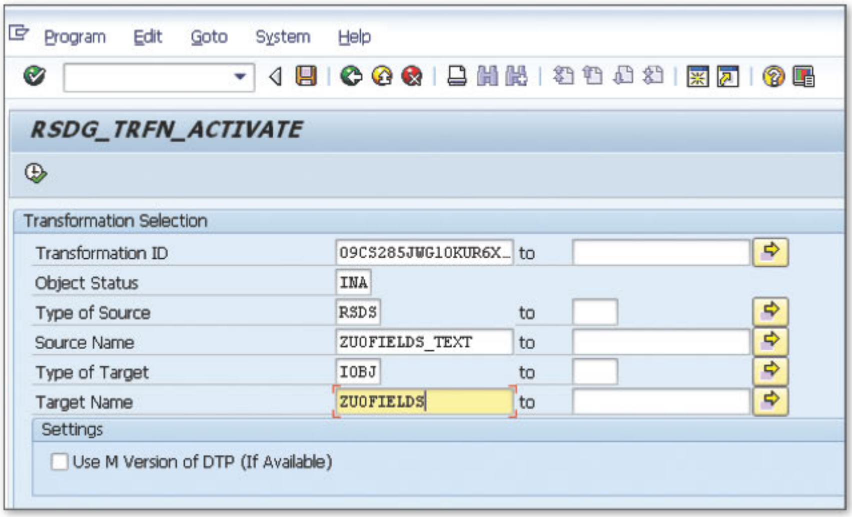
This tip provides step-by-step instructions for activating all transformations at once without opening the system for changes using a standard program available within the SAP system, RSDG_TRFN_ACTIVATE.

✓ And Here's How ...

In the case of a deactivation of transformations, the timestamps of the DataSources in the new source system (copied from the original source system of the original SAP NetWeaver BW system) differ from the timestamps of DataSources of the new SAP NetWeaver BW system (copied from the original SAP NetWeaver BW system). After this, when the DataSources are replicated, the impact handling during replication causes the transformations in the new SAP NetWeaver BW system to become inactive.

The following method can be used to perform a mass activation of transformations or activation of specific transformations.

1. Run Transaction SE38, and enter the PROGRAM "RSDG_TRFN_ACTIVATE". Click the EXECUTE button (F8) to run the program.
2. After the program has run, the system will ask you for the input parameters. In the TRANSFORMATION ID field, enter the ID of the transformation, or enter the range of transformation IDs if more than one transformation needs to be activated (see Figure 1).



Transformation Selection

Transformation ID	09CS285JWG10KUR6X_	to	
Object Status	INA		
Type of Source	RSDS	to	
Source Name	ZUOFIELDS_TEXT	to	
Type of Target	IOBJ	to	
Target Name	ZUOFIELDS	to	

Settings

☐ Use M Version of DTP (If Available)

Figure 1 Selection Screen to Specify the Transformations to Be Activated

For mass activation of transformations, leave the TRANSFORMATION ID field blank with no input and specify the SOURCE NAME and the TARGET NAME. All of the transformation objects in the system will be regenerated for specified source and target systems. If the program will be run without any selections, all objects of the transformation type (Table RSTRAN) will be checked and regenerated. Data Transfer Processes (DTPs) containing transformation of source-/target-type InfoSource also will be reactivated.

If you need to reactivate an already active transformation, you need to enter "ACT" in the OBJECT STATUS (OBJSTAT) field for that particular transformation (TRAN_ID).

You can also specify technical restrictions such as LAST CHANGED BY and LAST GENERATION TIME STAMP to restrict the selections further.

Tip 50

Adjusting the System's Default Package Size to Optimize Data Loading

You can use one of three methods to set up/tune the package size for SAP NetWeaver BW loads by defining threshold values for data loading.

The threshold value determines the number of records within an individual package that are extracted to SAP NetWeaver BW. SAP provides a default threshold value, but you have the option to change it to get the optimal results when performing the extraction. The higher the threshold value, the larger the number of records extracted within a data package, and hence the higher the memory consumption.

In this tip, we'll discuss three different methods to set the threshold value to optimize the data loading memory consumption.

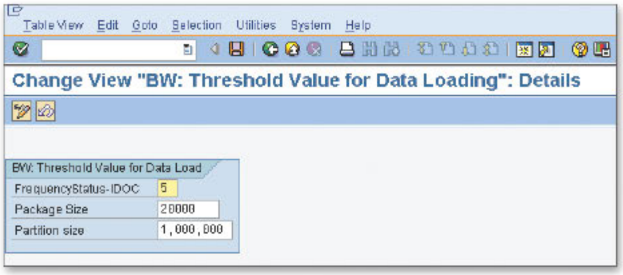
✓ And Here's How ...

While evaluating the threshold value, also consider whether the records are added/deleted during the start/end routines. There are three different methods to make the setting for the package size, which we'll discuss in the following sections:

- The global setting for the threshold value is set in Transaction RSCUSTV6.
- The default global value can be changed at the InfoPackage level.
- If semantic groups are defined, then this can be changed at the individual DTP level by specifying it in the DTP definition.

Change Global Settings

To change the global settings, execute Transaction RSCUSTV6. The screen shown in Figure 1 will appear. Specify the package size and click on SAVE.



BW: Threshold Value for Data Load

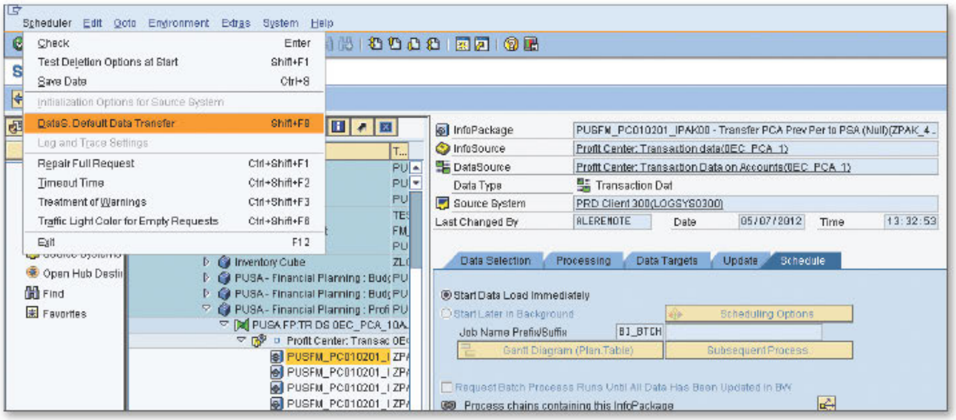
FrequencyStatus-IDOC	5
Package Size	20000
Partition size	1,000,000

Figure 1 Transaction RSCUSTV6 Screen to Specify the Package Size

While setting the PACKAGE SIZE, it's important to size it appropriately so that the data isn't spread across too many packages; this has a negative impact on performance. Ideally, the number of data packages per load process should not be higher than 100. The basic setting for PACKAGE SIZE should be between 5,000 and 20,000 records. To load a large volume of transaction data, the packet size can be changed from the default value to a value between 10,000 and 50,000 data records per package.

Change Package Size at the InfoPackage Level

To change the package size at the InfoPackage level, open up the InfoPackage screen by clicking on SCHEDULER • DATAS. DEFAULT DATA TRANSFER (see Figure 2).


Scheduler
Edit
Goto
Environment
Extras
System
Help
Check Enter
Test Deletion Options at Start Shift+F1
Save Data Ctrl+S
Initialization Options for Source System
DataS. Default Data Transfer Shift+F8
Log and Trace Settings
Repair Full Request Ctrl+Shift+F1
Timeout Time Ctrl+Shift+F2
Treatment of Warnings Ctrl+Shift+F3
Traffic Light Color for Empty Requests Ctrl+Shift+F8
Exit F12
Open Hub Desti
Find
Favorites
Inventory Cube
PUSA - Financial Planning : Bud
PUSA - Financial Planning : Bud
PUSA - Financial Planning : Profi
PUSA FP:TR DS 0EC_PCA_10A
Profit Center: Transac 0E
PUSFM_PC010201_I
PUSFM_PC010201_I
PUSFM_PC010201_I
PUSFM_PC010201_I
Data Selection
Processing
Data Targets
Update
Schedule
Scheduling Options
Gantt Diagram (Plan.Table)
Subsequent Process

InfoPackage	PUSFM_PC010201_IPAK00 - Transfer PCA Prev Per to PSA (Null)(ZPAK_4_
InfoSource	Profit Center: Transaction data(0EC_PCA_1)
DataSource	Profit Center: Transaction Data on Accounts(0EC_PCA_1)
Data Type	Transaction Dat
Source System	PRD Client 300(LOGSYS0300)
Last Changed By	ALEREMOTE
Date	05/07/2012
Time	13:32:53

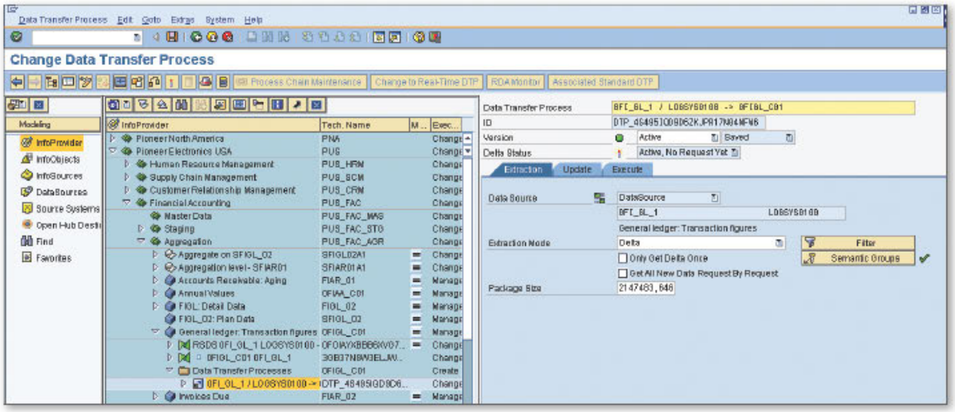
[✓ Start Data Load Immediately]
Start Later in Background
Job Name Prefix/Suffix BI_BTCH
Request Batch Process Runs Until All Data Has Been Updated in BW
Process chains containing this InfoPackage

Figure 2 InfoPackage Setting

The default is 20,000 records. This can be changed in the following screen.

Change Package Size at DTP Level

To change it at the DTP level, select the DTP, right-click, and select CHANGE. You can change the PACKAGE SIZE as shown in Figure 3.



InfoProvider	Tech. Name	M...	Exec...
Pioneer North America	PNA		Change
Pioneer Electronics USA	PUS		Change
Human Resource Management	PUS_HRM		Change
Supply Chain Management	PUS_SCM		Change
Customer Relationship Management	PUS_CRM		Change
Financial Accounting	PUS_FAC		Change
Master Data	PUS_FAC_MAS		Change
Staging	PUS_FAC_STG		Change
Aggregation	PUS_FAC_AGR		Change
Aggregate on SFIGL_02	SFIGL02A1	=	Change
Aggregation level - SFIAR01	SFIAR01A1	=	Change
Accounts Receivable: Aging	FIAR_01	=	Manage
Annual Values	OFIAA_C01	=	Manage
FIGL: Detail Data	FIGL_02	=	Manage
FIGL_02: Plan Data	SFIGL_02	=	Manage
General ledger: Transaction figures	0FIGL_C01	=	Manage
RSDS 0FI_GL_1 LOGSYS0100 -	0FOIAYXBBB6XVO7...	=	Change
0FIGL_C01 0FI_GL_1	3GB37N8WJELWV...		Change
Data Transfer Processes	0FIGL_C01		Create
0FI_GL_1 / LOGSYS0100 ->	DTP_48495IGD9C6...		Change
Invoices Due	FIAR_02	=	Manage

Data Transfer Process	0FI_GL_1 / LOGSYS0100 -> 0FIGL_C01
ID	DTP_46495IQ09062KJPR17N04MFN6
Version	Active / Saved
Delta Status	Active, No Request Yet
Data Source	DataSource
	0FI_GL_1 LOGSYS0100
	General ledger: Transaction figures
Extraction Mode	Delta
	☐ Only Get Delta Once
	☐ Get All New Data Request By Request
Package Size	2147483,648

Figure 3 Changing the Setting at the DTP Level

Tip 51

Restarting a Failed Full Repair Request for a DSO

You can use a standard program to repair and restart a failed full repair request during data load.

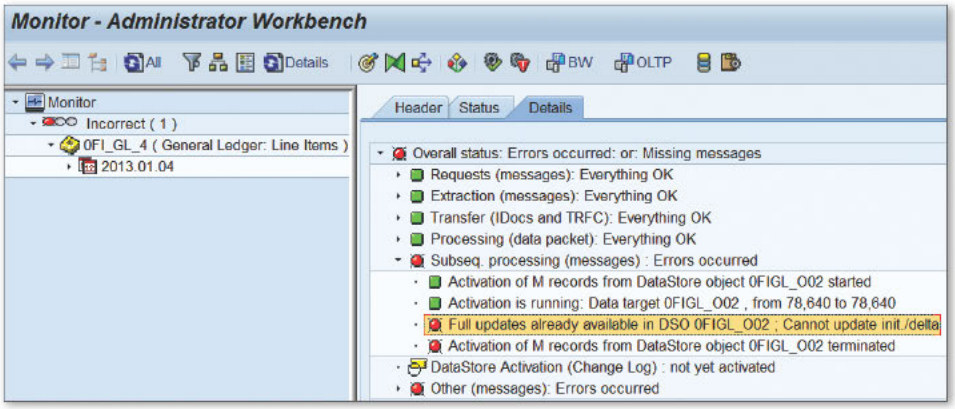
While performing data loads using the delta load functionality, it's common to execute full repair loads to recover missing records using a full load InfoPackage. Before the full repair load is executed, it's critical that an indicator on the InfoPackage is set to FULL REPAIR REQUEST. However, this step is commonly overlooked, resulting in failed loads. Additionally, all subsequent loads will also fail during the activation of *M* records during the delta upload to the DSO with a message stating "Full updates already available in DSO; cannot update init./delta."

This tip reiterates the importance of setting the FULL REPAIR REQUEST flag correctly while performing a full load to DSO and will also describe steps to recover the data loads in case of failure.

✓ And Here's How ...

The following scenario is used to demonstrate this tip: The activation of *M* records fails during the delta upload to DSO with a message stating that the full updates are already available in DSO and cannot update the init./delta further. Follow these steps:

1. Go to the Administrator Workbench in SAP NetWeaver BW, and click the DETAILS tab as shown in Figure 1. In the case of a failed load, you'll see the red stoplight with the message mentioned previously.

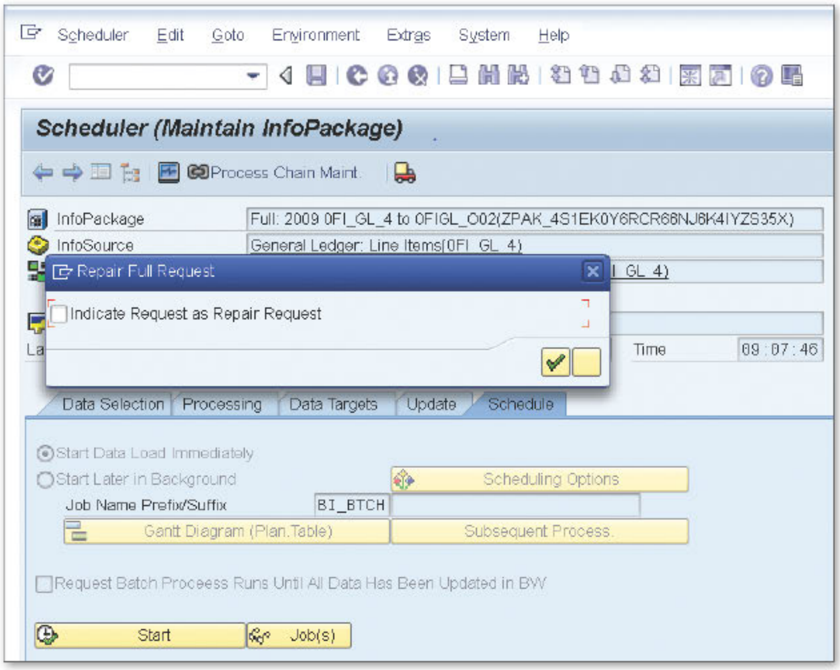


Monitor
Incorrect (1)
0FI_GL_4 (General Ledger: Line Items)
2013.01.04

Overall status: Errors occurred: or: Missing messages
Requests (messages): Everything OK
Extraction (messages): Everything OK
Transfer (IDocs and TRFC): Everything OK
Processing (data packet): Everything OK
Subseq. processing (messages) : Errors occurred
Activation of M records from DataStore object 0FIGL_O02 started
Activation is running: Data target 0FIGL_O02 , from 78,640 to 78,640
Full updates already available in DSO 0FIGL_O02 ; Cannot update init./delta
Activation of M records from DataStore object 0FIGL_O02 terminated
DataStore Activation (Change Log) : not yet activated
Other (messages): Errors occurred

Figure 1 Monitor Screen Showing Failed Full Repair Load

The common cause of this error is that the INDICATE REQUEST AS REPAIR REQUEST flag is not set while performing the full load to DSO, as shown in Figure 2.



InfoPackage: Full: 2009 0FI_GL_4 to 0FIGL_O02(ZPAK_4S1EK0Y6RCR66NJ6K4IYZS35X)
InfoSource: General Ledger: Line Items(0FI_GL_4)
_GL_4)
La
Time: 09 : 07 : 46

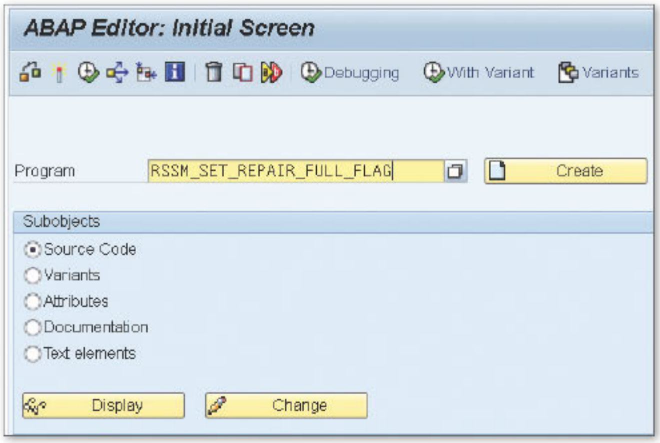
☐ Indicate Request as Repair Request

● Start Data Load Immediately
○ Start Later in Background
Job Name Prefix/Suffix: BI_BTCH
☐ Request Batch Process Runs Until All Data Has Been Updated in BW

Figure 2 Setting in the InfoPackage Where the Indicator for Repair Request is Set

Execute the following steps to restart the data loads and continue with the delta loads:

1. Go to Transaction SE38/Transaction SE80, enter the program name RSSM_SET_REPAIR_FULL_FLAG, and click EXECUTE (see Figure 3).


ABAP Editor: Initial Screen

Debugging | With Variant | Variants

Program: RSSM_SET_REPAIR_FULL_FLAG

Create

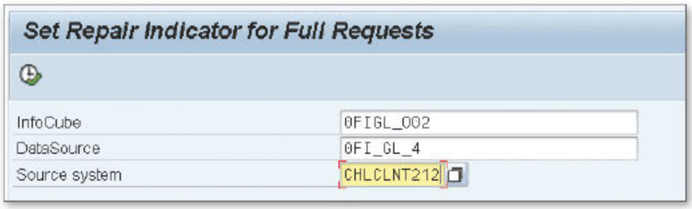
Subobjects

(●) Source Code
( ) Variants
( ) Attributes
( ) Documentation
( ) Text elements

Display | Change

Figure 3 Transaction SE38 Screen with the Standard SAP Program

2. In the following screen, enter parameters as required and click EXECUTE. In this case, enter the INFOCUBE, DATASOURCE, and SOURCE SYSTEM (see Figure 4).


Set Repair Indicator for Full Requests

InfoCube	0FIGL_002
DataSource	0FI_GL_4
Source system	CHLCLNT212

Figure 4 Selection Screen for Program RSSM_SET_REPAIR_FULL_FLAG

3. In the subsequent screen, click the GENERATE REPAIR FULL button in the top-left corner. Click YES in the popup that appears for confirmation.
4. The system will display a success message. Go back to the DSO activation queue and reactivate the delta request. The delta request should now get activated successfully.

This method can be used to resolve issues with failed full repair loads and to restart the delta process.

Tip 52

Using a Standard Table to Analyze and Delete Data from Transfer Process Error Logs

You can use standard ABAP Report RSBM_ERRORLOG_DELETE to analyze and delete DTP error logs and free up database space in the system.

When a Data Transfer Process (DTP) fails, it creates an error log in the system. If you don't clean up these error logs on a regular basis, the table size will grow and use up database space. Using SAP standard programs, you can analyze these error logs and delete them.

This tip describes the functionality of analyzing DTP error logs and deleting DTP error logs from Table RSBERRORLOG.

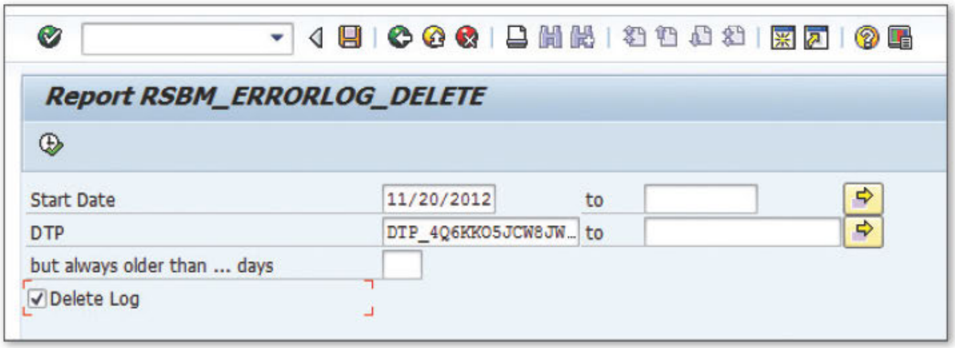
✓ And Here's How ...

To use the standard ABAP report, execute Transaction SE38 in the SAP NetWeaver BW system and follow these steps:

1. To analyze the error log, enter the program name "RSB_ANALYZE_ERRORLOG" in the ABAP Editor: Initial Screen.
2. Click the Display button, and you'll see a screen that shows a list of the DTP technical names and error messages.

Now, you want to delete the error logs. Execute Transaction SE38 in the SAP NetWeaver BW system and follow these steps:

1. Enter the Program name RSB_ANALYZE_ERRORLOG and click Execute.
2. In the resulting screen shown in Figure 1, enter the DTP technical name and Start Date from which you want to delete the error logs. Click Execute.

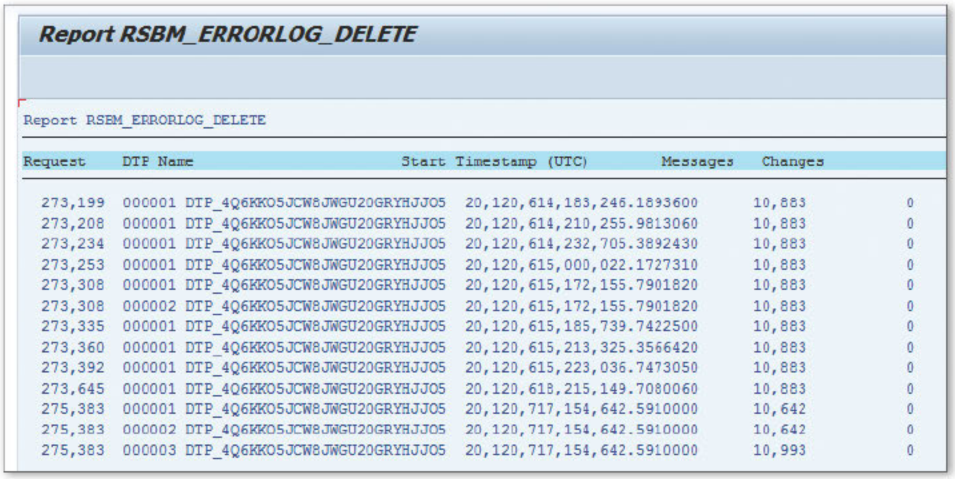


Report RSBM_ERRORLOG_DELETE

Start Date	11/20/2012	to	
DTP	DTP_4Q6KKO5JCW8JW...	to	
but always older than ... days			
✓ Delete Log			

Figure 1 Screen for Entering Selection Criteria for Deletion of Error Logs

The entries are deleted, and the screen shown in Figure 2 is displayed.



Report RSBM_ERRORLOG_DELETE

```
Report RSBM_ERRORLOG_DELETE

Request    DTP Name                                 Start Timestamp (UTC)          Messages   Changes

  273,199  000001 DTP_4Q6KKO5JCW8JWGU20GRYHJJO5  20,120,614,183,246.1893600     10,883          0
  273,208  000001 DTP_4Q6KKO5JCW8JWGU20GRYHJJO5  20,120,614,210,255.9813060     10,883          0
  273,234  000001 DTP_4Q6KKO5JCW8JWGU20GRYHJJO5  20,120,614,232,705.3892430     10,883          0
  273,253  000001 DTP_4Q6KKO5JCW8JWGU20GRYHJJO5  20,120,615,000,022.1727310     10,883          0
  273,308  000001 DTP_4Q6KKO5JCW8JWGU20GRYHJJO5  20,120,615,172,155.7901820     10,883          0
  273,308  000002 DTP_4Q6KKO5JCW8JWGU20GRYHJJO5  20,120,615,172,155.7901820     10,883          0
  273,335  000001 DTP_4Q6KKO5JCW8JWGU20GRYHJJO5  20,120,615,185,739.7422500     10,883          0
  273,360  000001 DTP_4Q6KKO5JCW8JWGU20GRYHJJO5  20,120,615,213,325.3566420     10,883          0
  273,392  000001 DTP_4Q6KKO5JCW8JWGU20GRYHJJO5  20,120,615,223,036.7473050     10,883          0
  273,645  000001 DTP_4Q6KKO5JCW8JWGU20GRYHJJO5  20,120,618,215,149.7080060     10,883          0
  275,383  000001 DTP_4Q6KKO5JCW8JWGU20GRYHJJO5  20,120,717,154,642.5910000     10,642          0
  275,383  000002 DTP_4Q6KKO5JCW8JWGU20GRYHJJO5  20,120,717,154,642.5910000     10,642          0
  275,383  000003 DTP_4Q6KKO5JCW8JWGU20GRYHJJO5  20,120,717,154,642.5910000     10,993          0
```

Figure 2 Output of the Error Log Deletion Process

Tip 53

Copying Queries between InfoCubes That Don't Have Similar Structures

You can use a function module to easily copy queries between InfoCubes that don't have matching structures.

In SAP NetWeaver BW, it's a common requirement to copy queries from one InfoCube to another. This can be easily accomplished using standard SAP NetWeaver BW functionality, but there is a restriction: the target InfoCube (InfoCube for the query copies) must contain all of the InfoObjects of the source InfoCube (InfoCube of the original queries).

However, in some scenarios you may need to copy queries between InfoCubes that don't have matching structures. For instance, you may need to make changes to the name of an InfoCube and that might require you to copy all of the queries to the new InfoCube and then delete the first cube. In other situations, you may need to copy the same query to another MultiProvider, cube, or DSO. In both instances, you need to copy queries from one InfoProvider to another InfoProvider, regardless of matching structures.

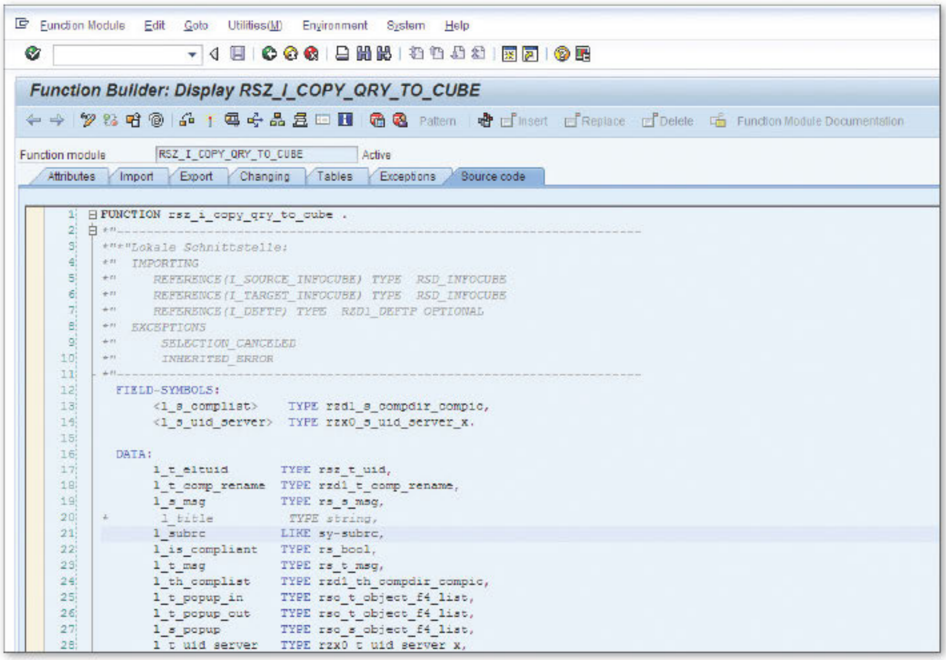
This tip describes a trick to leverage existing standard functionality to copy queries between InfoCubes that do not have matching structures.

✓ And Here's How ...

You can use Transaction RSZC to copy queries between InfoProviders that have the characteristics contained in the source InfoProvider, which should also be contained in the target InfoProvider. Because this tip deals with dissimilar Info-Cube structures, we'll use function module RSZ_I_COPY_QRY_TO_CUBE to copy queries. The main trick here is to use ABAP Debugger to manipulate the behavior of function module RSZ_I_COPY_QRY_TO_CUBE to ignore the check for matching structures for the source and target InfoCubes.

The steps for implementing this are as follows:

1. Log on to the SAP NetWeaver BW environment and run Transaction SE37. In the FUNCTION MODULE field, enter "RSZ_I_COPY_QRY_TO_CUBE" and click DISPLAY.
2. In the resulting code shown in Figure 1, insert a breakpoint in the code at the right spot so that the function module will stop at that point. Click DISPLAY, scroll down, and look for this line in the code: " `IF l_subrc <> 0 OR l_is_compliant = rs_c_false.`"
3. Position the cursor on the line, and set a breakpoint by clicking the STOP icon on the toolbar.



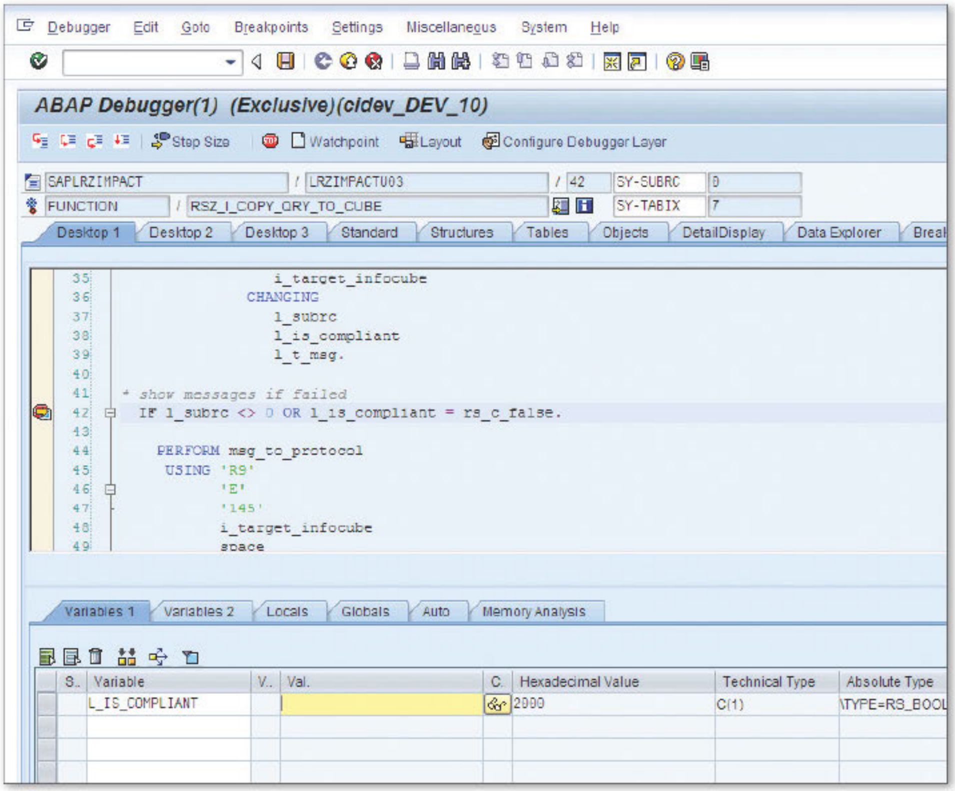
Function module RSZ_I_COPY_QRY_TO_CUBE Active

```
FUNCTION rsz_i_copy_qry_to_cube .
*"----------------------------------------------------------------------
*"*"Lokale Schnittstelle:
*"  IMPORTING
*"     REFERENCE(I_SOURCE_INFOCUBE) TYPE  RSD_INFOCUBE
*"     REFERENCE(I_TARGET_INFOCUBE) TYPE  RSD_INFOCUBE
*"     REFERENCE(I_DEFTP) TYPE  RZD1_DEFTP OPTIONAL
*"  EXCEPTIONS
*"      SELECTION_CANCELED
*"      INHERITED_ERROR
*"----------------------------------------------------------------------
  FIELD-SYMBOLS:
        <l_s_complist>    TYPE rzd1_s_compdir_compic,
        <l_s_uid_server>  TYPE rzx0_s_uid_server_x.

  DATA:
        l_t_eltuid       TYPE rsz_t_uid,
        l_t_comp_rename  TYPE rzd1_t_comp_rename,
        l_s_msg          TYPE rs_s_msg,
*        l_title          TYPE string,
        l_subrc          LIKE sy-subrc,
        l_is_complient   TYPE rs_bool,
        l_t_msg          TYPE rs_t_msg,
        l_th_complist    TYPE rzd1_th_compdir_compic,
        l_t_popup_in     TYPE rso_t_object_f4_list,
        l_t_popup_out    TYPE rso_t_object_f4_list,
        l_s_popup        TYPE rso_s_object_f4_list,
        l_t_uid_server   TYPE rzx0_t_uid_server_x,
```

Figure 1 Setting the Breakpoint in the Code

4. Go back and run the module by pressing F8 or clicking EXECUTE.

5. Type in your source and target cubes, and then press F8. The debugger will stop at the line where you set the breakpoint.
6. Double-click on L_IS_COMPLIANT to make it show up in field names below the code (see Figure 2). L_IS_COMPLIANT in the preceding code is checking for the matching structures between the InfoCubes, and the system will enforce a hard error if the structures are not compliant.
7. Navigate to the VARIABLES 1 tab at the bottom part of the screen. Enter an "X" in the VAL (value) column. Click the CHANGE icon to the right of the line and press F8. By doing this, you are allowing the function module to execute further even if the structures do not match.



ABAP Debugger(1) (Exclusive)(cidev_DEV_10)

SAPLRZIMPACT	/ LRZIMPACTU03	/ 42	SY-SUBRC	0
FUNCTION	/ RSZ_I_COPY_QRY_TO_CUBE		SY-TABIX	7

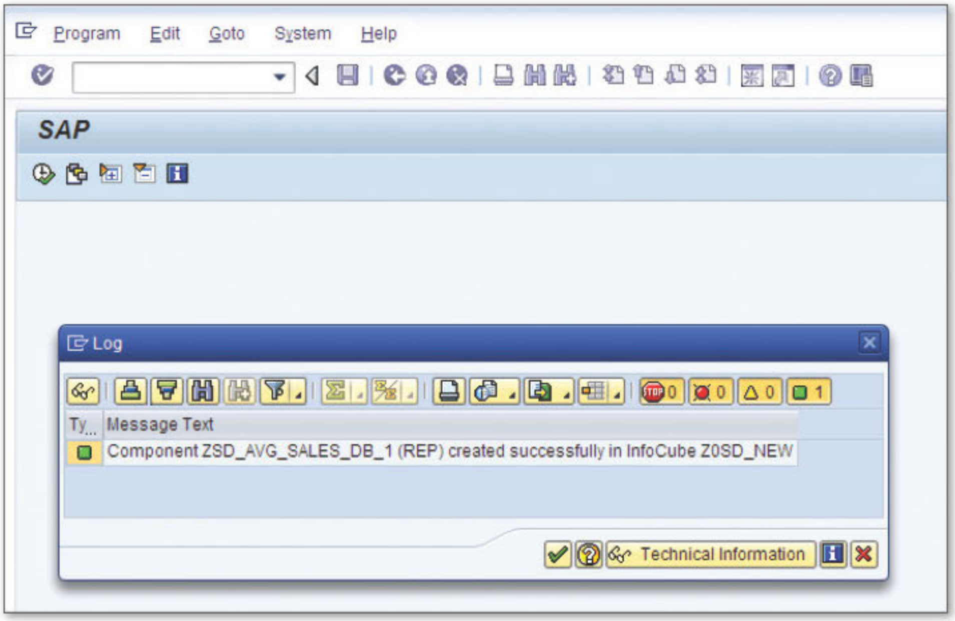
```
                   i_target_infocube
                 CHANGING
                   l_subrc
                   l_is_compliant
                   l_t_msg.

* show messages if failed
  IF l_subrc <> 0 OR l_is_compliant = rs_c_false.

    PERFORM msg_to_protocol
     USING 'R9'
           'E'
           '145'
           i_target_infocube
           space
```

S..	Variable	V..	Val.	C.	Hexadecimal Value	Technical Type	Absolute Type
	L_IS_COMPLIANT				2000	C(1)	\TYPE=RS_BOOL

Figure 2 ABAP Debugger Showing L_IS_COMPLIANT

8. The function module will output a window with a list of queries to choose from. Choose the query you want from the list and click TRANSFER SELECTIONS.

9. Scroll through the log; after all the error messages about the cubes not being identical, there should be one line confirming that your query was copied as shown in Figure 3.



Ty...	Message Text
🟩	Component ZSD_AVG_SALES_DB_1 (REP) created successfully in InfoCube Z0SD_NEW

Figure 3 Log Displaying Message about the Copying Process

Note that this process will not copy BEx workbooks. These must be recreated manually.

Tip 54

Using a Standard Functionality to Perform a Data Dump for an InfoCube

For a simple data dump of an InfoCube (characteristics and key figures), you can generate a .csv file without needing to set up the Open Hub Service.

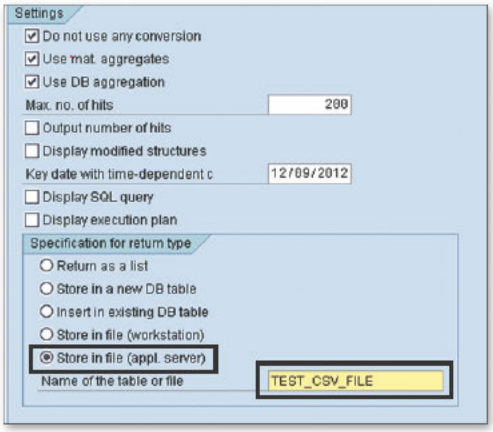
For some scenarios such as troubleshooting data load issues or inconsistency of query results, you may need to get a dump of a large volume of cube data in a *.csv* file so that you can perform further analysis on the data using Microsoft Excel. You can use the output of the BEx query on the cube for this purpose, but there is a restriction on the number of records you can display on the BEx query result. Performance of the query is a major concern as well.

This tip describes a simple method to get a data dump of the InfoCube without executing a BEx query. This method will overcome the performance concerns as well the limitation on the number of records that can be downloaded.

✓ And Here's How ...

If a simple data dump of the cube is required, SAP NetWeaver BW provides standard functionality instead of configuring the Open Hub Service. Follow these steps:

1. Use Transaction LISTCUBE (or in Transaction RSA1, right-click on Cube • Display Data).
2. In the Specifications for return type area, select the Store in file (appl. server) option (see Figure 1). Also specify a file name (e.g., "TEST_CSV_FILE").



Settings

- [✓] Do not use any conversion
- [✓] Use mat. aggregates
- [✓] Use DB aggregation

Max. no. of hits	200

- ☐ Output number of hits
- ☐ Display modified structures

Key date with time-dependent c	12/09/2012

- ☐ Display SQL query
- ☐ Display execution plan

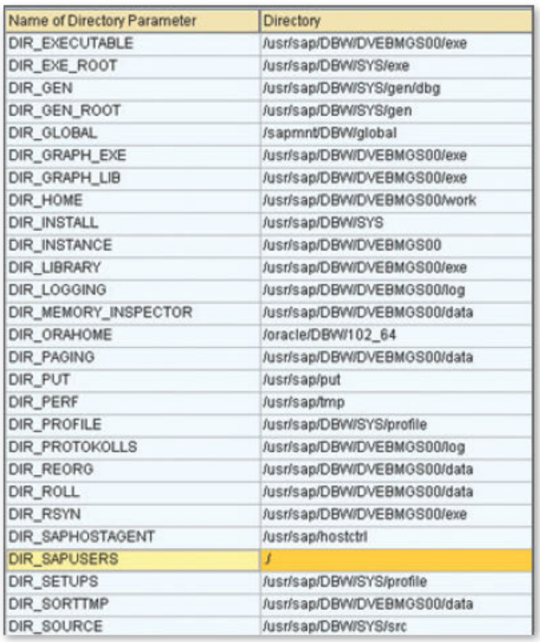
Specification for return type

- ○ Return as a list
- ○ Store in a new DB table
- ○ Insert in existing DB table
- ○ Store in file (workstation)
- [✓ ● Store in file (appl. server)]

Name of the table or file	TEST_CSV_FILE

« *Figure 1 Selection Options for Transaction LISTCUBE*

3. Click on Execute or press F8. To view/download the *.csv* file, use Transaction AL11. The file is typically stored under the directory DIR_SAPUSERS as shown in Figure 2.



Name of Directory Parameter	Directory
DIR_EXECUTABLE	/usr/sap/DBW/DVEBMGS00/exe
DIR_EXE_ROOT	/usr/sap/DBW/SYS/exe
DIR_GEN	/usr/sap/DBW/SYS/gen/dbg
DIR_GEN_ROOT	/usr/sap/DBW/SYS/gen
DIR_GLOBAL	/sapmnt/DBW/global
DIR_GRAPH_EXE	/usr/sap/DBW/DVEBMGS00/exe
DIR_GRAPH_LIB	/usr/sap/DBW/DVEBMGS00/exe
DIR_HOME	/usr/sap/DBW/DVEBMGS00/work
DIR_INSTALL	/usr/sap/DBW/SYS
DIR_INSTANCE	/usr/sap/DBW/DVEBMGS00
DIR_LIBRARY	/usr/sap/DBW/DVEBMGS00/exe
DIR_LOGGING	/usr/sap/DBW/DVEBMGS00/log
DIR_MEMORY_INSPECTOR	/usr/sap/DBW/DVEBMGS00/data
DIR_ORAHOME	/oracle/DBW/102_64
DIR_PAGING	/usr/sap/DBW/DVEBMGS00/data
DIR_PUT	/usr/sap/put
DIR_PERF	/usr/sap/tmp
DIR_PROFILE	/usr/sap/DBW/SYS/profile
DIR_PROTOKOLLS	/usr/sap/DBW/DVEBMGS00/log
DIR_REORG	/usr/sap/DBW/DVEBMGS00/data
DIR_ROLL	/usr/sap/DBW/DVEBMGS00/data
DIR_RSYN	/usr/sap/DBW/DVEBMGS00/exe
DIR_SAPHOSTAGENT	/usr/sap/hostctrl
DIR_SAPUSERS	./
DIR_SETUPS	/usr/sap/DBW/SYS/profile
DIR_SORTTMP	/usr/sap/DBW/DVEBMGS00/data
DIR_SOURCE	/usr/sap/DBW/SYS/src

« *Figure 2 Directory Options for Saving the CSV File into the Application Server*

4. Double-click on the line to see the file TEST_CSV_FILE.
5. To download the file, click on List • Save/Send • File. Here you can specify how you want the file to be saved.

Tip 55

Creating Monitor Entries to Display Custom Messages during Data Load

You can create monitor entries with custom status messages to provide the user with essential information when transaction data is loaded to an InfoProvider.

In an SAP NetWeaver BW system, data load transformation error messages are recorded and tracked in the monitor table and displayed in the monitor screen of the related DTP. In some scenarios, you may actually want to manipulate the application of business rules for certain criteria, or you may not want to process the data for certain criteria during the transaction data load. In addition, you may want to provide information or custom status messages regarding the data load process.

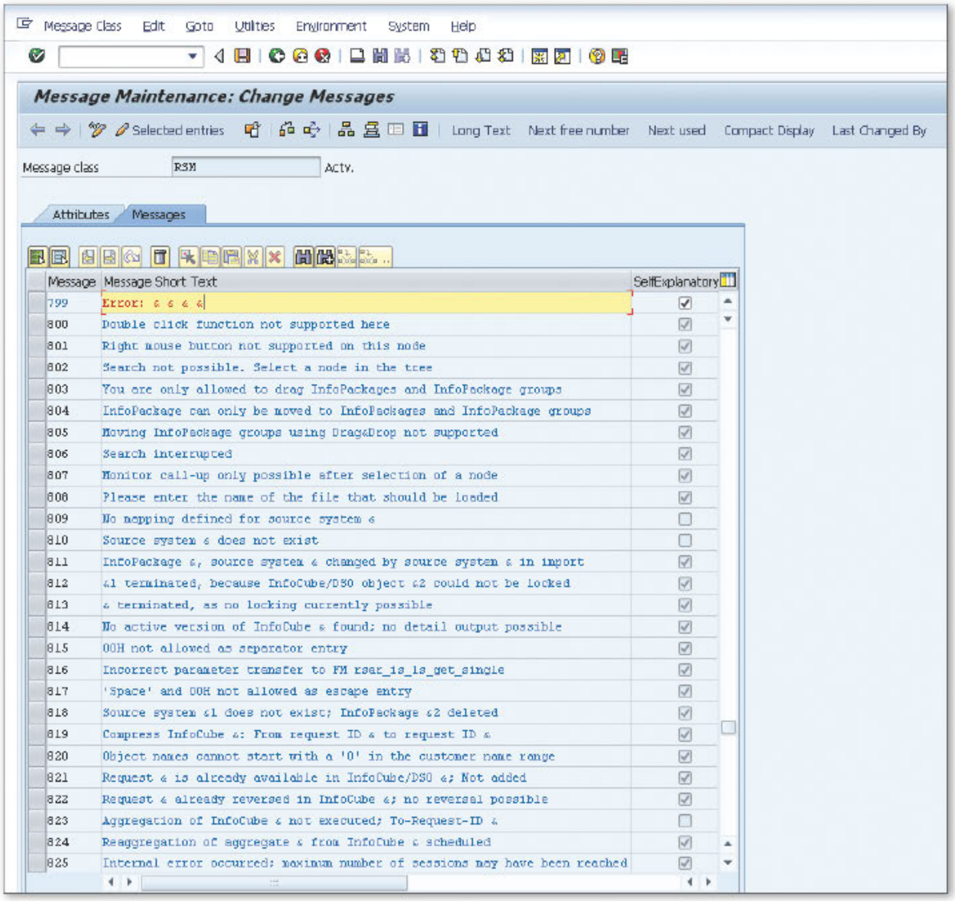
You can accomplish this by applying specific logic in the transformation to manipulate the data-loading process and create monitor entries that can be displayed in the DTP maintenance screen. These monitor entries provide valuable information to the data load monitoring person.

✓ And Here's How ...

Here, you'll create monitor entries when the data is loaded to an InfoProvider that are then saved for tracking purposes. Follow these steps:

1. Go to Transaction SE91, enter "RSM" in the Message Class field, and click on the Change button.

2. On the resulting popup, set the logon and original languages, and select the MAINT. IN LOGON LANG. button in the next screen.
3. In the following screen, you can choose an existing message you wanted to display in the DTP monitor screen, or you can create your own (see Figure 1).



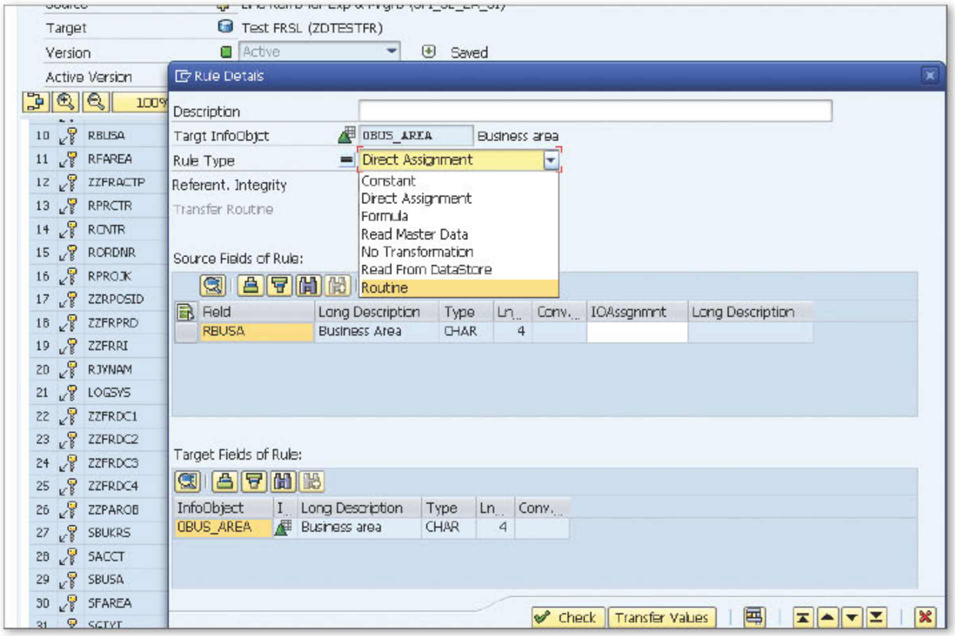
Message class RSM Actv.

Message	Message Short Text	SelfExplanatory
799	Error: & & & &	✓
800	Double click function not supported here	✓
801	Right mouse button not supported on this node	✓
802	Search not possible. Select a node in the tree	✓
803	You are only allowed to drag InfoPackages and InfoPackage groups	✓
804	InfoPackage can only be moved to InfoPackages and InfoPackage groups	✓
805	Moving InfoPackage groups using Drag&Drop not supported	✓
806	Search interrupted	✓
807	Monitor call-up only possible after selection of a node	✓
808	Please enter the name of the file that should be loaded	✓
809	No mapping defined for source system &	☐
810	Source system & does not exist	☐
811	InfoPackage &, source system & changed by source system & in import	✓
812	&1 terminated, because InfoCube/DSO object &2 could not be locked	✓
813	& terminated, as no locking currently possible	✓
814	No active version of InfoCube & found; no detail output possible	✓
815	00H not allowed as seperator entry	✓
816	Incorrect parameter transfer to FM rsar_is_is_get_single	✓
817	'Space' and 00H not allowed as escape entry	✓
818	Source system &1 does not exist; InfoPackage &2 deleted	✓
819	Compress InfoCube &: From request ID & to request ID &	✓
820	Object names cannot start with a '0' in the customer name range	✓
821	Request & is already available in InfoCube/DSO &; Not added	✓
822	Request & already reversed in InfoCube &; no reversal possible	✓
823	Aggregation of InfoCube & not executed; To-Request-ID &	☐
824	Reaggregation of aggregate & from InfoCube & scheduled	✓
825	Internal error occurred; maximum number of sessions may have been reached	✓

Figure 1 Create Error Message to Be Displayed in the DTP Monitor Screen

4. When you've chosen the correct message, execute Transaction RSA1 in the SAP NetWeaver BW system.
5. Select the transformation for which you want to create the monitor entries during data load.
6. In the MAINTENANCE screen of the transformation, select the characteristic for which you want to create messages.

7. In the Rule Details screen, select Routine (see Figure 2).



Target: Test FRSL (ZDTESTFR)
Version: Active — Saved
Active Version

10	RBUSA
11	RFAREA
12	ZZFRACTP
13	RPRCTR
14	RCNTR
15	RORDNR
16	RPROJK
17	ZZRPOSID
18	ZZFRPRD
19	ZZFRRI
20	RJVNAM
21	LOGSYS
22	ZZFRDC1
23	ZZFRDC2
24	ZZFRDC3
25	ZZFRDC4
26	ZZPAROB
27	SBUKRS
28	SACCT
29	SBUSA
30	SFAREA

Description:
Targt InfoObjct: 0BUS_AREA Business area
Rule Type: Direct Assignment
Referent. Integrity
Transfer Routine

Source Fields of Rule:

Field	Long Description	Type	Ln..	Conv..	IOAssgnmnt	Long Description
RBUSA	Business Area	CHAR	4			

Target Fields of Rule:

InfoObject	I..	Long Description	Type	Ln..	Conv..
0BUS_AREA		Business area	CHAR	4	

Figure 2 Transformation Rule

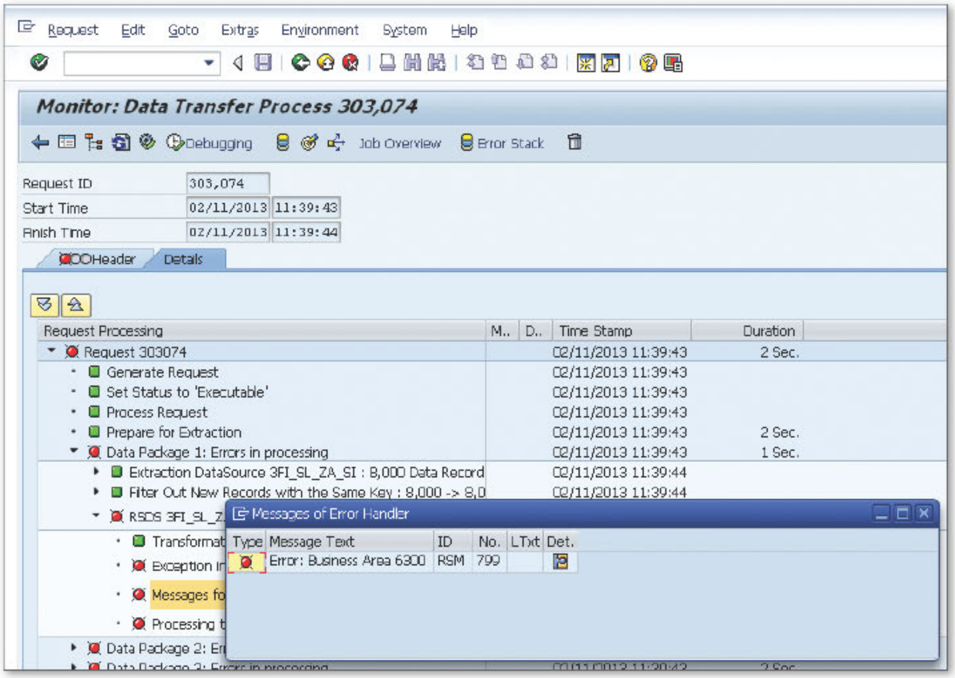
8. Input the following code, which will raise an exception if the business area = 6300 and also create an error message in the monitor for any records that meet this criteria:

```
IF SOURCE_FIELDS-RBUSA = '6300'.
       MONITOR_REC-MSGID = 'RSM'.
       MONITOR_REC-MSGTY = 'E'.
       MONITOR_REC-MSGNO = '799'.
       MONITOR_REC-MSGV1 = 'Business Area'.
       MONITOR_REC-MSGV2 = SOURCE_FIELDS-RBUSA.
 APPEND MONITOR_REC to MONITOR.
 RAISE exception type CX_RSROUT_SKIP_RECORD.
  ENDIF.
```

The exception that is raised in this logic will create the monitor entry, skip that particular record, and start processing the next record. You can also raise an exception to stop the data load completely.

9. Click on the Save button and activate the transformation.

10. Create the DTP and execute it. The DTP maintenance screen will now display the message that you selected earlier.
11. If you further drill down by right-clicking on the message and choosing Display Messages of Error Handler, the system will show the message you see in Figure 3.



Monitor: Data Transfer Process 303,074

Request ID	303,074	
Start Time	02/11/2013	11:39:43
Finish Time	02/11/2013	11:39:44

Request Processing	M..	D..	Time Stamp	Duration
Request 303074			02/11/2013 11:39:43	2 Sec.
Generate Request			02/11/2013 11:39:43	
Set Status to 'Executable'			02/11/2013 11:39:43	
Process Request			02/11/2013 11:39:43	
Prepare for Extraction			02/11/2013 11:39:43	2 Sec.
Data Package 1: Errors in processing			02/11/2013 11:39:43	1 Sec.
Extraction DataSource 3FI_SL_ZA_SI : 8,000 Data Record			02/11/2013 11:39:44	
Filter Out New Records with the Same Key : 8,000 -> 8,0			02/11/2013 11:39:44	
RSDS 3FI_SL_Z				
Transformat				
Exception in				
Messages fo				
Processing t				
Data Package 2: Er				
Data Package 3: Errors in processing			02/11/2013 11:39:43	2 Sec.

Messages of Error Handler

Type	Message Text	ID	No.	LTxt	Det.
🟥	Error: Business Area 6300	RSM	799		

Figure 3 Displaying a Custom Message

Tip 56

Improving the Load Time of Standard DSO Data Activation

You can adjust and maintain a few settings that will greatly improve the DSO data load performance in your SAP NetWeaver BW system.

In large SAP NetWeaver BW implementations, large data volumes often create severe load performance issues. DSO data activation takes a major share of the load time. This activation process for DSO is supposed to work with high level or parallelism and with better performance; however, it's common to see the activation for even a few data records taking a long time. Usually, the reason for the inferior performance is due to some settings that influence the DSO activation.

This tip describes some of the settings that can be maintained for improving the performance of the activation process of the DSO data loads.

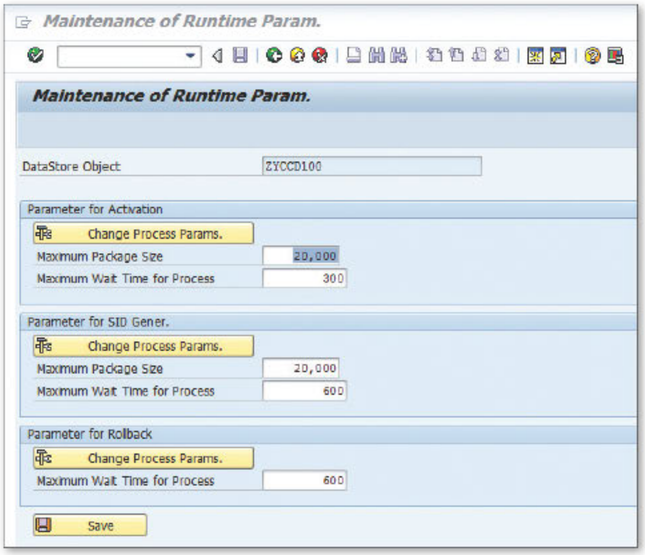
✓ And Here's How ...

To adjust your settings for data loading, follow these steps:

1. Log in to the SAP NetWeaver BW server and go to Transaction RSODSO_SETTINGS.
2. Select the DataStore Specific radio button, and enter the DataStore Object for which you want performance improvement in activation. Alternatively, you may access this transaction in the manage DSO view via Goto • Customizing DataStore.
3. In the resulting screen, you can make a Cross-DataStore setting, which is a global setting that applies to all of the DSOs, or you can make DataStore Specific settings that are specific to a DSO. For this tip, choose the DataStore

SPECIFIC setting and enter the name of the DSO "ZYCCD100" for which to maintain the setting.

4. Click on the CHANGE button to maintain settings for the DSO.
5. A new window appears (see Figure 1) where you can maintain settings specific to the DSO. You can maintain three types of parameters on this screen, PARAMETER FOR ACTIVATION, PARAMETER FOR SID GENER., and PARAMETER FOR ROLLBACK.



DataStore Object ZYCCD100

Parameter for Activation

Maximum Package Size	20,000
Maximum Wait Time for Process	300

Parameter for SID Gener.

Maximum Package Size	20,000
Maximum Wait Time for Process	600

Parameter for Rollback

Maximum Wait Time for Process	600

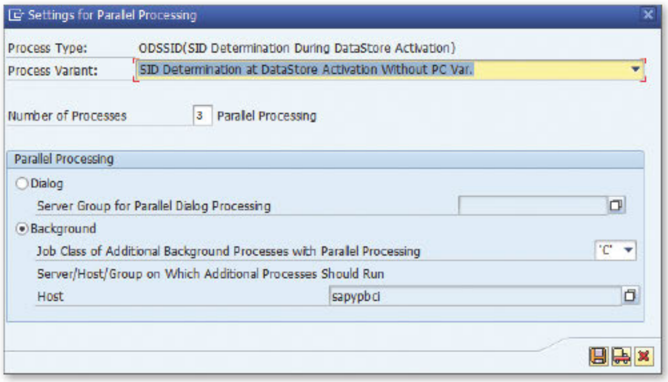
Figure 1 Maintenance of Runtime Parameters for DSO ZYCCD100

To improve the activation process for the DSO, follow these suggestions:

- The default number for the MAXIMUM PACKAGE SIZE is 20,000. This means that the total number of records will be split into smaller packages based on the number that is entered here. This number can be changed depending on your individual system and also the total number of records that need to be loaded. For a larger load, the MAXIMUM PACKAGE SIZE can be a higher number, or this number can be lower for a smaller load.
- It's common for this parameter to have values between 10,000 and 30,000. The MAXIMUM WAIT TIME FOR PROCESS option in this window defines how long the job is allowed to run and activate the data. If the activation job takes longer than the wait time defined here, then SAP NetWeaver BW will abort the data load job. Again, this setting can also be changed based on the number

of data loads in the system and how long the system can afford to wait on the activation process.

6. Click on the Change Process Params. option in the Parameter for Activation section, and a new popup window (see Figure 2) opens that allows some key settings for parallel processing of the activation job. This setting is also specific to your system and will determine how many parallel processes will be dedicated for this activation job. For mass data loads with an expected number of millions of records, you may set this parameter to a higher value, for example, 5 or 6, and for smaller loads you can set this parameter to a smaller value of 2 or 3.3.


Settings for Parallel Processing

Process Type: ODSSID(SID Determination During DataStore Activation)
Process Variant: SID Determination at DataStore Activation Without PC Var.

Number of Processes 3 Parallel Processing

Parallel Processing
Dialog
Server Group for Parallel Dialog Processing
[✓ Background]
Job Class of Additional Background Processes with Parallel Processing 'C'
Server/Host/Group on Which Additional Processes Should Run
Host sapypbci

Figure 2 Settings for Parallel Processing

Similarly from Figure 1, you may change the settings for Parameter for SID Gener. and Parameter for Rollback that can influence the SID generation and rollback processes.

Also, you may make all of the preceding settings globally for all of the DSOs by choosing the Cross-DataStore option shown earlier in Figure 1. In this case, all of the settings described can be made globally and be applied to all DSOs.

Tip 57

Reversing Data Flow Migration from 7.x to 3.x

You can reverse the migration of data flow from SAP NetWeaver BW version 3.x to 7.x to overcome the impact of migration errors.

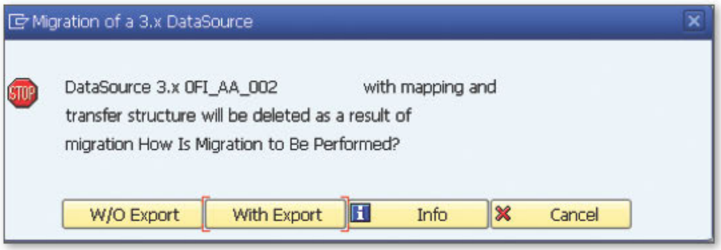
Typically after upgrading to SAP NetWeaver BW 7.x from 3.5, users want to convert all of the data flows to everything to 7.x data flows. The upgrade from 3.x to 7.x can be done using a migration wizard; during this process, the 3.x objects such as update rules, InfoSources, transfer rules, and DataSources are converted into 7.x data flows that include transformations, DTPs, and 7.x InfoSources.

However, in some cases, this migration to the 7.x data flow results in errors for specific data flows. If you are not able to resolve the issue in a timely manner to restart the data loads, then you may be required to reverse the migration. The major impact of this issue is that the data loads in the SAP NetWeaver BW system cannot be resumed until this issue is resolved.

This tip will describe a method to reverse the data flow migration from version 7.x to 3.x to enable the data loads to resume and provide more time to troubleshoot the migration error.

✓ And Here's How ...

Typically, when performing a migration of a DataSource, you would go to the relevant transfer rules and right-click on the DataSource. Choose the option MIGRATE and the screen shown in Figure 1 will be displayed.


Migration of a 3.x DataSource

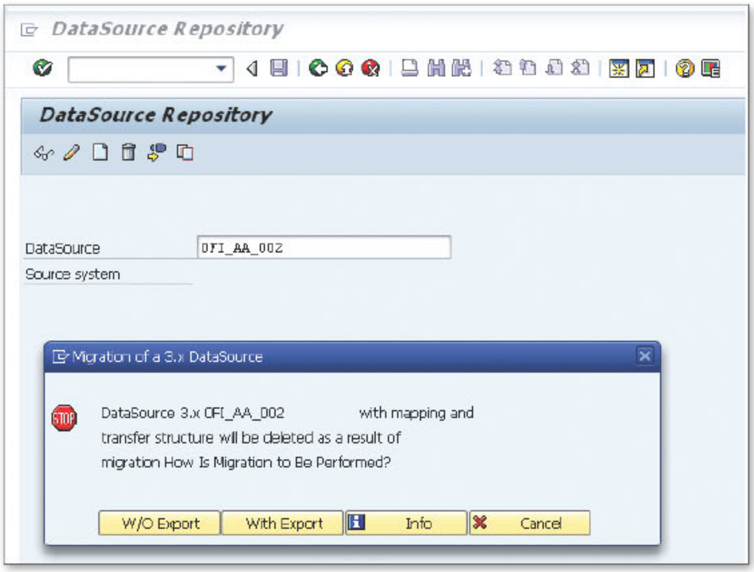
DataSource 3.x 0FI_AA_002 with mapping and transfer structure will be deleted as a result of migration How Is Migration to Be Performed?

Figure 1 Prompt during the Migration of a 3.x DataSource

You can only perform a reverse migration on a DataSource, which is the purpose of this tip, if the initial migration was carried out WITH EXPORT.

To reverse the migration, follow these steps:

1. Log in to the SAP NetWeaver BW system. Use Transaction RSDS.
2. Specify the DataSource and source system, and then press Enter. The screen shown in Figure 2 appears.


DataSource Repository

DataSource Repository

DataSource 0FI_AA_002
Source system

Migration of a 3.x DataSource

DataSource 3.x 0FI_AA_002 with mapping and transfer structure will be deleted as a result of migration How Is Migration to Be Performed?

Figure 2 Message to Confirm Recovery of the DataSource

3. Click on YES. The DataSource has now reverted back to the old version of 3.x.

The system reproduces the 3.x DataSource (R3TR ISFS), mapping (R3TR ISMP), and transfer structure (R3TR ISTS) objects with their premigration status.

Tip 58

Enabling Email Alerting for Failed Process Chain Steps

You can monitor your process chains remotely by generating simple error messages that will be sent automatically by email or SMS when a step fails.

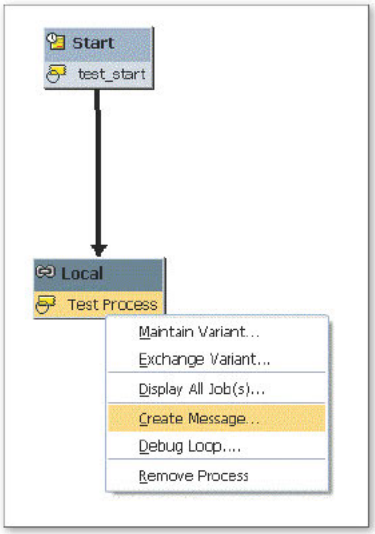
Process chain monitoring is an important activity in SAP NetWeaver BW, mainly during support. You'll need to know the different errors that may occur during the process chain execution and know the necessary actions to take to correct process chains in order to complete the data loads in a timely manner. However, you can't be glued to your screen, waiting to resolve errors at all times (no matter how much you love your job!).

This tip describes a method to configure the system to generate process chain error messages and send them to you via email or even an SMS text message when a process chain fails.

✓ And Here's How ...

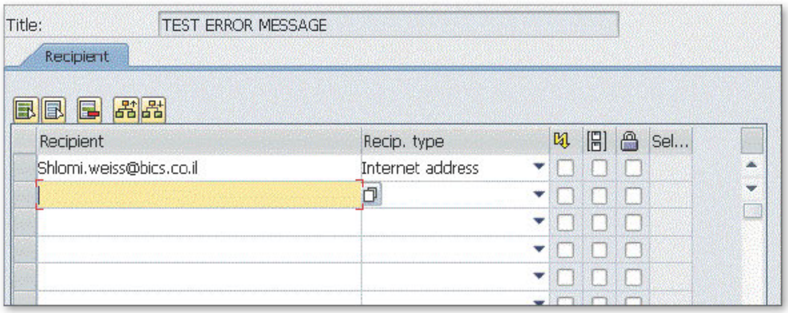
To configure the process chains for generating error messages, you need to add a simple error message to a local process chain process type (it executes another process chain) with a simple text and a recipient list. Follow these steps:

1. Log in to the SAP NetWeaver BW system and execute Transaction RSPC. Right-click on the LOCAL PROCESS CHAIN process type, and choose CREATE MESSAGE as shown in Figure 1.



« *Figure 1 Creating a Message on the Local Process Type*

2. This will display another popup window where you can choose the type of message for which you'd like to receive a message; that is, Successful, Errors, or Always. Select Errors and click Continue.
3. The next screen allows you to create the new message. Enter the description. This screen has two options: Edit Document and Maintain Recipient List.
4. Click on the Edit Document button to define the text that will appear in the message body. That will bring you to a simple editor where you can add your text.
5. Create your recipient list by clicking on the Maintain Recipient List button. This opens another screen, as shown in Figure 2, where you need to create your recipient list.

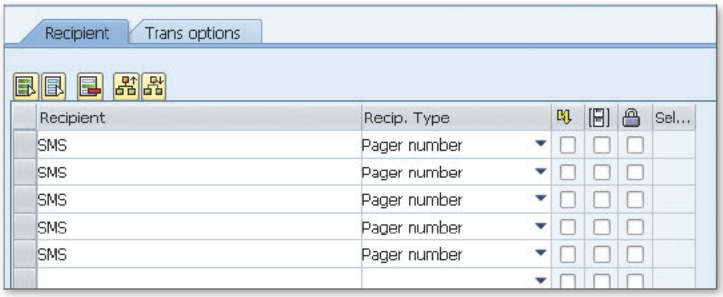

Title: TEST ERROR MESSAGE

Recipient

Recipient	Recip. type				Sel...
Shlomi.weiss@bics.co.il	Internet address	☐	☐	☐	
		☐	☐	☐	
		☐	☐	☐	
		☐	☐	☐	
		☐	☐	☐	

Figure 2 Adding Email ID Recipients to the List

If you want to send an SMS text message when your process fails, you need your Basis team to do some additional configurations. Then, in your recipient list, choose the SMS number and VIA PAGER on the RECIPIENT TYPE as shown in Figure 3.



Recipient	Recip. Type				Sel...
SMS	Pager number	☐	☐	☐	
SMS	Pager number	☐	☐	☐	
SMS	Pager number	☐	☐	☐	
SMS	Pager number	☐	☐	☐	
SMS	Pager number	☐	☐	☐	
		☐	☐	☐	

Figure 3 Adding SMS Numbers for Receiving Text Messages

With these steps, you've created a local process chain for generating error messages. Now you can include this in any process chains after any step for which you need to generate the message.

Tip 59

Scheduling Process Chains to Run in Loops

You can run process chains in loops using events as triggers to schedule a continuous update of data in SAP NetWeaver BW.

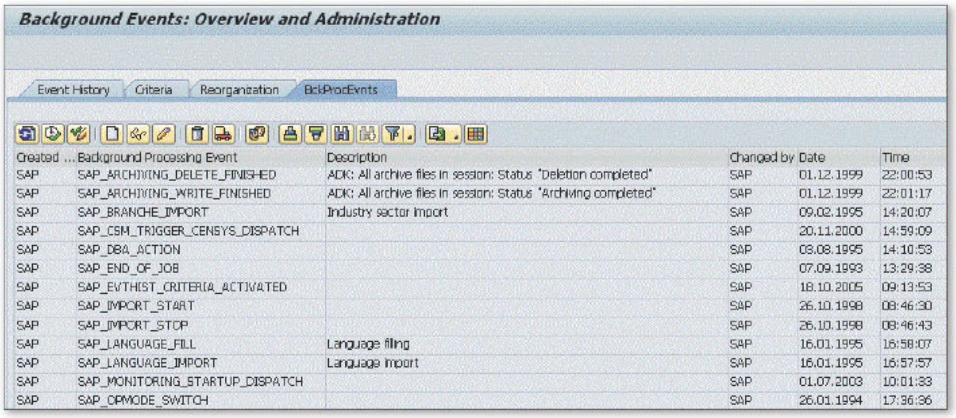
You can use process chains in SAP NetWeaver BW to schedule data loads. Process chains are basically a sequence of processes that wait in the background for an event. Some of these processes trigger a separate event that can start other processes in turn. In some cases, you'll want your process chain to run in a loop, which means that when it ends, it will start over automatically.

In some scenarios, you may want your data to be updated all the time instead of scheduling based on specific events. This tip describes a simple method to configure process chains to run in loops using events as triggers.

✓ And Here's How ...

To schedule the process chains, follow these steps:

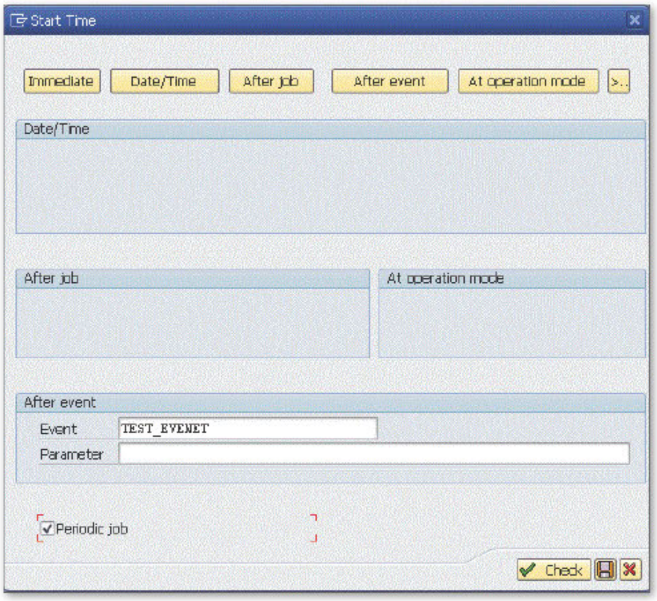
1. Log in to SAP NetWeaver BW and execute Transaction SM64. Create an event that can be used to trigger the process chain as shown in Figure 1.



Created ...	Background Processing Event	Description	Changed by	Date	Time
SAP	SAP_ARCHIVING_DELETE_FINISHED	ADK: All archive files in session: Status "Deletion completed"	SAP	01.12.1999	22:00:53
SAP	SAP_ARCHIVING_WRITE_FINISHED	ADK: All archive files in session: Status "Archiving completed"	SAP	01.12.1999	22:01:17
SAP	SAP_BRANCHE_IMPORT	Industry sector import	SAP	09.02.1995	14:20:07
SAP	SAP_CSM_TRIGGER_CENSYS_DISPATCH		SAP	20.11.2000	14:59:09
SAP	SAP_DBA_ACTION		SAP	03.08.1995	14:10:53
SAP	SAP_END_OF_JOB		SAP	07.09.1993	13:29:38
SAP	SAP_EVTHIST_CRITERIA_ACTIVATED		SAP	18.10.2005	09:13:53
SAP	SAP_IMPORT_START		SAP	26.10.1998	08:46:30
SAP	SAP_IMPORT_STOP		SAP	26.10.1998	08:46:43
SAP	SAP_LANGUAGE_FILL	Language filing	SAP	16.01.1995	16:58:07
SAP	SAP_LANGUAGE_IMPORT	Language import	SAP	16.01.1995	16:57:57
SAP	SAP_MONITORING_STARTUP_DISPATCH		SAP	01.07.2003	10:01:33
SAP	SAP_OPMODE_SWITCH		SAP	26.01.1994	17:36:36

Figure 1 Transaction SM64: Create Event as Trigger

2. Click on the CREATE icon in Figure 1 to display a popup entitled EVENT DEFINITION. Enter the name and description, and click the SAVE button for the event definition.
3. In your process chain, start the process type you'll need to start after the event. In the AFTER EVENT area, choose the new event that you've created, and make it periodic by clicking the PERIODIC JOB box as shown in Figure 2.



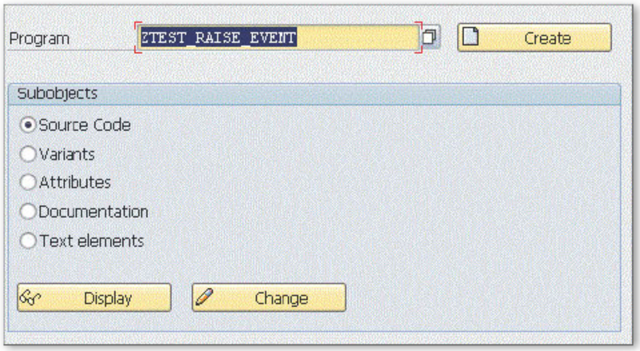
Event TEST_EVENET
Parameter

[✓] Periodic job

Figure 2 Defining the Start Condition for the Event

4. Create a simple program using Transaction SE38 that will raise the event and fire up the process. To do that, go to Transaction SE38 and create the program.

   In the example shown in Figure 3, we are creating a new program called ZTEST_RAISE_EVENT.


Program: ZTEST_RAISE_EVENT

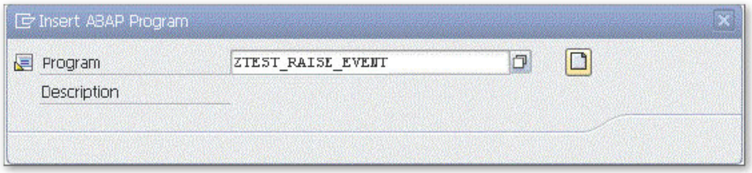
Subobjects
[✓ Source Code]
Variants
Attributes
Documentation
Text elements

Figure 3 Creating a Program with Transaction SE38

5. Click on CREATE, and then in the next screen, fill in the fields for the ABAP attributes. Click SAVE. That will bring you to the ABAP Editor where you can enter the source code for the program as shown in Figure 4. After adding the code, activate the program.

```
CALL FUNCTION 'BP_EVENT_RAISE'
  EXPORTING
    EVENTID                        = 'TEST_EVENET'
*   EVENTPARM                      = ' '
*   TARGET_INSTANCE                = ' '
 EXCEPTIONS
   BAD_EVENTID                    = 1
   EVENTID_DOES_NOT_EXIST         = 2
   EVENTID_MISSING                = 3
   RAISE_FAILED                   = 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.
```

6. Add an ABAP PROGRAM process type to your process chain and connect it to the program that you've just created. See Figure 4 for a reference, and then click CREATE.



Program: ZTEST_RAISE_EVENT

Figure 4 Inserting an ABAP Program

7. The PROCESS MAINTENANCE: ABAP PROGRAM screen is displayed. Enter the name of the ABAP program that you created in the earlier step. Save your changes, and now this variant that you created can be used in process chains.

Your final process chain should look like what's shown in Figure 5.

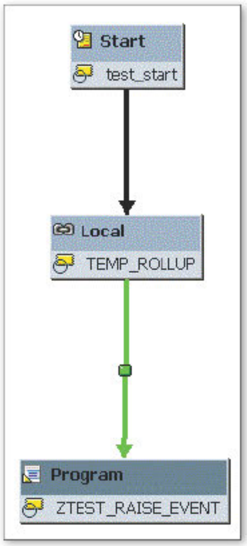


Figure 5 Process Chain

The TEMP_ROLLUP local process that you see in Figure 5 is just an example; whatever you will decide to put between the start process type and the ABAP routine will run inside the loop. Now, just activate your process chain.

Tip 60

Correlating BEx Query Elements in a Transport to a Common Query

With the help of a couple of transaction codes, you can simplify the process of validating query elements to the parent query.

After you've created a transport from SAP NetWeaver BW development system to your SAP NetWeaver BW QA/production systems, you need to verify that you transported only the relevant BEx queries. To do this, you'll use Transaction SE09 and open the query elements in the transport, but you can't discern the elements that are relevant to your query. This tip prevents the user from having to manually verify every object against its parent query.

✓ And Here's How ...

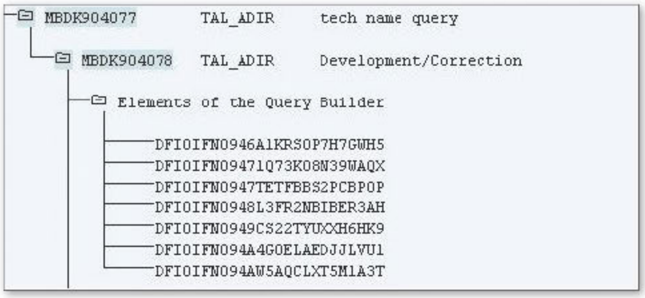
Note that the SAP NetWeaver BW system has three names for every query:

1. The query description
2. The technical name of the SAP NetWeaver BW query
3. The automatic numbering that the system has assigned

Also, all SAP NetWeaver BW report elements in the query receive an automated number.

Follow these steps to check the transported query elements:

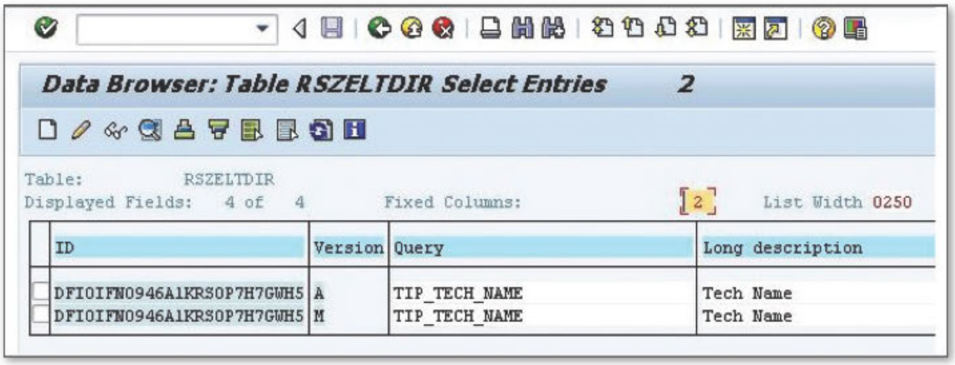
1. Log in to your SAP NetWeaver BW system and run Transaction SE09. As shown in Figure 1, you can see the query elements in the transport.



```
MBDK904077        TAL_ADIR     tech name query
  MBDK904078      TAL_ADIR     Development/Correction
    Elements of the Query Builder
      DFI0IFN0946A1KRS0P7H7GWH5
      DFI0IFN09471Q73K08N39WAQX
      DFI0IFN0947TETFBBS2PCBP0P
      DFI0IFN0948L3FR2NBIBER3AH
      DFI0IFN0949CS22TYUXXH6HK9
      DFI0IFN094A4G0ELAEDJJLVU1
      DFI0IFN094AW5AQCLXT5M1A3T
```

Figure 1 Query Elements

2. Run Transaction SE11 and specify table RSZELTDIR.
3. In the DATA BROWSER screen, enter one of the ID numbers that is given in Transaction SE09 in the ID field. In this example, we enter "0946A1KRS0P7H7GWH5" in the ID field.
4. After you've entered the ID, you will get the result shown in Figure 2. The ID you entered returns the name of the respective query
5. Now you can enter other IDs from Figure 1 to understand which query IDs belong to the respective query



Data Browser: Table RSZELTDIR Select Entries 2

```
Table:            RSZELTDIR
Displayed Fields:   4 of   4        Fixed Columns:           2     List Width 0250
```

ID	Version	Query	Long description
DFI0IFN0946A1KRS0P7H7GWH5	A	TIP_TECH_NAME	Tech Name
DFI0IFN0946A1KRS0P7H7GWH5	M	TIP_TECH_NAME	Tech Name

Figure 2 Results Screen for Query Element ID and Query Name

Tip 61

Merging Two Queries into One BEx Analyzer Workbook

You can quickly merge two queries into one to display joined data using a BEx Analyzer workbook.

Often, users may need to use data from different InfoProviders. Typically, you'll create a MultiProvider to merge multiple queries together. Alternatively, it's faster to put two or more queries into one BEx workbook and create a separate tab to display the joined data. Similar to how you can merge multiple queries together in SAP BusinessObjects Web Intelligence, the same process can be achieved through BEx Analyzer. This shortcut is useful in certain situations where creating or modifying the data model isn't a realistic possibility.

✓ And Here's How ...

Follow these steps to easily merge two queries into one using the BEx Analyzer workbook (note that the BEx queries should already exist):

1. Open one of the queries in the BEx Analyzer, and save it as a workbook.
2. Create two additional tabs in the workbook and name them as "Query1" and "Query2".
3. To add design items to the QUERY2 tab, click the BEX ANALYZER button, go to the DESIGN toolbar, and insert the analysis grid by clicking the icon (, see Figure 1).

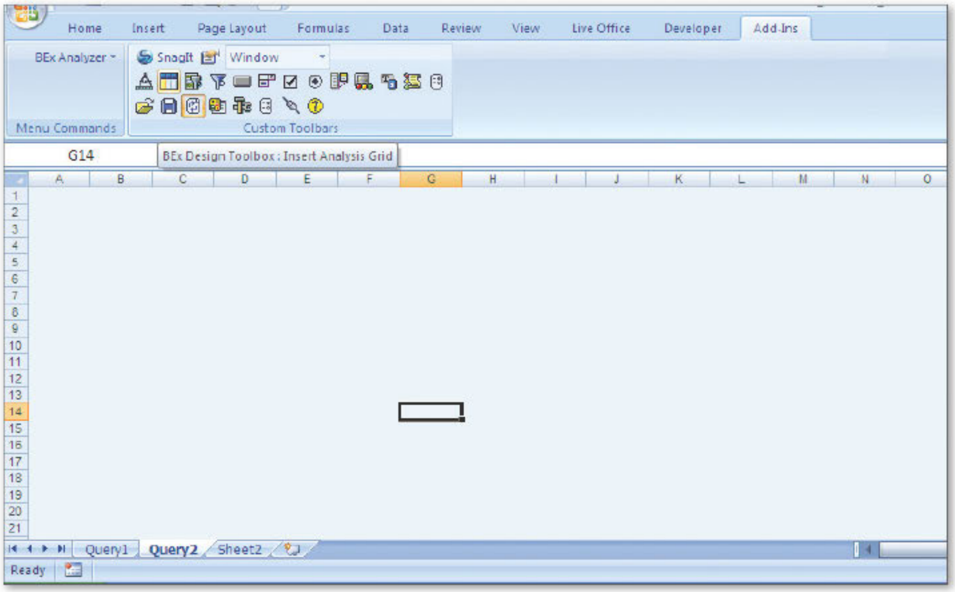
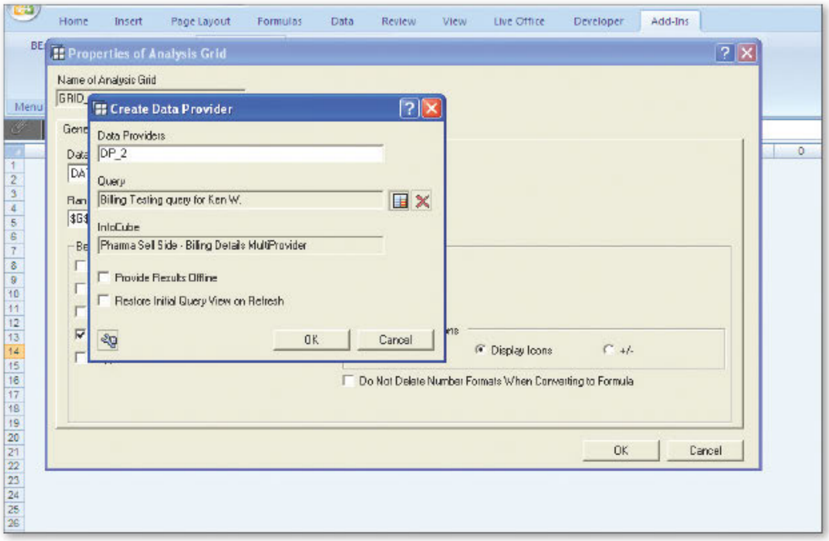


Figure 1 Add Design Items to the Query2 Tab

4. Click on the Properties dialog box, change the Data Provider field entry to reference Query2 (click on ▢ to find the query), and click on the OK button (see Figure 2).


Properties of Analysis Grid

Name of Analysis Grid: GRID_

Create Data Provider

Data Providers: DP_2

Query: Billing Testing query for Ken W.

InfoCube: Pharma Sell Side - Billing Details MultiProvider

☐ Provide Results Offline

☐ Restore Initial Query View on Refresh

OK Cancel

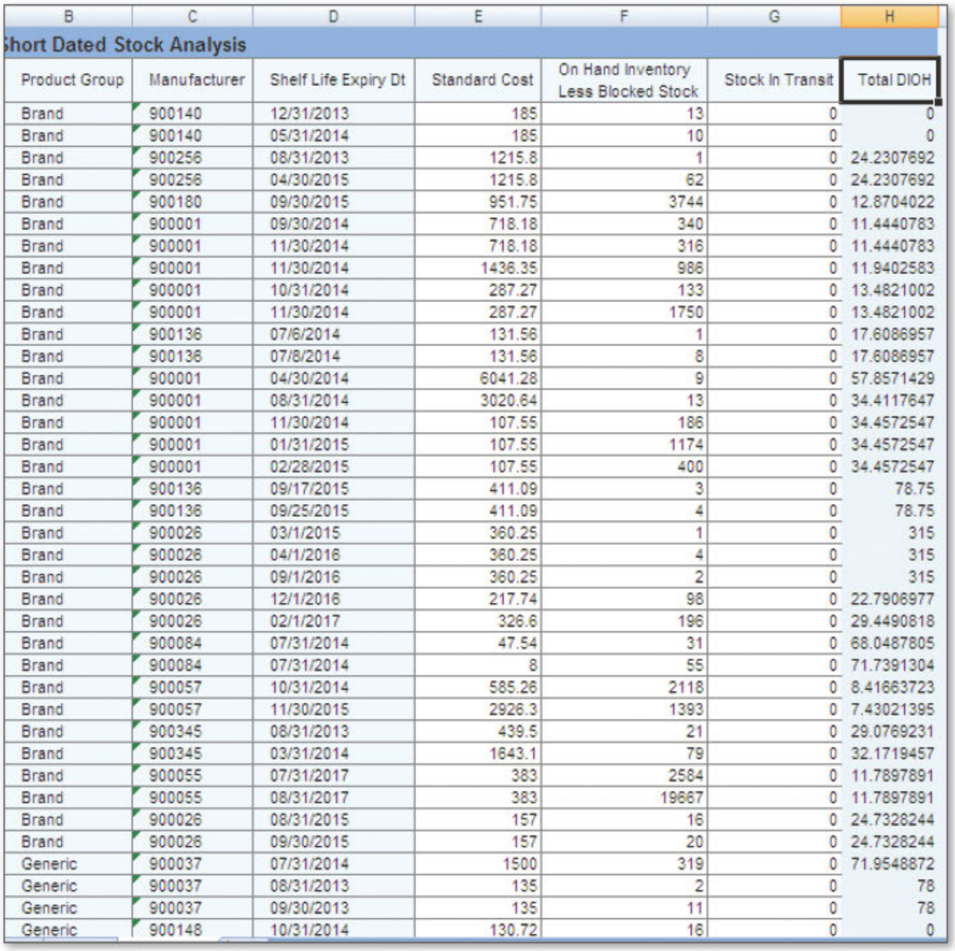
[✓ Display Icons] +/-

☐ Do Not Delete Number Formats When Converting to Formula

OK Cancel

Figure 2 Changing Data Provider to Reference Query2

5. Finally, create a table on the RESULTS tab that merges data from both queries (QUERY1 and QUERY2). Save the workbook (Figure 3).



B	C	D	E	F	G	H
Short Dated Stock Analysis						
Product Group	Manufacturer	Shelf Life Expiry Dt	Standard Cost	On Hand Inventory Less Blocked Stock	Stock In Transit	Total DIOH
Brand	900140	12/31/2013	185	13	0	0
Brand	900140	05/31/2014	185	10	0	0
Brand	900256	08/31/2013	1215.8	1	0	24.2307692
Brand	900256	04/30/2015	1215.8	62	0	24.2307692
Brand	900180	09/30/2015	951.75	3744	0	12.8704022
Brand	900001	09/30/2014	718.18	340	0	11.4440783
Brand	900001	11/30/2014	718.18	316	0	11.4440783
Brand	900001	11/30/2014	1436.35	986	0	11.9402583
Brand	900001	10/31/2014	287.27	133	0	13.4821002
Brand	900001	11/30/2014	287.27	1750	0	13.4821002
Brand	900136	07/6/2014	131.56	1	0	17.6086957
Brand	900136	07/8/2014	131.56	8	0	17.6086957
Brand	900001	04/30/2014	6041.28	9	0	57.8571429
Brand	900001	08/31/2014	3020.64	13	0	34.4117647
Brand	900001	11/30/2014	107.55	186	0	34.4572547
Brand	900001	01/31/2015	107.55	1174	0	34.4572547
Brand	900001	02/28/2015	107.55	400	0	34.4572547
Brand	900136	09/17/2015	411.09	3	0	78.75
Brand	900136	09/25/2015	411.09	4	0	78.75
Brand	900026	03/1/2015	360.25	1	0	315
Brand	900026	04/1/2016	360.25	4	0	315
Brand	900026	09/1/2016	360.25	2	0	315
Brand	900026	12/1/2016	217.74	98	0	22.7906977
Brand	900026	02/1/2017	326.6	196	0	29.4490818
Brand	900084	07/31/2014	47.54	31	0	68.0487805
Brand	900084	07/31/2014	8	55	0	71.7391304
Brand	900057	10/31/2014	585.26	2118	0	8.41663723
Brand	900057	11/30/2015	2926.3	1393	0	7.43021395
Brand	900345	08/31/2013	439.5	21	0	29.0769231
Brand	900345	03/31/2014	1643.1	79	0	32.1719457
Brand	900055	07/31/2017	383	2584	0	11.7897891
Brand	900055	08/31/2017	383	19667	0	11.7897891
Brand	900026	08/31/2015	157	16	0	24.7328244
Brand	900026	09/30/2015	157	20	0	24.7328244
Generic	900037	07/31/2014	1500	319	0	71.9548872
Generic	900037	08/31/2013	135	2	0	78
Generic	900037	09/30/2013	135	11	0	78
Generic	900148	10/31/2014	130.72	16	0	0

Figure 3 Add Design Items to Query2 Tab

You now have a simpler method to join data for analysis. This allows end users to perform this task on the frontend versus requesting IT to perform this task in SAP NetWeaver BW. Business users can quickly leverage this tip versus waiting on IT to remodel the data from an SAP NetWeaver BW perspective.

Tip 62

Modifying BEx Queries from SAP NetWeaver BW with the Use of Conditions

You can create BEx query conditions directly through the SAP NetWeaver BW interface to effect changes in a BEx query.

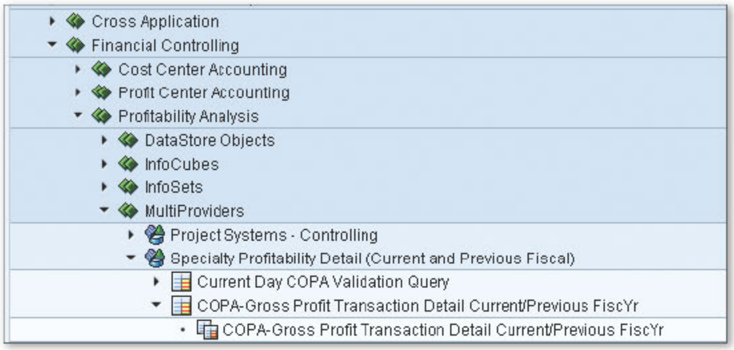
In most cases, you'll make any changes or modifications to BEx queries directly through the BEx Query Designer. However, the system offers an option that you can use to modify an existing BEx query if you're already in the SAP NetWeaver BW interface. For quick changes to the BEx query, this process eliminates the need to log in to the BEx Query Designer.

In this tip, we'll show you how to create conditions without the need for the BEx Query Designer, directly from the SAP NetWeaver BW interface.

✓ And Here's How ...

Log in to the SAP NetWeaver BW environment, execute Transaction RSCRM_BAPI, and follow these steps:

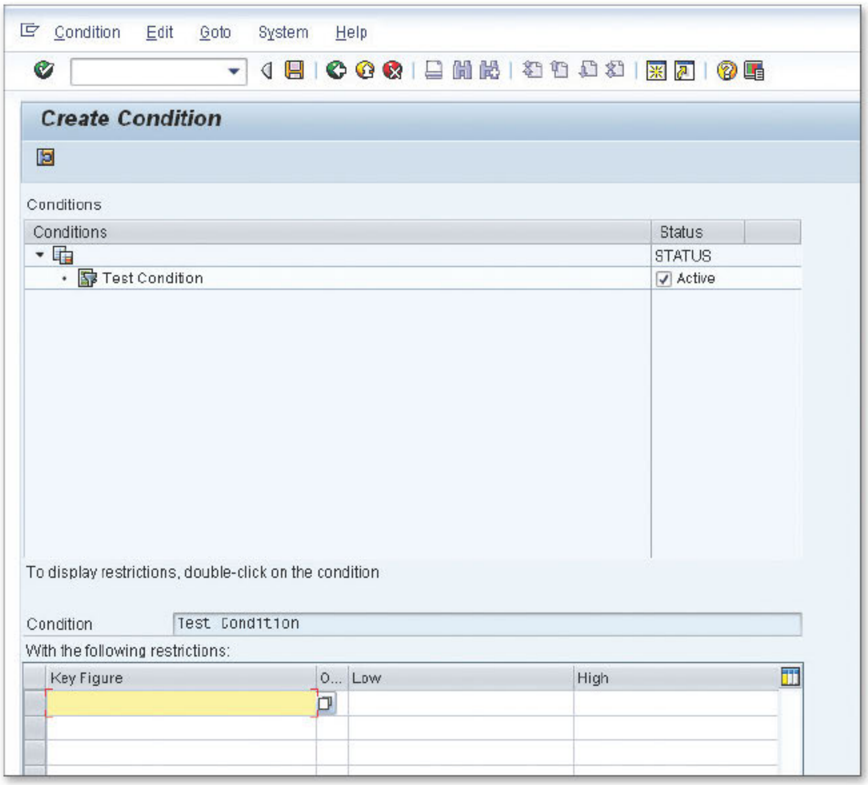
1. Navigate to the desired query and select it (see Figure 1).



- Cross Application
- Financial Controlling
  - Cost Center Accounting
  - Profit Center Accounting
  - Profitability Analysis
    - DataStore Objects
    - InfoCubes
    - InfoSets
    - MultiProviders
      - Project Systems - Controlling
      - Specialty Profitability Detail (Current and Previous Fiscal)
        - Current Day COPA Validation Query
        - COPA-Gross Profit Transaction Detail Current/Previous FiscYr
          - COPA-Gross Profit Transaction Detail Current/Previous FiscYr

Figure 1 Select Desired Query

2. After the query pane is open in SAP NetWeaver BW, click on the CONDITIONS button on the toolbar. You'll see the CREATE CONDITION area as shown in Figure 2.



Create Condition

Conditions

Conditions	Status
	STATUS
Test Condition	✓ Active

To display restrictions, double-click on the condition

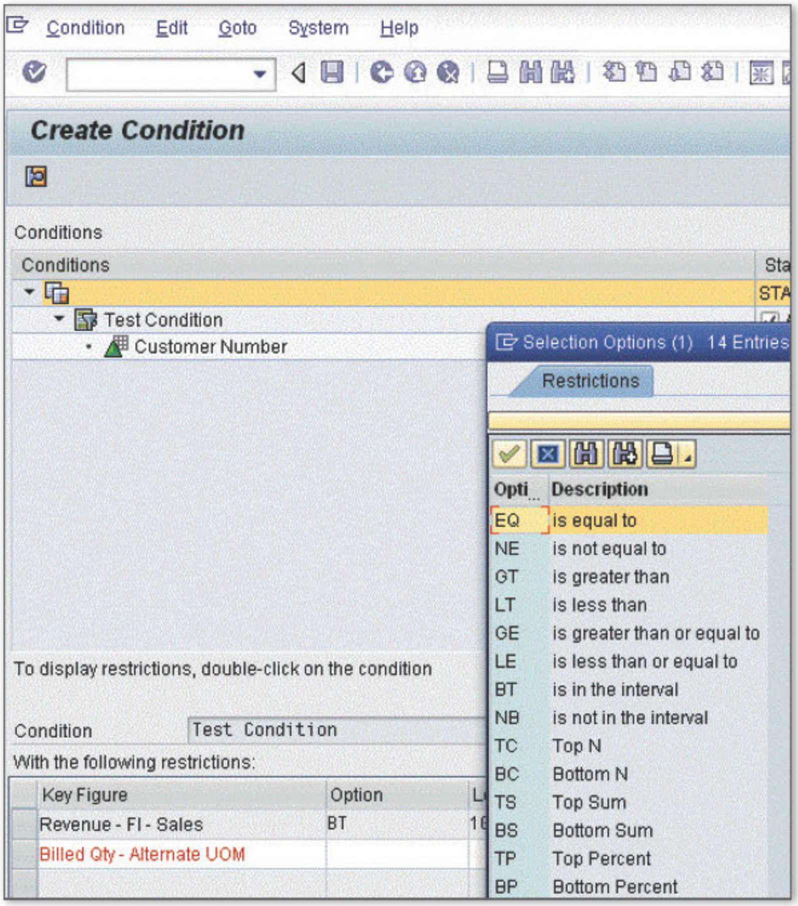
Condition: Test Condition

With the following restrictions:

Key Figure	O...	Low	High

Figure 2 Create Condition Pane

3. As shown in Figure 3, here we will create the condition by selecting the key figure (i.e., REVENUE – FI – SALES), Option (i.e., BT, which stands for IS IN THE INTERVAL), Low (i.e., 10000), and High (i.e., 1000000) condition restrictions.



To display restrictions, double-click on the condition

Condition	Test Condition

With the following restrictions:

Key Figure	Option	L
Revenue - FI - Sales	BT	1
Billed Qty - Alternate UOM		

Opti...	Description
EQ	is equal to
NE	is not equal to
GT	is greater than
LT	is less than
GE	is greater than or equal to
LE	is less than or equal to
BT	is in the interval
NB	is not in the interval
TC	Top N
BC	Bottom N
TS	Top Sum
BS	Bottom Sum
TP	Top Percent
BP	Bottom Percent

Figure 3 Selecting Key Figure, Option, and Low and High Condition Restrictions

Now the condition has been created. The condition will be enforced in either the SAP NetWeaver BW or BEx query interface.

Tip 63

Using Restricted Key Figures to Simulate the OR Operator in a Single Query

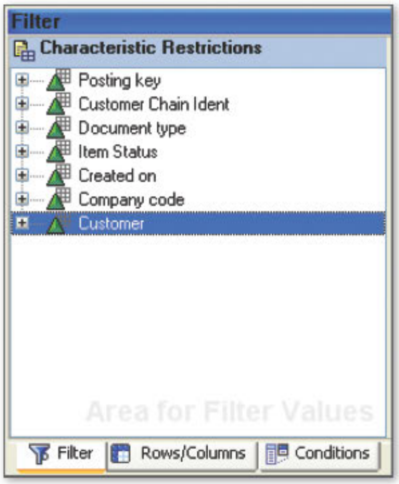
You can simulate the OR operator with variables in the BEx Query Designer to work with multiple variables in SAP NetWeaver BW.

When building a BEx query in conjunction with variables, the SAP NetWeaver BW system inherently handles the AND operator but unfortunately, not the OR operator. The use of the OR operator is common in cases where the user selects two or more related variables. Typically, to accomplish this, you have to separate the query into two parts. This tip provides a workaround so that you can accomplish this in one BEx query through the use of restricted key figures to handle the multiple variable selections.

✓ And Here's How ...

The examples in this tip assume two related characteristics: customer and customer chain. Follow these steps to simulate the OR operator when using variables in the BEx Query Designer:

1. Open the BEx Query Designer, and open an existing BEx query. After the query is open, observe the FILTER pane (Figure 1) and the ROWS/COLUMNS pane (Figure 2). Characteristics CUSTOMER and CUSTOMER CHAIN are present in the FILTER tab and ROWS/COLUMNS tab through the ROWS and FREE CHARACTERISTICS panes.



Filter

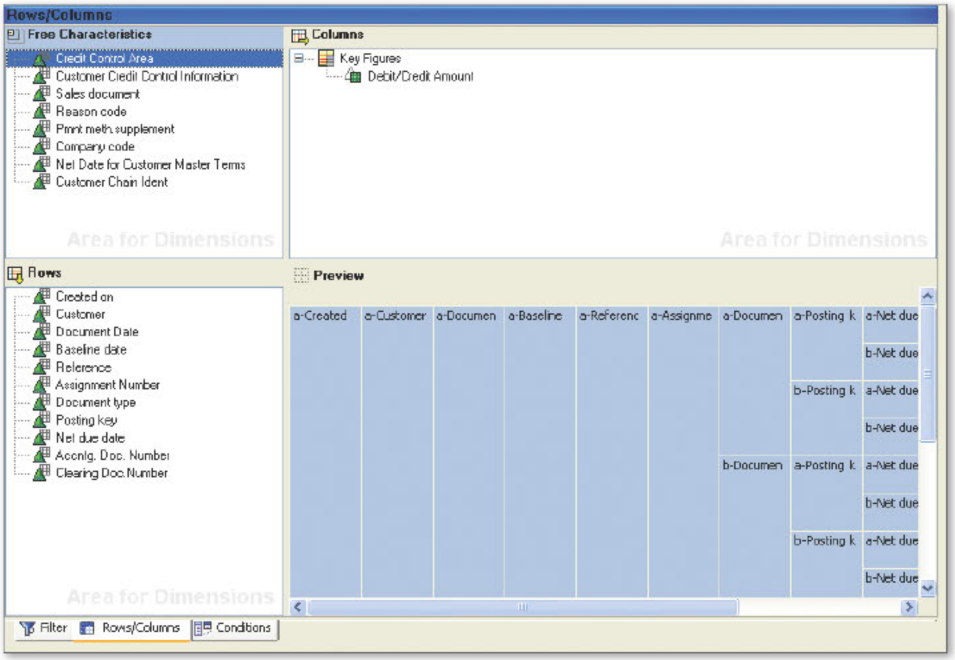
Characteristic Restrictions

- Posting key
- Customer Chain Ident
- Document type
- Item Status
- Created on
- Company code
- Customer

Area for Filter Values

Filter | Rows/Columns | Conditions

Figure 1 Filter Pane with Customer and Customer Chain ID



Rows/Columns

Free Characteristics

- Credit Control Area
- Customer Credit Control Information
- Sales document
- Reason code
- Pmnt meth. supplement
- Company code
- Net Date for Customer Master Terms
- Customer Chain Ident

Area for Dimensions

Columns

- Key Figures
  - Debit/Credit Amount

Area for Dimensions

Rows

- Created on
- Customer
- Document Date
- Baseline date
- Reference
- Assignment Number
- Document type
- Posting key
- Net due date
- Acctg. Doc. Number
- Clearing Doc.Number

Area for Dimensions

Preview

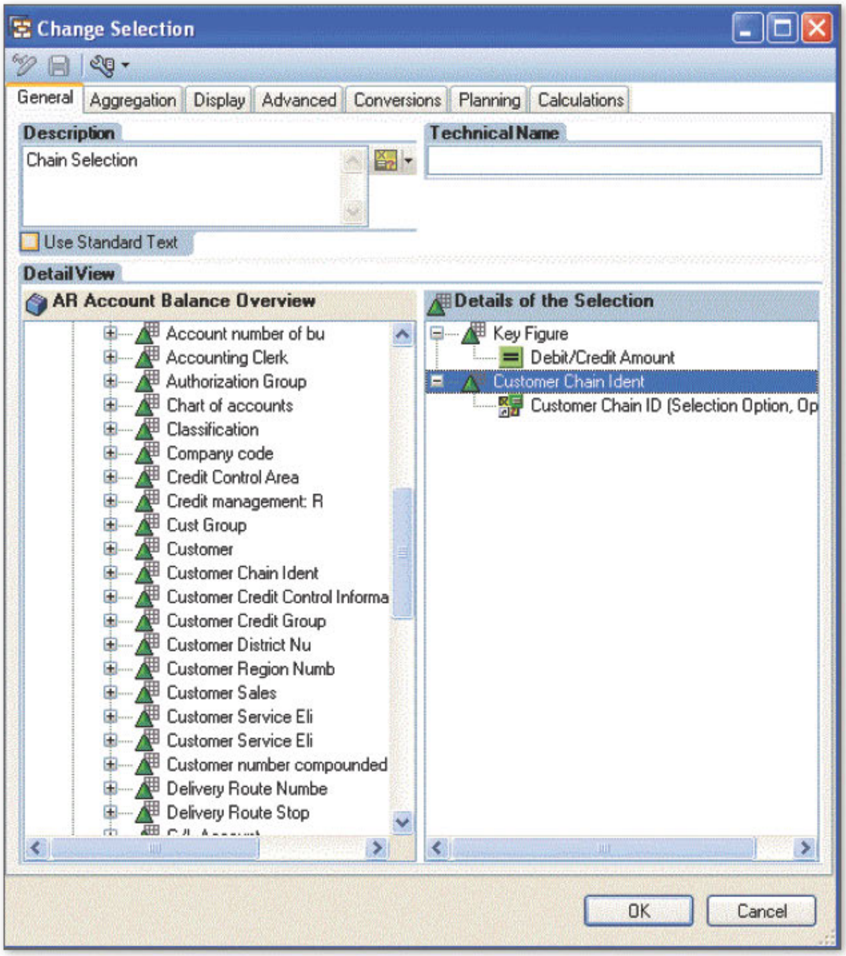
a-Created	a-Customer	a-Documen	a-Baseline	a-Referenc	a-Assignme	a-Documen	a-Posting k	a-Net due
								b-Net due
							b-Posting k	a-Net due
								b-Net due
						b-Documen	a-Posting k	a-Net due
								b-Net due
							b-Posting k	a-Net due
								b-Net due

Filter | Rows/Columns | Conditions

Figure 2 Rows/Columns Pane with Customer and Customer Chain ID

2. To create a new restricted key figure, right-click on the KEY FIGURES folder and select NEW RESTRICTED KEY FIGURE from the context menu.

3. Click and drag the key figure (i.e., Debit/Credit Amount) and variable (Customer) to the Detail View of the selection pane, and enter "Customer selection" in the Description field.
4. Create another restricted key figure; click and drag the key figure (i.e., Debit/Credit Amount) and the variable (Customer Chain) to the Details View of the selection pane, and enter "Chain selection" in the Description field (Figure 3).



Description: Chain Selection

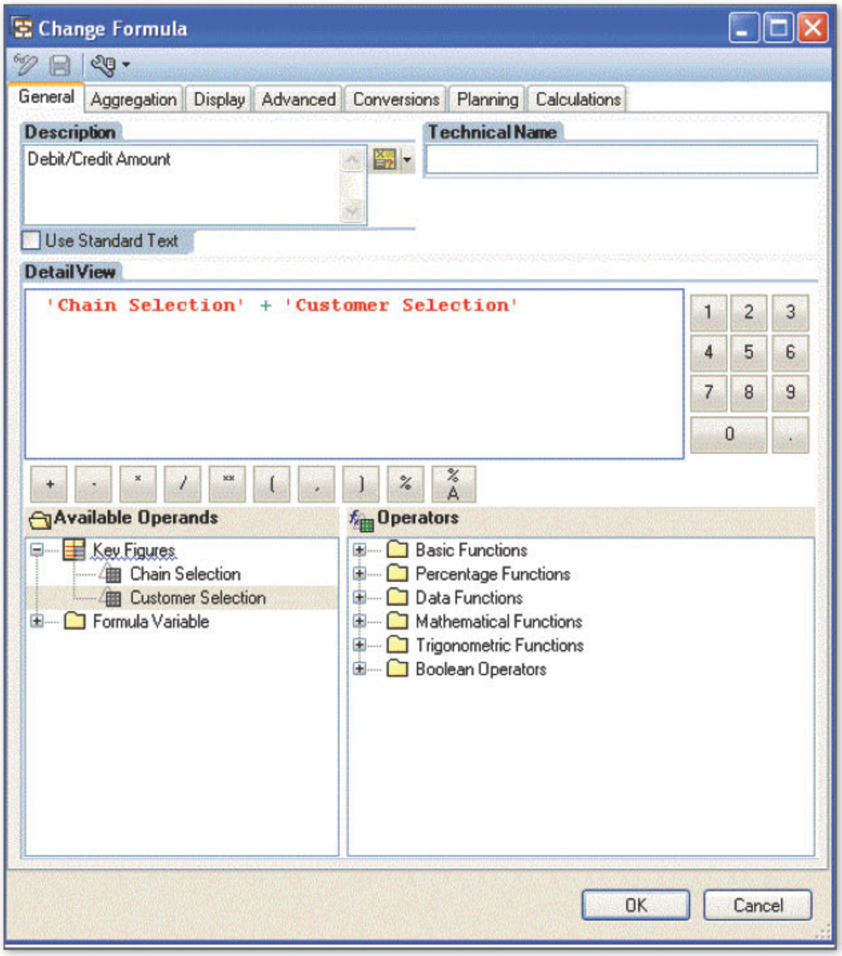
AR Account Balance Overview:
- Account number of bu
- Accounting Clerk
- Authorization Group
- Chart of accounts
- Classification
- Company code
- Credit Control Area
- Credit management: R
- Cust Group
- Customer
- Customer Chain Ident
- Customer Credit Control Informa
- Customer Credit Group
- Customer District Nu
- Customer Region Numb
- Customer Sales
- Customer Service Eli
- Customer Service Eli
- Customer number compounded
- Delivery Route Numbe
- Delivery Route Stop

Details of the Selection:
- Key Figure
  - Debit/Credit Amount
- Customer Chain Ident
  - Customer Chain ID (Selection Option, Op

Figure 3 Create Restricted Key Figure with Customer Chain Variable

5. Create a new calculated key figure by right-clicking on the Key Figures folder and selecting New Calculated Key Figure from the context menu.

6. Click and drag the newly created restricted key figures to the DETAIL VIEW, and append a + sign (e.g., 'CHAIN SELECTION' + 'CUSTOMER SELECTION') as shown in Figure 4. Enter "Debit/Credit Amount" in the DESCRIPTION field.



Description: Debit/Credit Amount

Detail View:

```
'Chain Selection' + 'Customer Selection'
```

Figure 4 Create Calculated Key Figure with Both Restricted Key Figures

7. Now the calculated key figure—DEBIT/CREDIT AMOUNT—has been created. Run the query to filter on either customer or customer chain variables.

Part 4

SAP NetWeaver BW Administration and Development

Things You'll Learn in this Section

This part of the book will cover some key areas of SAP NetWeaver BW administration. This includes system performance optimization and tuning in the SAP NetWeaver BW system, as it drives the user adoptability and lower total cost of ownership. We'll also provide insight into performance analysis and the related critical housekeeping that administrators will need to be familiar with to keep the system running smoothly. We'll take a brief segway into authorizations in SAP NetWeaver BW as a part of administration, as well.

Tip 64

Restoring a BEx Query from Version 7.x to 3.x

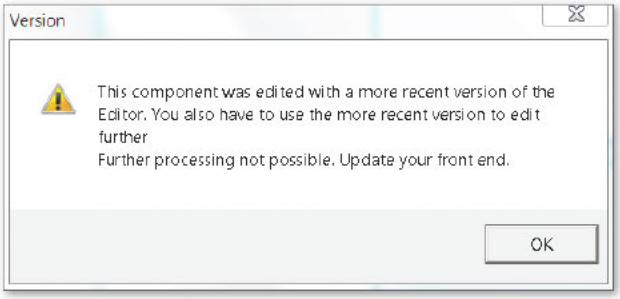
If you have a dual SAP NetWeaver BW environment, you can restore BEx 3.x version queries that are accidently saved as a 7.x version without having to recreate the query.

In a dual SAP NetWeaver BW environment where 7.x and 3.x queries coexist, a common issue is the accidental conversion of 3.x queries to the 7.x version. For instance, if you're changing a 3.x version of the query, and one of the 7.x features that is not available in the 3.x version is inadvertently used, the query will be saved as a 7.x version. And, when the saved query is opened in the 3.x version, the system displays a message stating, "Query definition version 7.x is NOT compatible with the versions 3.x." The normal resolution for this issue is to recreate the query in 3.x.

This tip will show you how to restore accidentally converted 7.x queries back to the 3.x version using query backup and the SAP standard query restore utility program, without having to recreate the query. This will minimize the impact of this issue because the queries can be restored with minimal effort.

✓ And Here's How ...

Saved queries or query components can *only* be edited using the more recent version of the editor. The message shown in Figure 1 appears for any query edited with Query Designer 7.x and opened with Query Designer 3.x.


Version

This component was edited with a more recent version of the Editor. You also have to use the more recent version to edit further
Further processing not possible. Update your front end.

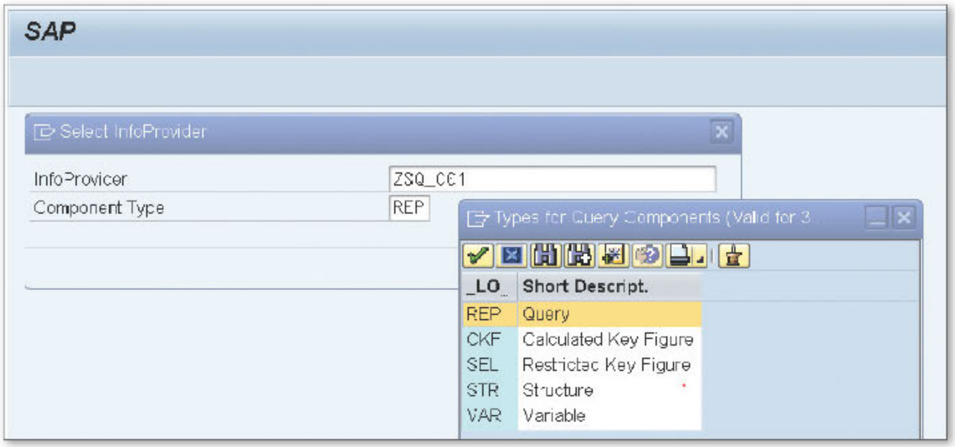
Figure 1 Error Message When a 7.x Version Query Is Opened in 3.x

Any query or query component originally created in the SAP NetWeaver BW 3.x system is backed up when this component is opened for editing in Query Designer. The backed-up queries and query components definitions in 3.x compatible format will be stored in the database tables that contain the query definition.

Note that queries and query components originally created in SAP NetWeaver BW 7.x have no backup version.

Follow these steps to convert a query:

1. Go to Transaction SE38, enter program name "COMPONENT_RESTORE", and click EXECUTE.
2. In the following screen, enter the INFOPROVIDER name, select the COMPONENT TYPE, and click EXECUTE. The COMPONENT TYPE lists all types of components that can be restored. For this scenario, select the REP type that indicates QUERY (see Figure 2).

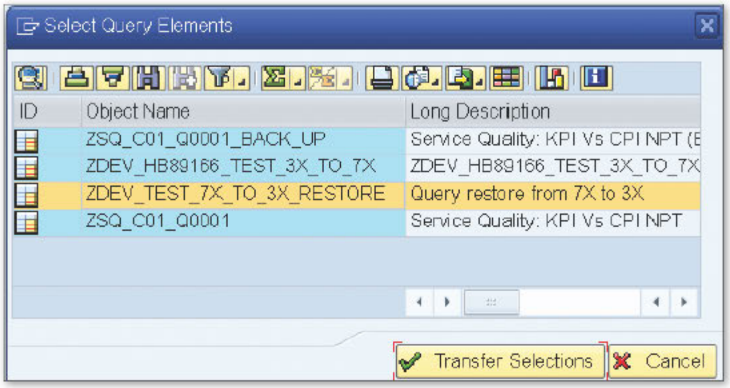


InfoProvider	ZSQ_CC1
Component Type	REP

_LO	Short Descript.
REP	Query
CKF	Calculated Key Figure
SEL	Restricted Key Figure
STR	Structure
VAR	Variable

Figure 2 Types of Query Components That Can Be Restored

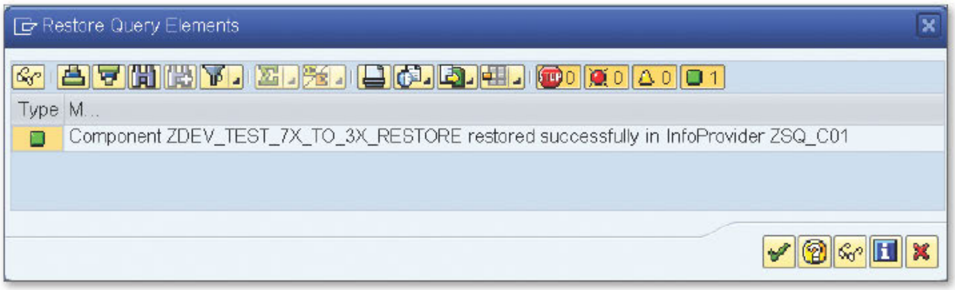
3. In the following screen shown in Figure 3, select the query that has to be restored to version 3.x.



ID	Object Name	Long Description
	ZSQ_C01_Q0001_BACK_UP	Service Quality: KPI Vs CPI NPT (E
	ZDEV_HB89166_TEST_3X_TO_7X	ZDEV_HB89166_TEST_3X_TO_7X
	ZDEV_TEST_7X_TO_3X_RESTORE	Query restore from 7X to 3X
	ZSQ_C01_Q0001	Service Quality: KPI Vs CPI NPT

Figure 3 Selecting Query to Be Restored

4. In the next screen, click on YES to confirm the restore. The system displays a success message as shown in Figure 4.



Type	M...
🟩	Component ZDEV_TEST_7X_TO_3X_RESTORE restored successfully in InfoProvider ZSQ_C01

Figure 4 Successful Restoration

Using a standard SAP program and few simple steps, you can restore any accidently converted 7.x queries back to the 3.x version, thus minimizing the overall impact when undesired migration of queries or reusable query components occurs from version 3.x to version 7.x.

Tip 65

Documenting SAP NetWeaver BW Transformations

You can document transformations for multiple tracking purposes using some features available in SAP NetWeaver BW.

In a typical BI project environment, you are always faced with the task of documenting the technical design per the project methodology. The technical documentation is a useful tool for conducting reviews, transitioning information to the support team, and serving as a reference point on the design. However, SAP NetWeaver BW doesn't offer an easy way of capturing documentation on technical objects, so developers usually spend lot of time manually documenting the objects.

This tip discusses a couple of intuitive approaches that can be used to document transformations in SAP NetWeaver BW.

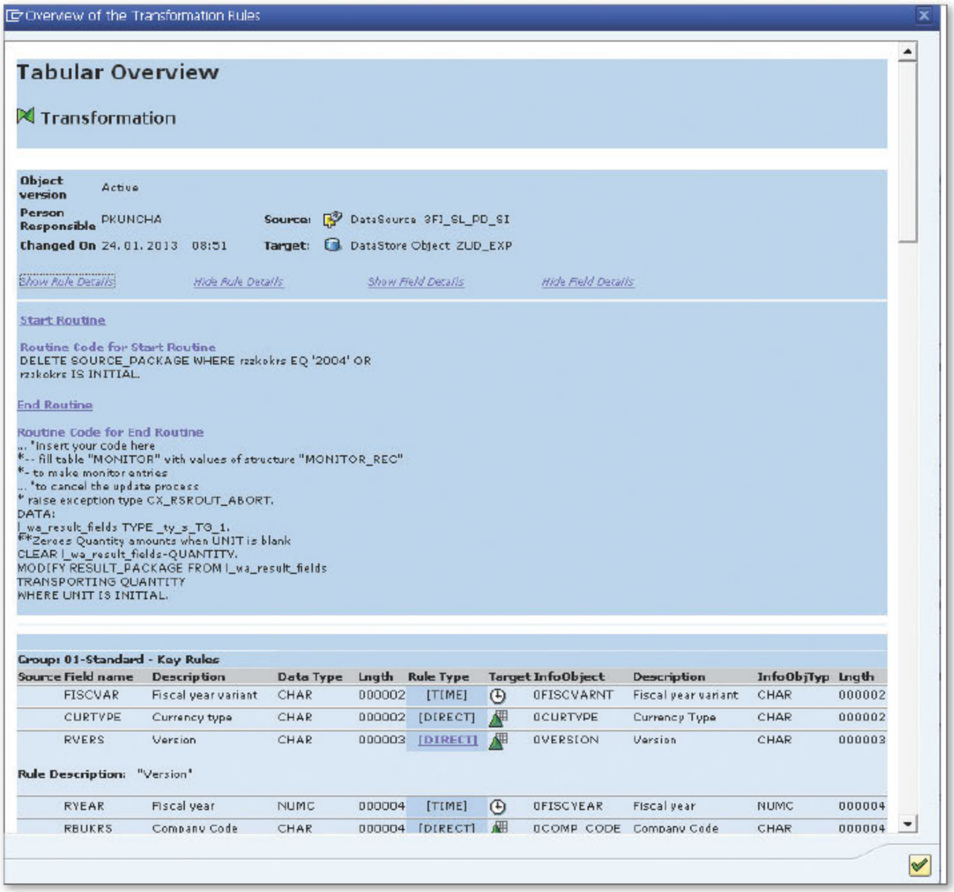
✓ And Here's How ...

In this tip, we'll discuss two approaches to documenting transformations in SAP NetWeaver BW, which you'll find in the following sections.

Tabular View Option

With this option, access the SAP NetWeaver BW system and execute Transaction RSA1. Open the transformation that you want to document, and click EXTRAS • TABULAR VIEW.

If you click on the SHOW RULE DETAILS link, it will show the details of START ROUTINE/END ROUTINE/EXPERT ROUTINE. In this example, the transformation has a start routine and end routine, so when you click on Show Rule Details, the system shows the code inside the routine (see Figure 1).



Overview of the Transformation Rules

Tabular Overview

Transformation

Object version	Active		
Person Responsible	PKUNCHA	**Source:**	DataSource 3FI_SL_PD_SI
Changed On	24.01.2013 08:51	**Target:**	DataStore Object ZUD_EXP

Show Rule Details   Hide Rule Details   Show Field Details   Hide Field Details

Start Routine

Routine Code for Start Routine

```
DELETE SOURCE_PACKAGE WHERE rzzkokrs EQ '2004' OR
rzzkokrs IS INITIAL.
```

End Routine

Routine Code for End Routine

```
... "insert your code here
*-- fill table "MONITOR" with values of structure "MONITOR_REC"
*- to make monitor entries
... "to cancel the update process
* raise exception type CX_RSROUT_ABORT.
DATA:
l_wa_result_fields TYPE _ty_s_TG_1.
**Zeroes Quantity amounts when UNIT is blank
CLEAR l_wa_result_fields-QUANTITY.
MODIFY RESULT_PACKAGE FROM l_wa_result_fields
TRANSPORTING QUANTITY
WHERE UNIT IS INITIAL.
```

Group: 01-Standard - Key Rules

Source Field name	Description	Data Type	Lngth	Rule Type	Target InfoObject	Description	InfoObjTyp	Lngth
FISCVAR	Fiscal year variant	CHAR	000002	[TIME]	0FISCVARNT	Fiscal year variant	CHAR	000002
CURTYPE	Currency type	CHAR	000002	[DIRECT]	0CURTYPE	Currency Type	CHAR	000002
RVERS	Version	CHAR	000003	[DIRECT]	0VERSION	Version	CHAR	000003
Rule Description:	"Version"							
RYEAR	Fiscal year	NUMC	000004	[TIME]	0FISCYEAR	Fiscal year	NUMC	000004
RBUKRS	Company Code	CHAR	000004	[DIRECT]	0COMP_CODE	Company Code	CHAR	000004

Figure 1 Show Rule Details View

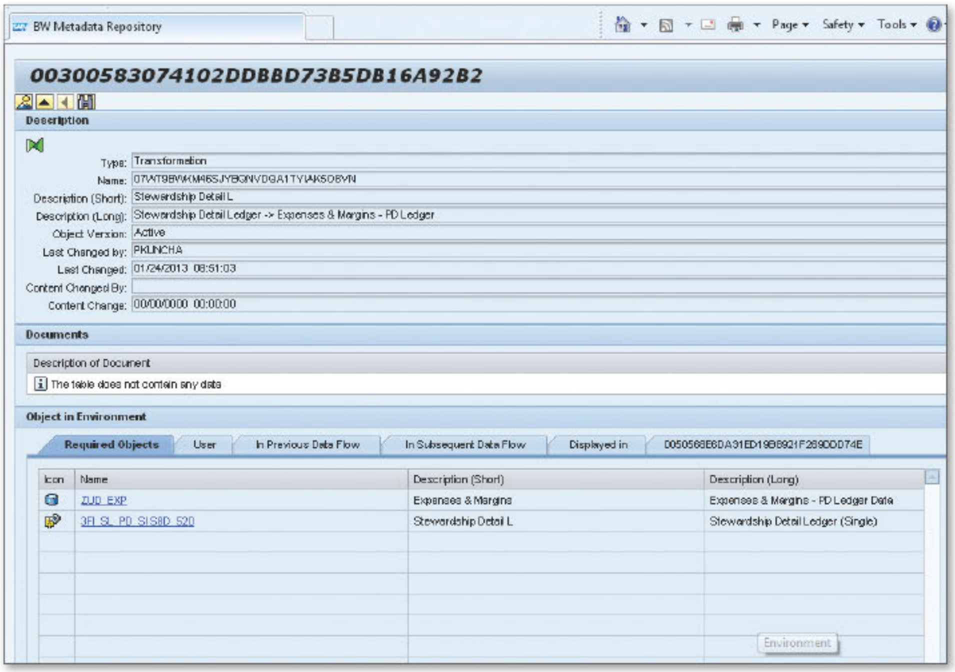
If you click on Show Field Details, the system shows the detail information of all of the InfoObjects in the transformation, such as Description of InfoObject, InfoObject type, InfoObject Length, and so on.

You can copy the content of these windows into your documentation using the clipboard or snipping tools.

SAP NetWeaver BW Metadata Repository

This option displays the details of the transformations in the SAP NetWeaver BW Metadata Repository by opening a web browser.

Right-click on the transformation that you want to document, and click DOCUMENTATION. A new browser window pops up. The new window will have all of the detail information of the transformation as shown in Figure 2.



003005830741O2DDBBD73B5DB16A92B2

Description

Type:	Transformation
Name:	07WT9BWKM46SJYBGNVDGA1TYIAK5OBVN
Description (Short):	Stewardship Detail L
Description (Long):	Stewardship Detail Ledger -> Expenses & Margins - PD Ledger
Object Version:	Active
Last Changed by:	PKLINCHA
Last Changed:	01/24/2013 08:51:03
Content Changed By:	
Content Change:	00/00/0000 00:00:00

Documents

Description of Document
The table does not contain any data

Object in Environment

Required Objects | User | In Previous Data Flow | In Subsequent Data Flow | Displayed in | 0050568E6DA31ED19B8921F289DD074E

Icon	Name	Description (Short)	Description (Long)
	ZUD_EXP	Expenses & Margins	Expenses & Margins - PD Ledger Data
	3FI_SL_PD_SIS8D_520	Stewardship Detail L	Stewardship Detail Ledger (Single)

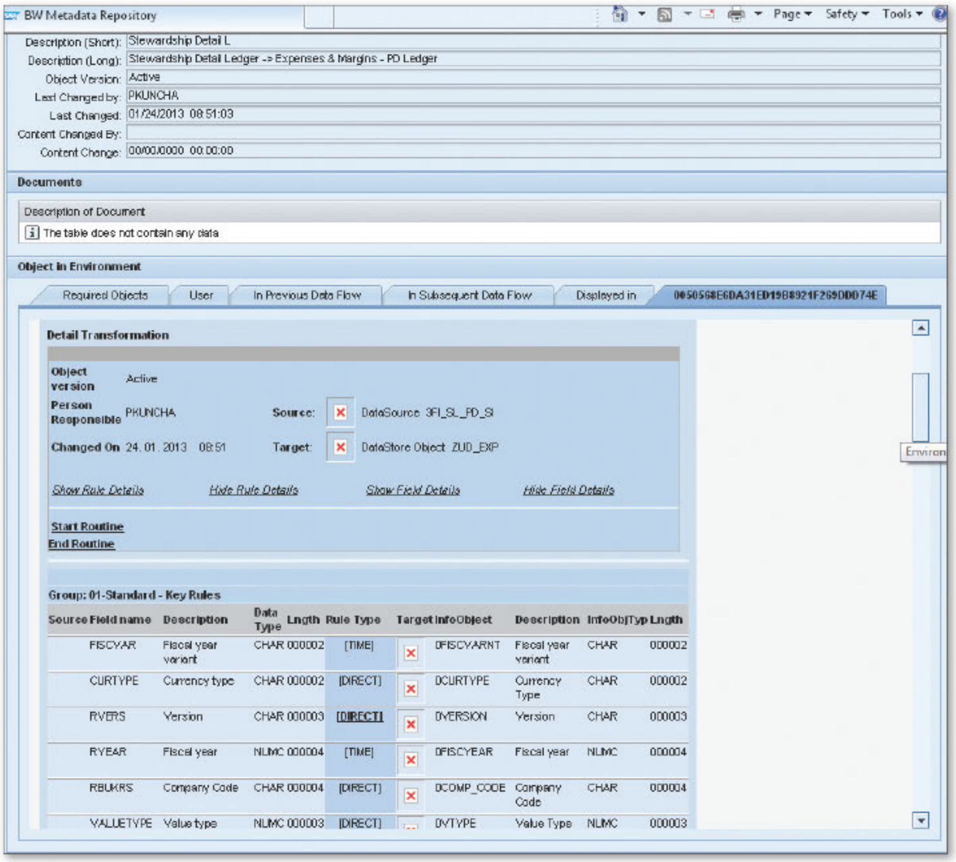
Figure 2 Display Transformation

Take note of the following sections and tabs on the screen:

- DESCRIPTION
  This section at the top of the screen includes the details such as the technical NAME of the transformation, the DESCRIPTION, and so on.
- REQUIRED OBJECTS
  This tab includes the source and destination of the transformation.
- USER
  This tab includes the DTP information.

- ▶ In Previous Data Flow
  This tab lists the source of the transformation.
- ▶ In Subsequent Data Flow
  This tab lists the target of the transformation.

The last tab on the screen will have a generated name and include the InfoObject mapping details of the transformation as shown in Figure 3.



Description (Short):	Stewardship Detail L
Description (Long):	Stewardship Detail Ledger -> Expenses & Margins - PD Ledger
Object Version:	Active
Last Changed by:	PKUNCHA
Last Changed:	01/24/2013 08:51:03
Content Changed By:	
Content Change:	00/00/0000 00:00:00

Documents

Description of Document

The table does not contain any data

Object in Environment

Required Objects | User | In Previous Data Flow | In Subsequent Data Flow | Displayed in | 005056BE6DA31ED19B8921F269DD074E

Detail Transformation

Object version Active

Person Responsible PKUNCHA **Source:** DataSource 3FI_SL_PD_SI

Changed On 24. 01. 2013 08:51 **Target:** DataStore Object ZUD_EXP

Show Rule Details *Hide Rule Details* *Show Field Details* *Hide Field Details*

Start Routine
End Routine

Group: 01-Standard - Key Rules

Source Field name	Description	Data Type	Lngth	Rule Type	Target InfoObject	Description	InfoObjTyp	Lngth
FISCYAR	Fiscal year variant	CHAR	000002	[TIME]	0FISCYARNT	Fiscal year variant	CHAR	000002
CURTYPE	Currency type	CHAR	000002	[DIRECT]	0CURTYPE	Currency Type	CHAR	000002
RVERS	Version	CHAR	000003	[DIRECT]	0VERSION	Version	CHAR	000003
RYEAR	Fiscal year	NUMC	000004	[TIME]	0FISCYEAR	Fiscal year	NUMC	000004
RBUKRS	Company Code	CHAR	000004	[DIRECT]	0COMP_CODE	Company Code	CHAR	000004
VALUETYPE	Value type	NUMC	000003	[DIRECT]	0VTYPE	Value Type	NUMC	000003

Figure 3 InfoObject Mapping Details

You can either save this page in HTML format or embed it into your documentation as an attached file. Another possibility is to just include the URL link.

Tip 66

Clearing the Logistics Cockpit Delta Queue to Ensure Data Consistency

To avoid losing data or finding data inconsistency, you can clear the Logistics (LO) Cockpit delta queue and move data to SAP NetWeaver BW prior to making any DataSource structure changes.

To support specific business requirements, it's very common to add new fields to the extract structure of the Logistics Cockpit. However, this process is always a major challenge for SAP NetWeaver BW developers because any mistake in this process can result in inconsistency of data on the reporting side.

Because the Logistics Cockpit DataSources use special processing queues and programs to capture delta updates, to ensure that you don't lose any data, you first need to properly clear all of the pending delta and move it to SAP NetWeaver BW before any structure changes are made to the DataSource in SAP ERP. This tip discusses the steps you need to implement to clear LO Cockpit delta queues prior to moving structure changes to a LO Cockpit DataSource.

✓ And Here's How ...

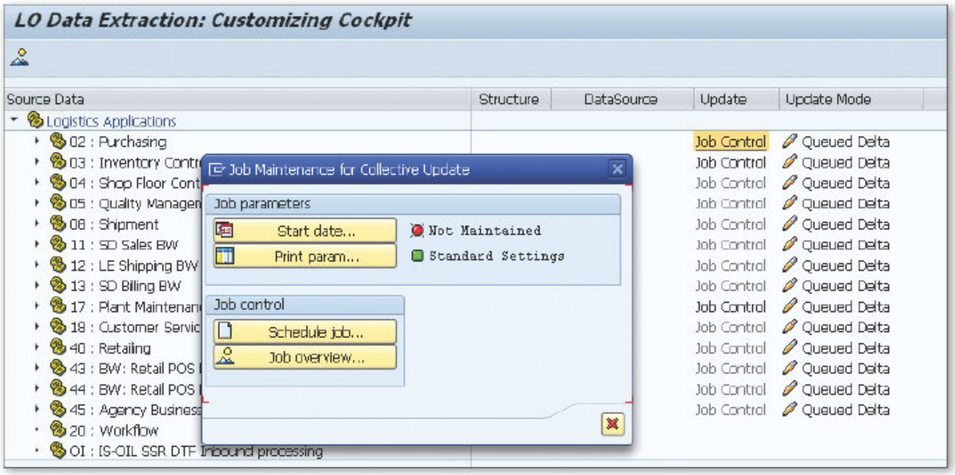
You should only execute this tip in the SAP ERP system when no transactions are being posted for this application. You can also lock the corresponding transactions from user access and stop any background jobs from posting any updates to the application for which the DataSource structure change is being processed. For this tip, we'll use a Purchasing DataSource, 2LIS_02_ITM, to illustrate the steps.

In the source system (SAP ERP system), stop the V3 delta collection job that is associated with the extractor being changed by removing the corresponding V3 job from the schedule. For this, you can use Transaction LBWE and click on Job Control on the 2LIS_02_ITM line.

The naming convention for the V3 collection job will be LIS-BW-VB-APPLICATION_<application>_<client number>, if the V3 job was directly scheduled in your system using Transaction LBWE. Alternatively, you can use the job name that was scheduled from Transaction SM36 or use any scheduling tools. For example, if the Transaction LBWE Job Control option was used to schedule these jobs, the background jobs will have the following naming convention for these applications:

- LIS-BW-VB_APPLICATION_02_<client number> - Purchasing
- LIS-BW-VB_APPLICATION_03_<client number> - Inventory
- LIS-BW-VB_APPLICATION_17_<client number> - Plant Maintenance

After the V3 delta collective run job is removed from the schedule, manually execute this job to process any pending V3 delta records so that they can be moved to the RSA7 SAP NetWeaver BW delta queue. Execute this job from Transaction LBWE, and then choose Job Control • Schedule Job (see Figure 1).


LO Data Extraction: Customizing Cockpit

Source Data	Structure	DataSource	Update	Update Mode
Logistics Applications				
02 : Purchasing			Job Control	Queued Delta
03 : Inventory Contr			Job Control	Queued Delta
04 : Shop Floor Cont			Job Control	Queued Delta
05 : Quality Managem			Job Control	Queued Delta
08 : Shipment			Job Control	Queued Delta
11 : SD Sales BW			Job Control	Queued Delta
12 : LE Shipping BW			Job Control	Queued Delta
13 : SD Billing BW			Job Control	Queued Delta
17 : Plant Maintenan			Job Control	Queued Delta
18 : Customer Servic			Job Control	Queued Delta
40 : Retailing			Job Control	Queued Delta
43 : BW: Retail POS			Job Control	Queued Delta
44 : BW: Retail POS			Job Control	Queued Delta
45 : Agency Business			Job Control	Queued Delta
20 : Workflow				
OI : IS-OIL SSR DTF Inbound processing				

Job Maintenance for Collective Update

Job parameters

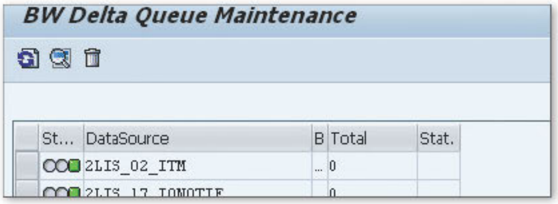
Start date...	🔴 Not Maintained
Print param...	🟩 Standard Settings

Job control

Figure 1 LO Cockpit Extraction: Scheduling and Executing Jobs

Extract the delta twice to SAP NetWeaver BW using the corresponding InfoPackage to ensure that any pending delta in the RSA7 SAP NetWeaver BW delta queue in SAP ERP is loaded to SAP NetWeaver BW. Note that the second delta must bring

0 records only, which is an indication that all pending deltas have been extracted to SAP NetWeaver BW (see Figure 2).


BW Delta Queue Maintenance

St...	DataSource	B	Total	Stat.
○○🟩	2LIS_02_ITM	...	0	
○○🟩	2LIS_17_I0NOTIF		0	

Figure 2 RSA7 SAP NetWeaver BW Delta Queue Maintenance

If any data is still left in the SAP ERP setup table from prior initializations, delete the data using Transaction LBWG for the corresponding application to which the LO Cockpit extractor belongs.

Transport the extractor structure changes from the source SAP NetWeaver BW system to the target SAP NetWeaver BW system (development to acceptance, development to production, etc.). Activate the replicated DataSource in SAP NetWeaver BW, and send any SAP NetWeaver BW transports to map any new fields from the DataSource to the DSO (if applicable).

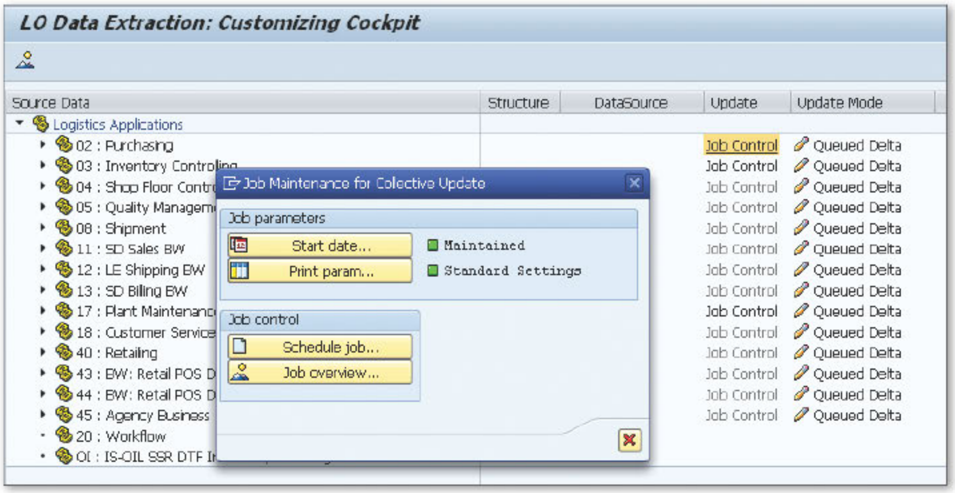
Delete and reinitialize the DataSource in SAP NetWeaver BW with no data transfer option using the appropriate InfoPackage available for the DataSource.

Unlock the SAP ERP transactions (if applicable) or open the system back up for user access, and put back any background jobs on schedule so that updates to the application can be posted to the system.

Put the V3 delta collection job back into its schedule in SAP ERP with the desired frequency (hourly, every 30 minutes, daily, etc.) as shown in Figure 3.

Transaction SM37 shows the V3 delta collective run job has been scheduled to run (see Figure 4).

Schedule a delta InfoPackage in SAP NetWeaver BW and ensure that data is extracted to SAP NetWeaver BW if any new deltas have been created in SAP ERP since the system was reopened.



LO Data Extraction: Customizing Cockpit

Source Data	Structure	DataSource	Update	Update Mode
Logistics Applications				
02 : Purchasing			Job Control	Queued Delta
03 : Inventory Controlling			Job Control	Queued Delta
04 : Shop Floor Contr			Job Control	Queued Delta
05 : Quality Managem			Job Control	Queued Delta
08 : Shipment			Job Control	Queued Delta
11 : SD Sales BW			Job Control	Queued Delta
12 : LE Shipping BW			Job Control	Queued Delta
13 : SD Billing BW			Job Control	Queued Delta
17 : Plant Maintenanc			Job Control	Queued Delta
18 : Customer Service			Job Control	Queued Delta
40 : Retailing			Job Control	Queued Delta
43 : BW: Retail POS D			Job Control	Queued Delta
44 : BW: Retail POS D			Job Control	Queued Delta
45 : Agency Business			Job Control	Queued Delta
20 : Workflow				
OI : IS-OIL SSR DTF Ir				

Job Maintenance for Collective Update

Job parameters

Start date... 🟩 Maintained

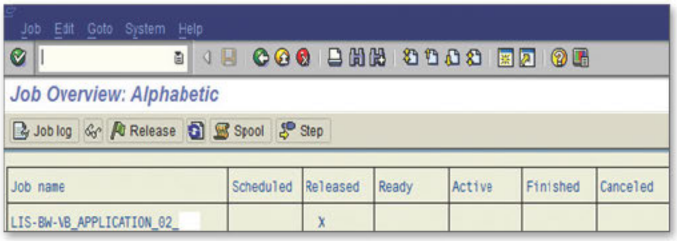
Print param... 🟩 Standard Settings

Job control

Schedule job...

Job overview...

Figure 3 Scheduling the Job



Job Edit Goto System Help

Job Overview: Alphabetic

Job log | Release | Spool | Step

Job name	Scheduled	Released	Ready	Active	Finished	Canceled
LIS-BW-VB_APPLICATION_02_		X				

Figure 4 Transaction SM37 Screen Showing the V3 Delta Collection Job

Tip 67

Converting Logical System Names during a System Copy

When you copy one SAP NetWeaver BW system to another, you can convert the logical system name so that it points to the correct place without larger system consequences.

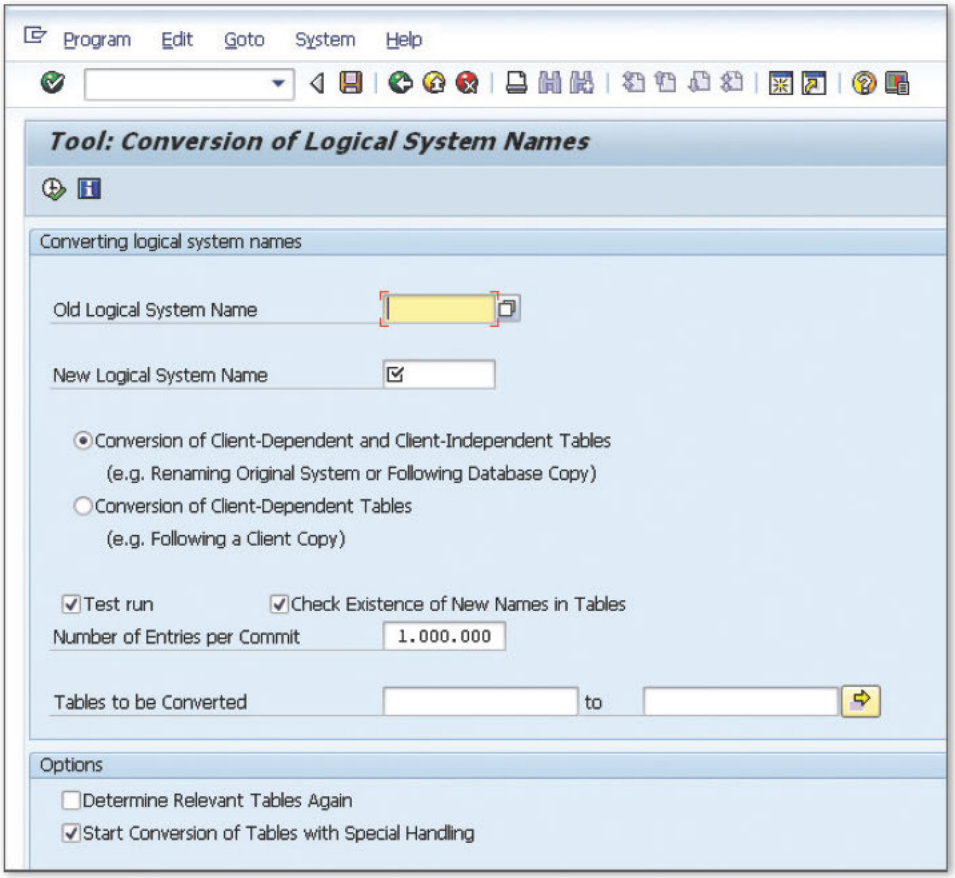
From time to time, you may need to copy one SAP NetWeaver BW system to another SAP NetWeaver BW system (e.g., the test SAP NetWeaver BW [TBW] system needs to be built from the production SAP NetWeaver BW system or migrated from one system to another). To facilitate this process, you'll perform a system copy. However, the source system for data loads will still point to the old logical system name. So the source system assignments in all the transformations, DTPs, and so on need to be changed to the new logical system name for the new source system.

This tip describes how to convert a logical system name to change the source system assignment details without any impact to the underlying transfer rules, transformations, and DTPs.

✓ And Here's How ...

Before starting this tip, you need to make sure that an RFC connection for the new source system is already created, and the connection to the source system is tested successfully.

To change the source system assignment and convert the logical system name, run Transaction BDLS in the SAP NetWeaver BW system. The screen shown in Figure 1 is displayed.



Tool: Conversion of Logical System Names

Converting logical system names

Old Logical System Name	
New Logical System Name	

(●) Conversion of Client-Dependent and Client-Independent Tables
(e.g. Renaming Original System or Following Database Copy)
( ) Conversion of Client-Dependent Tables
(e.g. Following a Client Copy)

[✓] Test run [✓] Check Existence of New Names in Tables

Number of Entries per Commit	1.000.000
Tables to be Converted	to

Options

[ ] Determine Relevant Tables Again
[✓] Start Conversion of Tables with Special Handling

Figure 1 Logical System Conversion Screen

This screen has several parameters that can be used for the conversion of logical system names:

- Test run
  Allows you to run this transaction in a test mode to check and correct any errors or warnings. The test mode will generate a log that needs to be analyzed before the original run.
- Check Existence of New Names in Tables
  The system will check whether the new logical system exists in any of the tables.

You can execute this transaction in two modes:

- Conversion of Client-Dependent and Client-Independent Tables
  Can be used if you are renaming an original system or converting the logical system after a database copy.
- Conversion of Client-Dependent Tables
  Can be used if you are converting the logical system name after a client copy.

Now follow these steps:

1. Specify the parameters for old and new logical system names. For example, if the old source system was ECC-100 (production) and the new source system is ECC-200 (test), then enter the names in the Old Logical System Name and New Logical System Name fields.
2. Choose the Conversion of Client-Dependent and Client-Independent Tables mode, check the Test Run option, and check the Check Existence of New Names in Tables option.
3. Click on the Execute button. The test mode generates a log that needs to be analyzed before the original run. Any errors that appear need to be resolved before you proceed further.
4. Analyze the warning messages as well to check whether the warnings are critical or not. You may have to execute this transaction with Test run a couple of times to ensure that there are no errors.
5. Uncheck the Test run and Check Existence of New Names in Tables options, and click on the Execute button. This job might take several minutes to complete depending on the number of objects in the system.

When the job is completed, you can see that the old logical system name is converted to the new logical system name. If the job executes without errors, then you can resume your data loads from the new source system.

If there are certain applications that have been built using the old logical system name, these will have to be manually converted/changed.

Tip 68

Creating a Custom Method to Search for Invalid Characters in an InfoObject

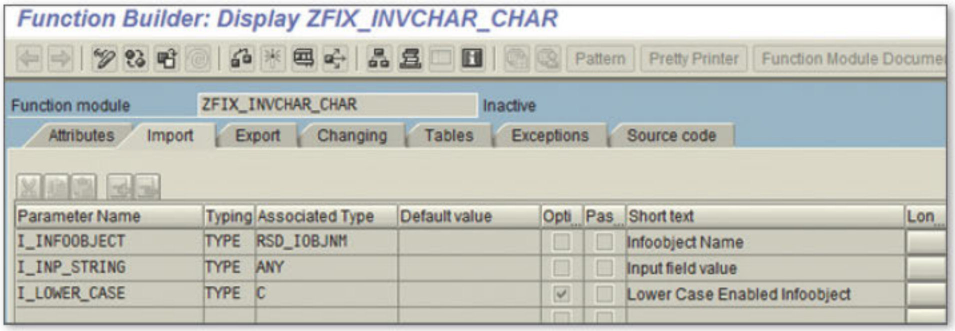
You can create a custom function module to help prevent load failure by checking for invalid characters in an InfoObject.

Handling invalid characters is always a problem in SAP NetWeaver BW and is a common cause for load failures. This tip describes a method you can use to check for invalid characters on text fields that are extracted from source systems in SAP NetWeaver BW. We'll show you how to create a custom function module to code the logic necessary to check a text field for invalid characters in SAP NetWeaver BW so that you can attempt to convert to a valid character prior to stopping the extraction due to an invalid character error.

✓ And Here's How ...

The ABAP logic discussed here will call a standard function module RSKC_CHAVL_OF_IOBJ_CHECK in SAP to check for invalid characters against permitted characters in Transaction RSKC and the standard character set allowed in SAP NetWeaver BW.

First, create a function module ZFIX_INVCHAR_CHAR with input parameters to pass the InfoObject name and the field content containing the text string that needs to be checked for invalid characters (see Figure 1). In this case, I_LOWE_CASE will contain the text string and I_INFOOBJECT will have the name of the InfoObject.



Function module ZFIX_INVCHAR_CHAR Inactive

Parameter Name	Typing	Associated Type	Default value	Opti...	Pas...	Short text	Lon...
I_INFOOBJECT	TYPE	RSD_IOBJNM		☐	☐	Infoobject Name	
I_INP_STRING	TYPE	ANY		☐	☐	Input field value	
I_LOWER_CASE	TYPE	C		✓	☐	Lower Case Enabled Infoobject	

Figure 1 Function Builder with Definition of Custom Function Module ZFIX_INVCHAR_CHAR

This function module, if passed by a string, should check for invalid characters and convert them to blanks if found. The function module will call SAP-delivered, standard function module RSKC_CHAVL_OF_IOBJ_CHECK to pass the entire string to check if any invalid characters are found. If it returns `sy-subrc NE 0`, then it means the text string contains an invalid character as defined in your SAP NetWeaver BW system, including any allowed characters maintained in Transaction RSKC.

If invalid characters are found, loop further through the content of the text field one character at a time and check for invalid characters. When found, replace them if needed using a valid character such as a blank or another special character that is identifiable as a character that replaced the invalid character. This helps to filter in a query or backend to produce a report for the source system to fix the data for reprocessing.

If the Lower Case Enabled Infoobject flag is checked for the InfoObject in RSD1, add the optional parameter `I_LOWER_CASE = 'X'` when calling this method to prevent the method from converting to uppercase. Otherwise, you don't need to include the optional parameter `I_LOWER_CASE` when calling this method. Use this method to check and convert invalid characters only on a `CHAR` field.

Optionally, you can also code logic to check if the entire string has a value of "#" or "!" and blank out the entire output string so that SAP NetWeaver BW doesn't flag this as an error.

The sample code to implement the preceding logic in function module `ZFIX_INVCHAR_CHAR` is as follows:

```
*"----------------------------------------------------------------
*"This method, if passed a string, will check for invalid
```

```
*"characters & convert to blanks if found. If lowercase flag is
*"checked for the InfoObject in RSD1, Add the optional parameter
*"I_LOWER_CASE = 'X' when calling this method to prevent the
*"method from converting to uppercase. Otherwise, you do not need
*"to include the optional parameter I_LOWER_CASE when calling
*"this method. Use this method to check and convert invalid
*"characters only on a CHAR field.
*"
*"*"Local Interface:
*"    IMPORTING
*"      REFERENCE(I_INFOOBJECT) TYPE RSD_IOBJNM
*"      REFERENCE(I_INP_STRING) TYPE ANY
*"      REFERENCE(I_LOWER_CASE) TYPE C OPTIONAL
*"    EXPORTING
        REFERENCE(O_OUT_STRING) TYPE ANY
*"-----------------------------------------------------------------

DATA:
     lv_length LIKE sy-Index,
     lv_char TYPE c,
     lv_index TYPE sy-index.

CLEAR : O_OUT_STRING.

        O_OUT_STRING = I_INP_STRING. "source field
* If Lower Case Enabled flag is set, do not convert the input
* field to upper case.
*

IF I_LOWER_CASE IS INITIAL.
   TRANSLATE O_OUT_STRING TO UPPER CASE.
ENDIF.

IF O_OUT_STRING = '#'.
  CLEAR O_OUT_STRING.
ELSEIF O_OUT_STRING(1) = '!'.
   O_OUT_STRING(1) = ' '.
ENDIF.
```

```
*
* Check the entire input field to see if it contains any invalid
* characters to see if further check needs to be done at a
* character level to replace them with blank.
* If no Invalid character is found, no further checks will be
* necessary.
*
IF sy-subrc NE 0.

* Since the above check determined that there may be one or more
* invalid characters in the input field, check individual
* characters in the input field to see if any of characters are
* invalid and replace them with blank.
*

    lv_length = STRLEN( O_OUT_STRING ).

    DO lv_length TIMES.

       lv_index = sy-index = 1.
       lv_char = O_OUT_STRING+lv_index(1).

       CALL FUNCTION 'RSKC_CHAVL_OF_IOBJ_CHECK'
         EXPORTING
           i_chavl   = O_OUT_STRING
           i_iobjnm  = I_INFOOBJECT
         EXCEPTIONS
          chavl_not_allowed = 1.
          OTHERS            = 2.

       IF sy-subrc NE 0.

          IF lv_char = '#' OR
                            lv_char = '!'.
                            O_OUT_STRING(1) = ' '.
                        ENDIF.
              ENDIF.
     ENDDO.
```

Tip 69

Deleting SAP NetWeaver BW Statistics for Better Statistic Query Performance

You can use two different methods to delete SAP NetWeaver BW statistics in your system to free up system space.

SAP NetWeaver BW statistics allow you to record runtime data for SAP NetWeaver BW processes and events in the BEx suite, Analytic Engine, and in the data warehousing functionality. The system records performance-intensive processing steps (known as statistics events). It calculates the net times of an event by calculating the runtime using the difference between the start and end times.

You can also use the statistics to record the monitoring status of SAP NetWeaver BW objects from the Data Warehouse; however, query runtime statistics in particular generate a large amount of statistics data in the SAP NetWeaver BW statistics tables, so it is recommended to delete the data from these tables on a regular basis in order to free up system space.

This tip will describe an SAP standard functionality that can be used to delete SAP NetWeaver BW statistics.

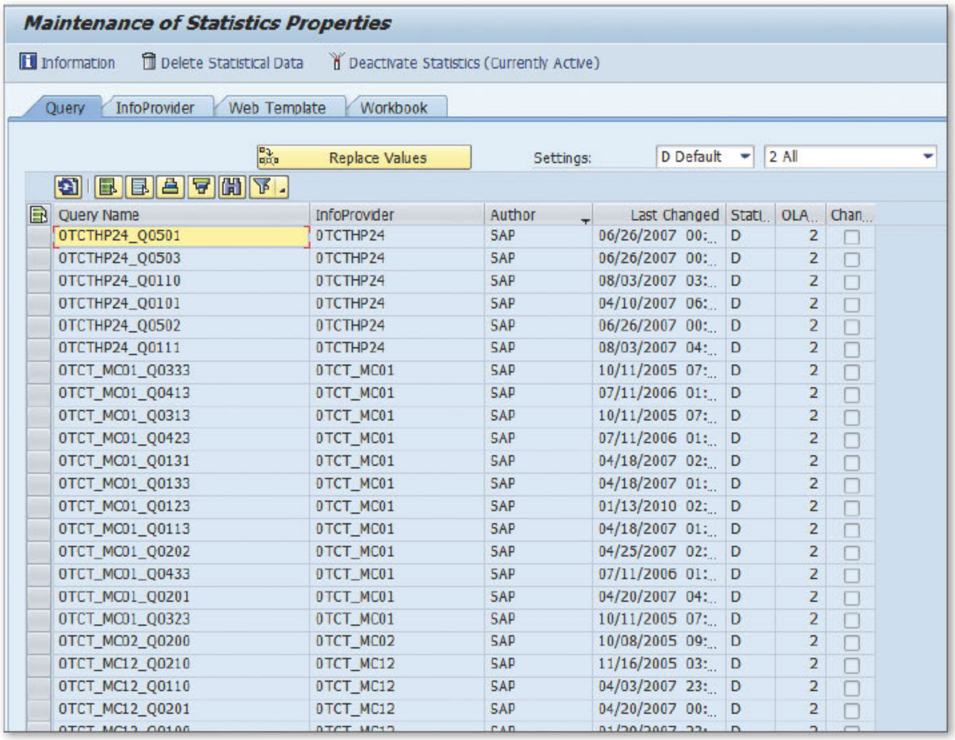
✓ And Here's How ...

There are two methods you can use for deletion, which we'll explain in the following sections.

SAP Standard Transaction RSDDSTAT

After you're logged in to the SAP NetWeaver BW system, follow these steps:

1. Execute Transaction RSDDSTAT in the Administrator Workbench. This will display the screen shown in Figure 1.



Maintenance of Statistics Properties

Information | Delete Statistical Data | Deactivate Statistics (Currently Active)

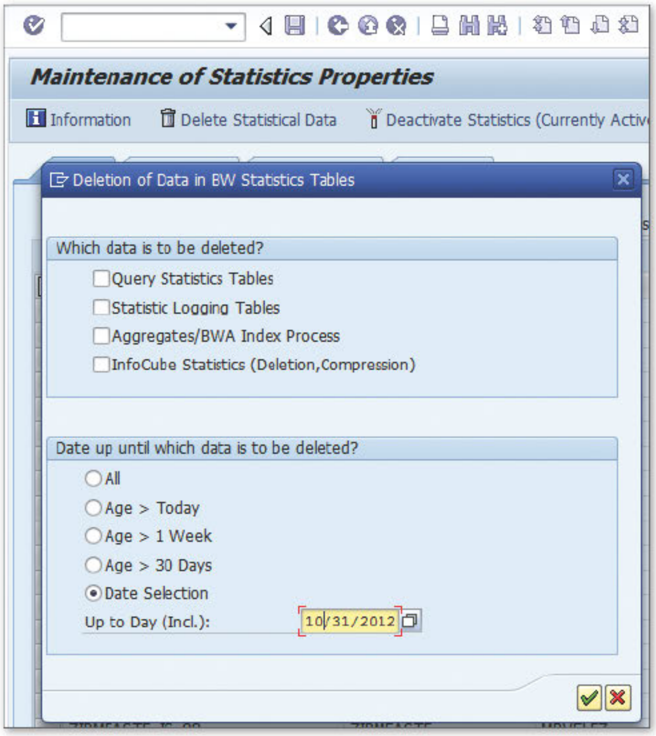
Query | InfoProvider | Web Template | Workbook

Replace Values — Settings: D Default — 2 All

Query Name	InfoProvider	Author	Last Changed	Stati...	OLA...	Chan...
0TCTHP24_Q0501	0TCTHP24	SAP	06/26/2007 00:...	D	2	☐
0TCTHP24_Q0503	0TCTHP24	SAP	06/26/2007 00:...	D	2	☐
0TCTHP24_Q0110	0TCTHP24	SAP	08/03/2007 03:...	D	2	☐
0TCTHP24_Q0101	0TCTHP24	SAP	04/10/2007 06:...	D	2	☐
0TCTHP24_Q0502	0TCTHP24	SAP	06/26/2007 00:...	D	2	☐
0TCTHP24_Q0111	0TCTHP24	SAP	08/03/2007 04:...	D	2	☐
0TCT_MC01_Q0333	0TCT_MC01	SAP	10/11/2005 07:...	D	2	☐
0TCT_MC01_Q0413	0TCT_MC01	SAP	07/11/2006 01:...	D	2	☐
0TCT_MC01_Q0313	0TCT_MC01	SAP	10/11/2005 07:...	D	2	☐
0TCT_MC01_Q0423	0TCT_MC01	SAP	07/11/2006 01:...	D	2	☐
0TCT_MC01_Q0131	0TCT_MC01	SAP	04/18/2007 02:...	D	2	☐
0TCT_MC01_Q0133	0TCT_MC01	SAP	04/18/2007 01:...	D	2	☐
0TCT_MC01_Q0123	0TCT_MC01	SAP	01/13/2010 02:...	D	2	☐
0TCT_MC01_Q0113	0TCT_MC01	SAP	04/18/2007 01:...	D	2	☐
0TCT_MC01_Q0202	0TCT_MC01	SAP	04/25/2007 02:...	D	2	☐
0TCT_MC01_Q0433	0TCT_MC01	SAP	07/11/2006 01:...	D	2	☐
0TCT_MC01_Q0201	0TCT_MC01	SAP	04/20/2007 04:...	D	2	☐
0TCT_MC01_Q0323	0TCT_MC01	SAP	10/11/2005 07:...	D	2	☐
0TCT_MC02_Q0200	0TCT_MC02	SAP	10/08/2005 09:...	D	2	☐
0TCT_MC12_Q0210	0TCT_MC12	SAP	11/16/2005 03:...	D	2	☐
0TCT_MC12_Q0110	0TCT_MC12	SAP	04/03/2007 23:...	D	2	☐
0TCT_MC12_Q0201	0TCT_MC12	SAP	04/20/2007 00:...	D	2	☐
0TCT_MC12_Q0100	0TCT_MC12	SAP	01/20/2007 23:...	D	2	☐

Figure 1 RSDDSTAT Initial Screen

2. Click on the Delete Statistical Data button at the top of the screen, and the popup shown in Figure 2 will appear.


Maintenance of Statistics Properties

Information | Delete Statistical Data | Deactivate Statistics (Currently Activ

Deletion of Data in BW Statistics Tables

Which data is to be deleted?

- ☐ Query Statistics Tables
- ☐ Statistic Logging Tables
- ☐ Aggregates/BWA Index Process
- ☐ InfoCube Statistics (Deletion,Compression)

Date up until which data is to be deleted?

- ○ All
- ○ Age > Today
- ○ Age > 1 Week
- ○ Age > 30 Days
- ● Date Selection

Up to Day (Incl.): 10/31/2012

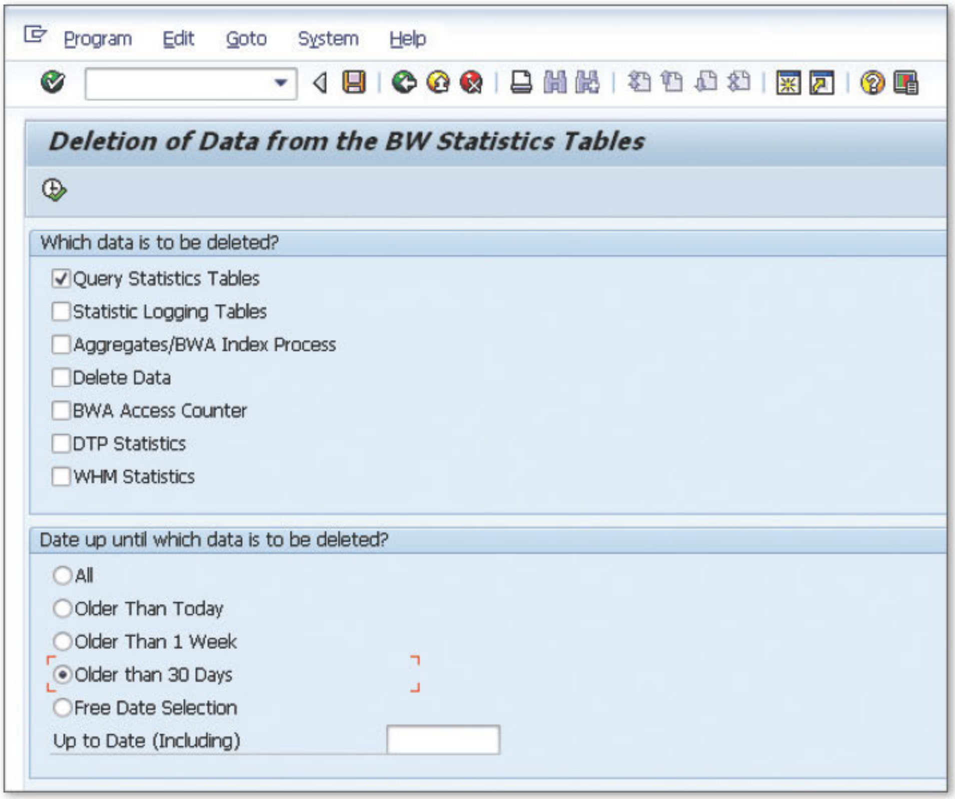
Figure 2 Specifying Date for Deletion of Statistics Data

3. Select the check box for which you want to delete the data, and select the radio button to specify the date until which the data is to be deleted. Then press the CONTINUE button.
4. Save the settings by clicking the SAVE button at the top of the screen.

The system will delete the data from the SAP NetWeaver BW statistics tables for the selection.

SAP Program RSDDSTAT_DATA_DELETE

1. Execute Transaction SE38 in the SAP NetWeaver BW system.
2. Enter the program name RSDDSTAT_DATA_DELETE and click EXECUTE. The resulting screen is displayed as shown in Figure 3.



Deletion of Data from the BW Statistics Tables

Which data is to be deleted?

- [x] Query Statistics Tables
- [ ] Statistic Logging Tables
- [ ] Aggregates/BWA Index Process
- [ ] Delete Data
- [ ] BWA Access Counter
- [ ] DTP Statistics
- [ ] WHM Statistics

Date up until which data is to be deleted?

○ All
○ Older Than Today
○ Older Than 1 Week
● Older than 30 Days
○ Free Date Selection

Up to Date (Including)

Figure 3 Specifying Criteria for Deletion of Statistics Data

3. Select the check box in the WHICH DATA IS TO BE DELETED? area and DATE UP UNTIL WHICH DATA IS TO BE DELETED? and click EXECUTE.

The system will delete the data from the SAP NetWeaver BW statistics tables for the selection.

Tip 70

Analyzing InfoProviders in Detail in SAP NetWeaver BW

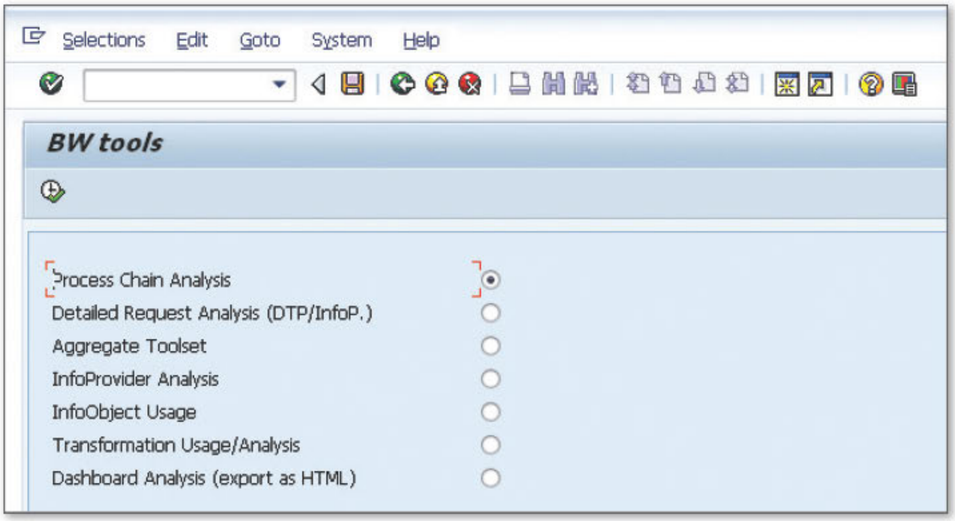
You can use an SAP delivered standard program to perform extensive analysis of any SAP NetWeaver BW InfoProvider.

In an SAP NetWeaver BW project or support environment, you'll usually need to perform an advanced performance analysis on InfoProviders. This type of analysis is very helpful in developing an efficient model for the InfoProvider during the development phase. Alternatively, this method will be a very helpful tool for troubleshooting performance issues when you're reporting on the InfoProvider.

This tip will discuss the different selection options that are available in the standard SAP delivered program /SSA/BWT, and explain how to use the results to analyze the InfoProvider in detail.

✓ And Here's How ...

To access the SAP program, first call up Transaction SE38. Specify the program /SSA/BWT and click on EXECUTE. The screen shown in Figure 1 will be displayed.



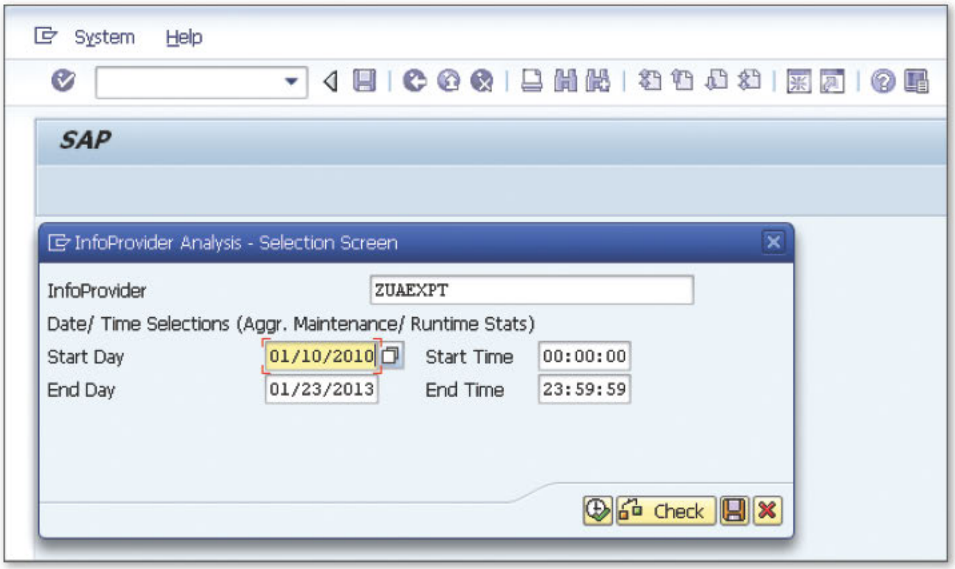
Process Chain Analysis [✓]
Detailed Request Analysis (DTP/InfoP.)
Aggregate Toolset
InfoProvider Analysis
InfoObject Usage
Transformation Usage/Analysis
Dashboard Analysis (export as HTML)

Figure 1 /SSA/BWT Screen

The following options are available in the screen:

- Process Chain Analysis
  Allows for the analysis of process chains either at the process chain level or process type level.
- Detailed Request Analysis
  Allows for analysis by a specific request. The technical name of the request ID is needed, which can be found from the InfoProvider/DSO manage screen.
- Aggregate Toolset
  Allows for the analysis of the aggregates associated with a particular InfoCube.
- Template Report Analysis
  Allows for the analysis of specific workbooks/queries.
- InfoProvider Analysis
  Provides details of an Infocube (e.g., information related to InfoCube design, SID tables and NRIV information, databases indexes, etc.).

The default selection is Process Chain Analysis. For the purpose of this tip, select InfoProvider Analysis and choose Execute to see the screen shown in Figure 2. Enter the InfoProvider for which the analysis needs to be performed, the date range, and then click Execute.

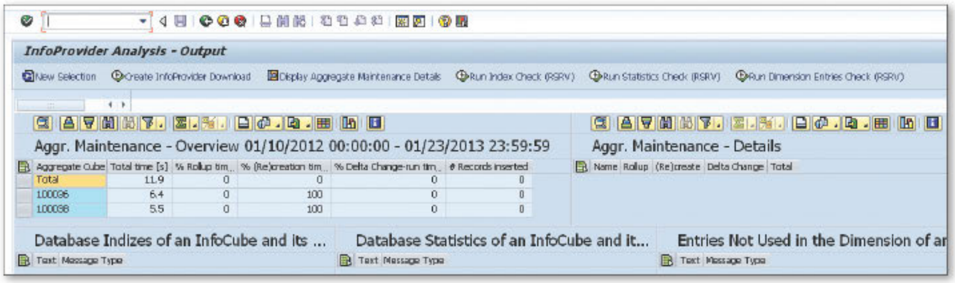


InfoProvider: ZUAEXPT
Date/ Time Selections (Aggr. Maintenance/ Runtime Stats)
Start Day: 01/10/2010 Start Time: 00:00:00
End Day: 01/23/2013 End Time: 23:59:59

Figure 2 Selection Screen for InfoProvider Analysis

You'll now see the screen shown in Figure 3 with different options:

- New Selection
  This button allows you to change the InfoProvider selection.
- Created InfoProvider Download
  Creates a GUID-based download for InfoProvider tables.
- Display Aggregate Maintenance Details
  Displays aggregate maintenance deatils about the InfoCube.
- Run Index Check(RSRV)
  Displays database indexes for an InfoCube and its aggregates.
- Run Statistics check(RSRV)
  Displays database statistics for an InfoCube and its aggregates.
- Run Dimension Entries Check(RSRV)
  Displays entries that aren't used in dimensions of an InfoCube.


InfoProvider Analysis - Output
New Selection
Create InfoProvider Download
Display Aggregate Maintenance Details
Run Index Check (RSRV)
Run Statistics Check (RSRV)
Run Dimension Entries Check (RSRV)

Aggr. Maintenance - Overview 01/10/2012 00:00:00 - 01/23/2013 23:59:59

Aggregate Cube	Total time [s]	% Rollup tim..	% (Re)creation tim..	% Delta Change-run tim..	# Records inserted
Total	11.9	0	0	0	0
100036	6.4	0	100	0	0
100038	5.5	0	100	0	0

Aggr. Maintenance - Details

Name	Rollup	(Re)create	Delta Change	Total

Database Indizes of an InfoCube and its ...

Text	Message Type

Database Statistics of an InfoCube and it...

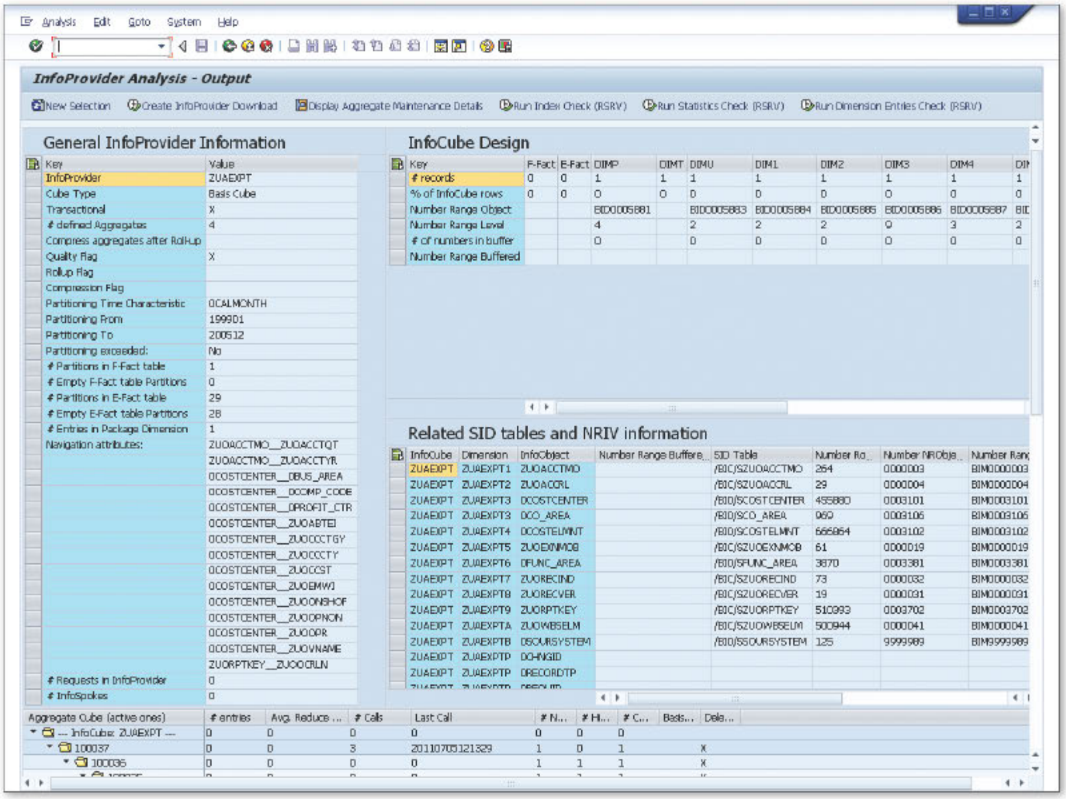
Text	Message Type

Entries Not Used in the Dimension of ar

Text	Message Type

Figure 3 InfoProvider Analysis: Output

The next section of the screen will provide you with detail analysis of the InfoProvider as shown in Figure 4.


Analysis Edit Goto System Help
InfoProvider Analysis - Output
New Selection
Create InfoProvider Download
Display Aggregate Maintenance Details
Run Index Check (RSRV)
Run Statistics Check (RSRV)
Run Dimension Entries Check (RSRV)

General InfoProvider Information

Key	Value
InfoProvider	ZUAEXPT
Cube Type	Basis Cube
Transactional	X
# defined Aggregates	4
Compress aggregates after Rollup	
Quality Flag	X
Rollup Flag	
Compression Flag	
Partitioning Time Characteristic	0CALMONTH
Partitioning From	199901
Partitioning To	200512
Partitioning exceeded:	No
# Partitions in F-Fact table	1
# Empty F-Fact table Partitions	0
# Partitions in E-Fact table	29
# Empty E-Fact table Partitions	28
# Entries in Package Dimension	1
Navigation attributes:	ZUOACCTMO__ZUOACCTQT
	ZUOACCTMO__ZUOACCTYR
	0COSTCENTER__0BUS_AREA
	0COSTCENTER__0COMP_CODE
	0COSTCENTER__0PROFIT_CTR
	0COSTCENTER__ZUOABTEI
	0COSTCENTER__ZUOCCCTGY
	0COSTCENTER__ZUOCCCTY
	0COSTCENTER__ZUOCCST
	0COSTCENTER__ZUOEMWI
	0COSTCENTER__ZUOONSHOF
	0COSTCENTER__ZUOOPNON
	0COSTCENTER__ZUOOPR
	0COSTCENTER__ZUOVNAME
	ZUORPTKEY__ZUOOCRLN
# Requests in InfoProvider	0
# InfoSpokes	0

InfoCube Design

Key	F-Fact	E-Fact	DIMP	DIMT	DIMU	DIM1	DIM2	DIM3	DIM4	DI
# records	0	0	1	1	1	1	1	1	1	1
% of InfoCube rows	0	0	0	0	0	0	0	0	0	0
Number Range Object			BID0005881		BID0005883	BID0005884	BID0005885	BID0005886	BID0005887	BID
Number Range Level			4		2	2	2	9	3	2
# of numbers in buffer			0		0	0	0	0	0	0
Number Range Buffered										

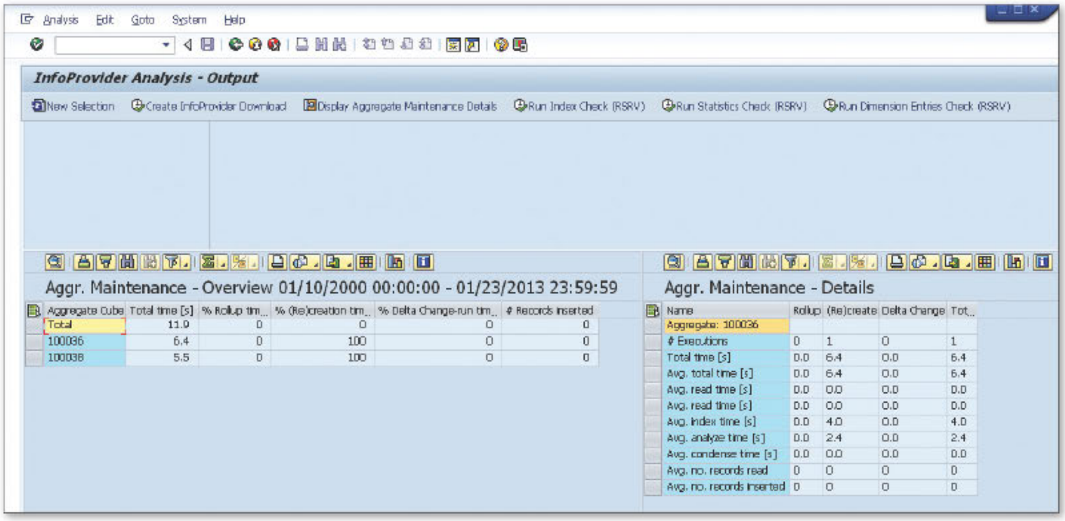
Related SID tables and NRIV information

InfoCube	Dimension	InfoObject	Number Range Buffere..	SID Table	Number Ro..	Number NRObje..	Number Rang
ZUAEXPT	ZUAEXPT1	ZUOACCTMO		/BIC/SZUOACCTMO	264	0000003	BIM0000003
ZUAEXPT	ZUAEXPT2	ZUOACCRL		/BIC/SZUOACCRL	29	0000004	BIM0000004
ZUAEXPT	ZUAEXPT3	0COSTCENTER		/BI0/SCOSTCENTER	455880	0003101	BIM0003101
ZUAEXPT	ZUAEXPT3	0CO_AREA		/BI0/SCO_AREA	969	0003106	BIM0003106
ZUAEXPT	ZUAEXPT4	0COSTELMNT		/BI0/SCOSTELMNT	666864	0003102	BIM0003102
ZUAEXPT	ZUAEXPT5	ZUOEXNMOB		/BIC/SZUOEXNMOB	61	0000019	BIM0000019
ZUAEXPT	ZUAEXPT6	0FUNC_AREA		/BI0/SFUNC_AREA	3870	0003381	BIM0003381
ZUAEXPT	ZUAEXPT7	ZUORECIND		/BIC/SZUORECIND	73	0000032	BIM0000032
ZUAEXPT	ZUAEXPT8	ZUORECVER		/BIC/SZUORECVER	19	0000031	BIM0000031
ZUAEXPT	ZUAEXPT9	ZUORPTKEY		/BIC/SZUORPTKEY	510993	0003702	BIM0003702
ZUAEXPT	ZUAEXPTA	ZUOWBSELM		/BIC/SZUOWBSELM	500944	0000041	BIM0000041
ZUAEXPT	ZUAEXPTB	0SOURSYSTEM		/BI0/SSOURSYSTEM	125	9999989	BIM9999989
ZUAEXPT	ZUAEXPTP	0CHNGID					
ZUAEXPT	ZUAEXPTP	0RECORDTP					

Aggregate Cube (active ones)	# entries	Avg. Reduce ...	# Calls	Last Call	# N...	# H...	# C...	Basis...	Dele...
--- InfoCube: ZUAEXPT ---	0	0	0	0	0	0	0		
100037	0	0	3	20110705121329	1	0	1		X
100036	0	0	0	0	1	1	1		X

Figure 4 General InfoProvider Information

- GENERAL INFOPROVIDER INFORMATION
  This section provides you the InfoProvider details such as cube type, patitioned InfoObject, navigation attributes used in the InfoCube, and InfoSpokes created.
- INFOCUBE DESIGN
  This section provides you information like the number range object of InfoCube dimensions.
- RELATED SID TABLES AND NRIV INFORMATION
  This section provides SID table and number range information of the InfoObjects that are used in the InfoCube.
- AGGR. MAINTENANCE-OVERVIEW
  This section provides overviews of the aggregates that are created in the InfoCube.
- AGGR.MAINTENANCE DETAILS
  This section provides the details of the aggregates. To display the details, select the row above the screen and click the DISPLAY AGGREGATE MAINTENANCE DETAILS icon. Deatils of the aggregate are displayed as shown in Figure 5.


Analysis
Edit
Goto
System
Help
InfoProvider Analysis - Output
New Selection
Create InfoProvider Download
Display Aggregate Maintenance Details
Run Index Check (RSRV)
Run Statistics Check (RSRV)
Run Dimension Entries Check (RSRV)

Aggr. Maintenance - Overview 01/10/2000 00:00:00 - 01/23/2013 23:59:59

Aggregate Cube	Total time [s]	% Rollup tim...	% (Re)creation tim...	% Delta Change-run tim...	# Records inserted
Total	11.9	0	0	0	0
100036	6.4	0	100	0	0
100038	5.5	0	100	0	0

Aggr. Maintenance - Details

Name	Rollup	(Re)create	Delta Change	Tot...
Aggregate: 100036				
# Executions	0	1	0	1
Total time [s]	0.0	6.4	0.0	6.4
Avg. total time [s]	0.0	6.4	0.0	6.4
Avg. read time [s]	0.0	0.0	0.0	0.0
Avg. read time [s]	0.0	0.0	0.0	0.0
Avg. index time [s]	0.0	4.0	0.0	4.0
Avg. analyze time [s]	0.0	2.4	0.0	2.4
Avg. condense time [s]	0.0	0.0	0.0	0.0
Avg. no. records read	0	0	0	0
Avg. no. records inserted	0	0	0	0

Figure 5 Aggregate Information

Tip 71

Setting Up a Transport Request to Delete Obsolete SAP NetWeaver BW Objects from Inaccessible QA and PRD Systems

If you have obsolete objects that still exist in the production and quality systems, but you only have access to the development system, you can delete these objects with a transport request.

In large SAP NetWeaver BW implementations, it's common for a few obsolete objects to be present in the production (PRD) and quality (QA) systems but be missing from the development (DEV) system. No developer will have edit access in QA and PRD, so to delete those obsolete objects, you need to send a transport request from DEV.

This tip describes a simple method you can use to delete the obsolete objects.

✓ And Here's How ...

To set up the transport request to delete the obsolete objects, follow these steps:

1. Go to Transaction SE01, and create the transport request by clicking the CREATE icon (see Figure 1).

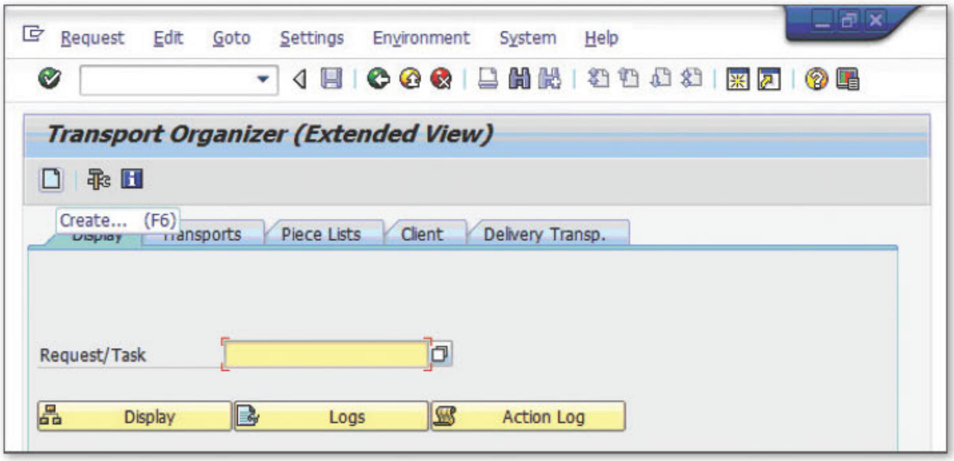
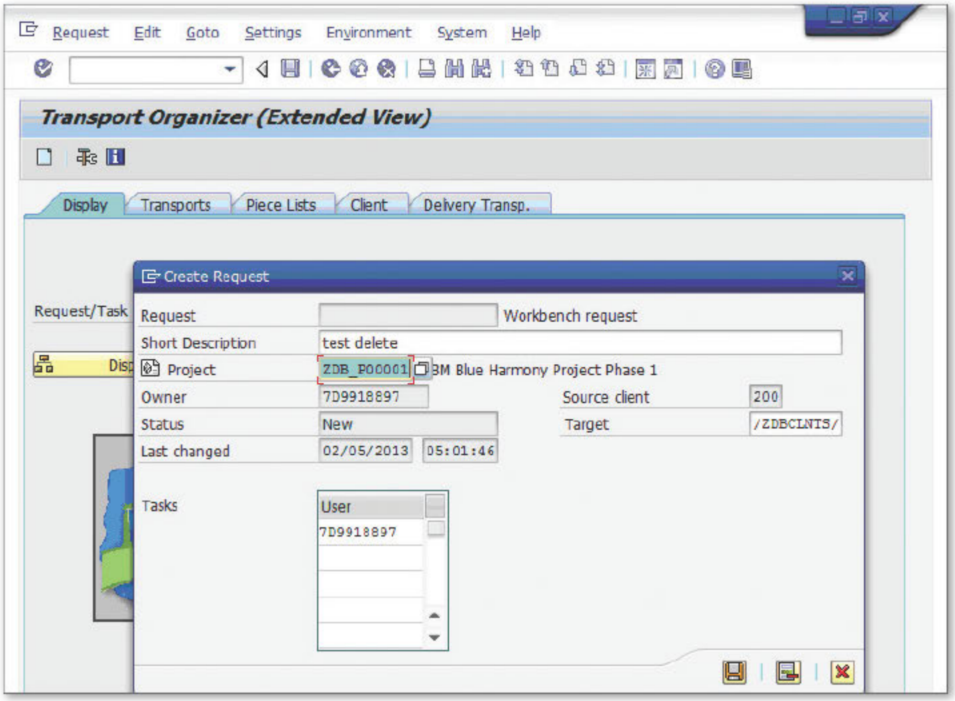

Request Edit Goto Settings Environment System Help
Transport Organizer (Extended View)
Create... (F6)
Display Transports Piece Lists Client Delivery Transp.
Request/Task
Display
Logs
Action Log

Figure 1 Transaction SE01 Transport Organizer

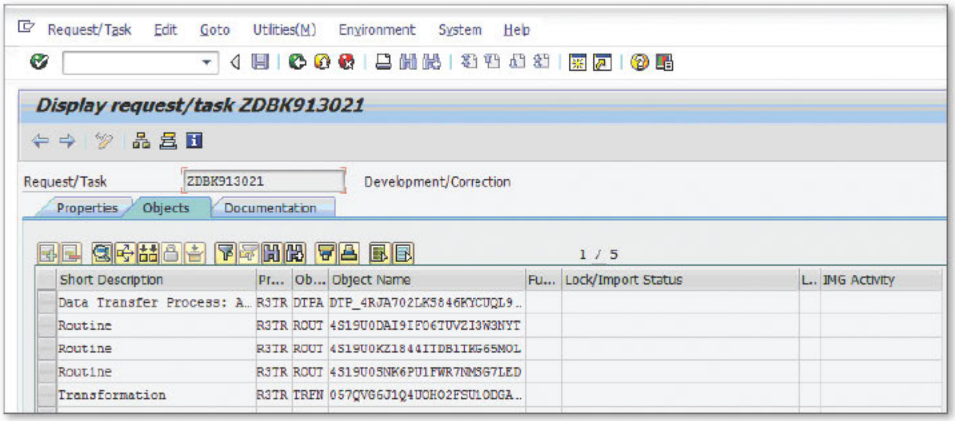
2. Enter the necessary information in the following screen for identifying the obsolete objects (see Figure 2).


Request Edit Goto Settings Environment System Help
Transport Organizer (Extended View)
Display Transports Piece Lists Client Delivery Transp.
Request/Task
Create Request

Request | Workbench request
Short Description: test delete
Project: ZDB_P00001 BM Blue Harmony Project Phase 1
Owner: 7D9918897 | Source client: 200
Status: New | Target: /ZDBCLNTS/
Last changed: 02/05/2013 05:01:46
Tasks: User 7D9918897

Figure 2 Creating a Request to Delete Obsolete Objects in QA and PROD

3. Enter the missing objects information as shown in Figure 3 (the objects you want to delete from the other systems). The system automatically identifies that these objects are missing and need to be deleted in QA and PRD. After the transport reaches the destination, the obsolete objects will be deleted.



Request/Task ZDBK913021 Development/Correction

1 / 5

Short Description	Pr...	Ob...	Object Name	Fu...	Lock/Import Status	L..	IMG Activity
Data Transfer Process: A...	R3TR	DTPA	DTP_4RJA702LK5846KYCUQL9...				
Routine	R3TR	ROUT	4S19U0DAI9IF06TUVZI3W3NYT				
Routine	R3TR	ROUT	4S19U0KZ1844IIDB1IKG65MOL				
Routine	R3TR	ROUT	4S19U05NK6PU1FWR7NMSG7LED				
Transformation	R3TR	TRFN	057QVG6J1Q4UOHO2FSU1ODGA...				

Figure 3 Transport Request Showing the Objects That Will Be Deleted

Tip 72

Restoring an ABAP Program to an Earlier Working Version to Correct a Bug

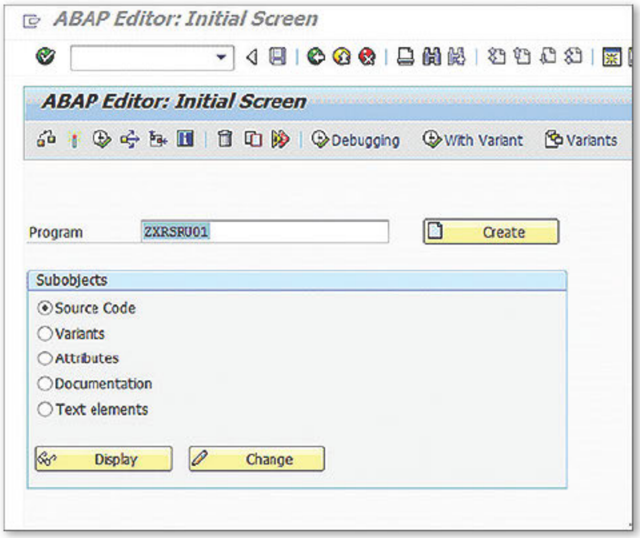
If you notice inconsistencies in the current version of an ABAP program during an implementation, you can revert the ABAP code changes back to an earlier version.

In large SAP NetWeaver BW implementations, it's common for the latest ABAP code changes in the ABAP program to cause bugs in the existing essential functionality. These changes can be revisions made by the developers or errors introduced by an upgrade. Sometimes, it's hard to find out what caused the issue in a short period of time. As a quick bug-fixing mode in production, you can easily revert the code changes that have errors in the ABAP program.

✓ And Here's How ...

To revert the ABAP code changes with a simple fix, follow these steps:

1. Go to Transaction SE38, and enter the ABAP program with errors that is recently pushed to the production system. In this tip, we'll use the user exit program "ZXRSRU01" (see Figure 1), but you may use any program that needs a fix.

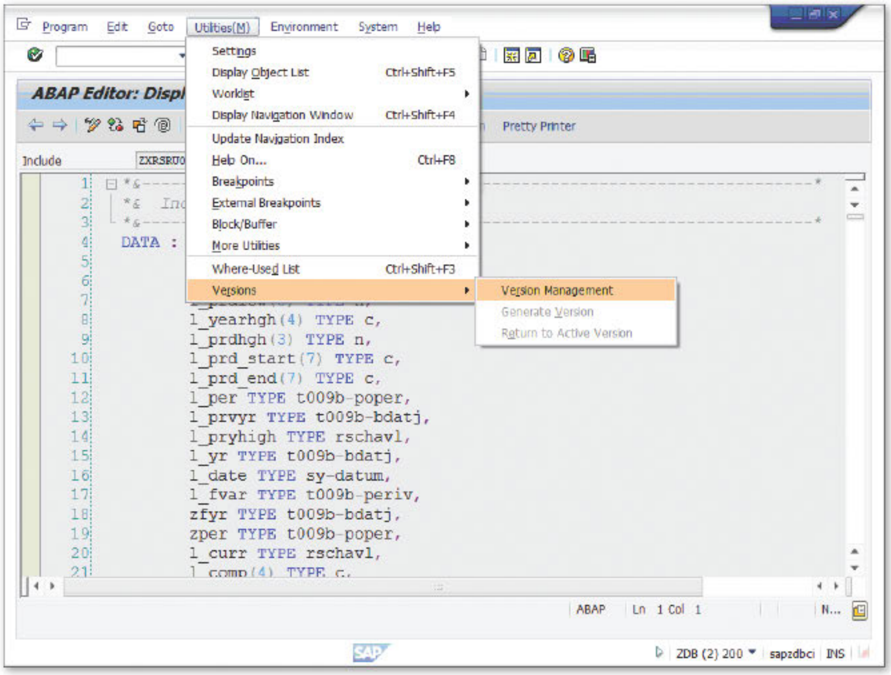


Program ZXRSRU01

Subobjects
- [✓ Source Code]
- Variants
- Attributes
- Documentation
- Text elements

« *Figure 1 Transaction SE38: Enter Program Name*

2. On the resulting screen (see Figure 2), click the DISPLAY button, and choose VERSIONS • VERSION MANAGEMENT.

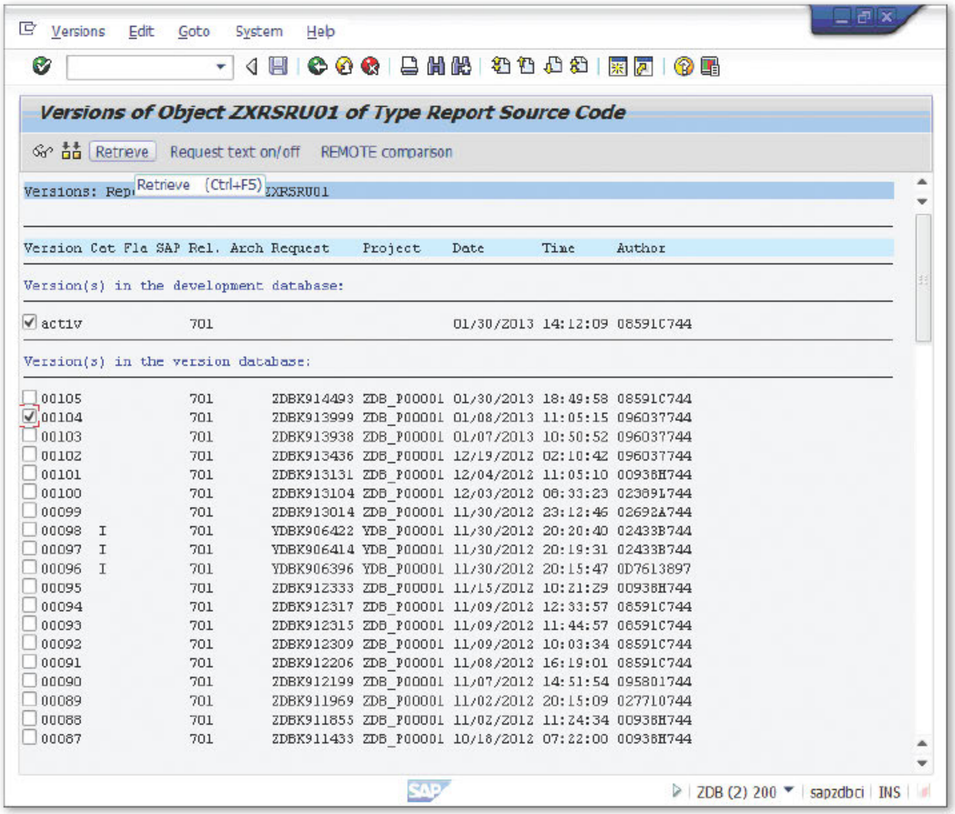


Include ZXRSRU0

```
*&----
*&   Inc
*&----
DATA :
         l_yearhgh(4) TYPE c,
         l_prdhgh(3) TYPE n,
         l_prd_start(7) TYPE c,
         l_prd_end(7) TYPE c,
         l_per TYPE t009b-poper,
         l_prvyr TYPE t009b-bdatj,
         l_pryhigh TYPE rschavl,
         l_yr TYPE t009b-bdatj,
         l_date TYPE sy-datum,
         l_fvar TYPE t009b-periv,
         zfyr TYPE t009b-bdatj,
         zper TYPE t009b-poper,
         l_curr TYPE rschavl,
         l_comp(4) TYPE c.
```

Figure 2 Selecting Version Management on the Display Program Screen

The resulting screen (Figure 3) shows all of the versions of the program. You can choose the version that you want to revert back to by checking the checkbox near the version and then clicking RETRIEVE.



Versions of Object ZXRSRU01 of Type Report Source Code

```
Versions: Rep        ZXRSRU01

Version Cat Fla SAP Rel. Arch Request    Project     Date       Time     Author

Version(s) in the development database:

✓ activ             701                              01/30/2013 14:12:09 08591C744

Version(s) in the version database:

☐ 00105             701       ZDBK914493 ZDB_P00001 01/30/2013 18:49:58 08591C744
✓ 00104             701       ZDBK913999 ZDB_P00001 01/08/2013 11:05:15 096037744
☐ 00103             701       ZDBK913938 ZDB_P00001 01/07/2013 10:50:52 096037744
☐ 00102             701       ZDBK913436 ZDB_P00001 12/19/2012 02:10:42 096037744
☐ 00101             701       ZDBK913131 ZDB_P00001 12/04/2012 11:05:10 00938H744
☐ 00100             701       ZDBK913104 ZDB_P00001 12/03/2012 08:33:23 02389L744
☐ 00099             701       ZDBK913014 ZDB_P00001 11/30/2012 23:12:46 02692A744
☐ 00098  I          701       YDBK906422 YDB_P00001 11/30/2012 20:20:40 02433B744
☐ 00097  I          701       YDBK906414 YDB_P00001 11/30/2012 20:19:31 02433B744
☐ 00096  I          701       YDBK906396 YDB_P00001 11/30/2012 20:15:47 0D7613897
☐ 00095             701       ZDBK912333 ZDB_P00001 11/15/2012 10:21:29 00938H744
☐ 00094             701       ZDBK912317 ZDB_P00001 11/09/2012 12:33:57 08591C744
☐ 00093             701       ZDBK912315 ZDB_P00001 11/09/2012 11:44:57 08591C744
☐ 00092             701       ZDBK912309 ZDB_P00001 11/09/2012 10:03:34 08591C744
☐ 00091             701       ZDBK912206 ZDB_P00001 11/08/2012 16:19:01 08591C744
☐ 00090             701       ZDBK912199 ZDB_P00001 11/07/2012 14:51:54 095801744
☐ 00089             701       ZDBK911969 ZDB_P00001 11/02/2012 20:15:09 027710744
☐ 00088             701       ZDBK911855 ZDB_P00001 11/02/2012 11:24:34 00938H744
☐ 00087             701       ZDBK911433 ZDB_P00001 10/18/2012 07:22:00 00938H744
```

Figure 3 Different Versions of the Program in the System

You can also do a remote comparison of the program by clicking on REMOTE COMPARISON button. This compares the code between the two systems and highlights the differences.

This process can be used to revert back to any version of the program.

Tip 73

Checking the Patch Level of Different SAP NetWeaver BW Components

If you've encountered an issue when working with SAP NetWeaver BW, you can find the patch level of the system to know what to install.

A user may encounter technical issues when working with the SAP NetWeaver BW system. These issues might stem from an application-related error, or they might have roots in the version (read patch) of one of the SAP NetWeaver BW software components.

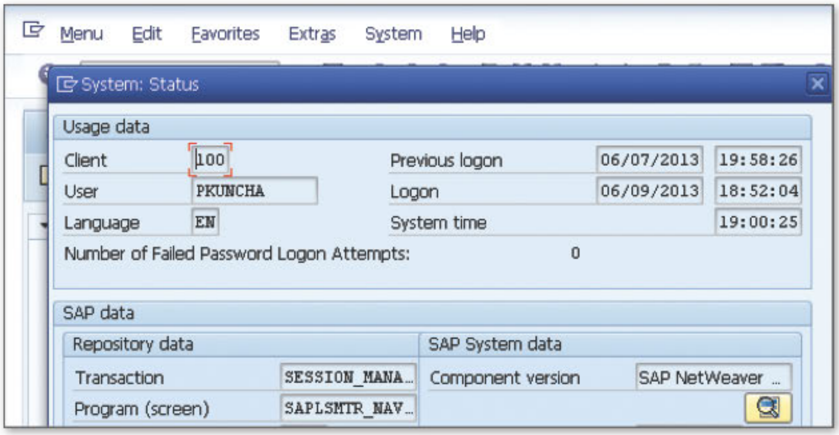
From time to time, SAP releases fixes to bugs in an earlier version. Sometimes bugs are resolved by applying a specific change, and at other times an upgrade to a different patch level is required. Also if a bug is discovered and an SAP Note has to be created, SAP typically requires the patch level of the component to be specified when creating the SAP Note. Hence, it's important to understand the patch level of the different components of the SAP NetWeaver BW system.

This tip describes how to check the patch level for the different components installed in SAP NetWeaver BW.

✓ And Here's How ...

SAP provides a simple way to determine the patch level of the SAP NetWeaver BW component. Follow these steps:

1. Log on to the SAP NetWeaver BW system. In the ensuing screen, click on SYSTEM • STATUS.
2. On the resulting screen, click on the COMPONENT INFORMATION' button (see Figure 1).



Usage data

Client	100	Previous logon	06/07/2013	19:58:26
User	PKUNCHA	Logon	06/09/2013	18:52:04
Language	EN	System time		19:00:25

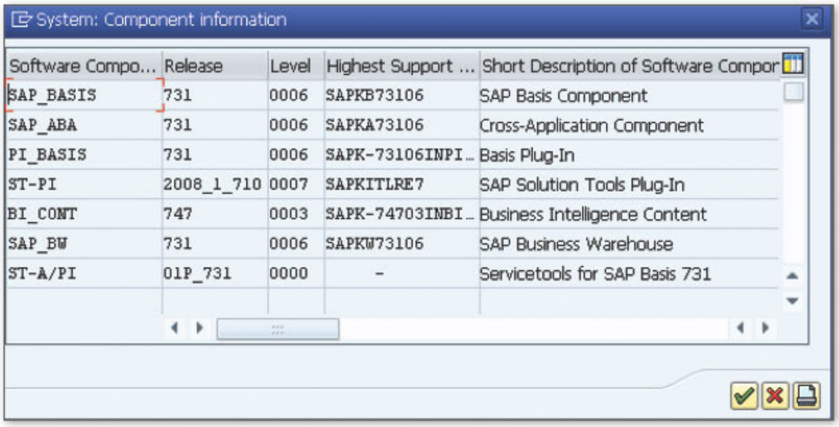
Number of Failed Password Logon Attempts: 0

SAP data

Repository data		SAP System data	
Transaction	SESSION_MANA...	Component version	SAP NetWeaver ...
Program (screen)	SAPLSMTR_NAV...		

Figure 1 Component Information

3. A popup screen will appear that shows the status and patch level of the different components installed (see Figure 2).



Software Compo...	Release	Level	Highest Support ...	Short Description of Software Compor
SAP_BASIS	731	0006	SAPKB73106	SAP Basis Component
SAP_ABA	731	0006	SAPKA73106	Cross-Application Component
PI_BASIS	731	0006	SAPK-73106INPI...	Basis Plug-In
ST-PI	2008_1_710	0007	SAPKITLRE7	SAP Solution Tools Plug-In
BI_CONT	747	0003	SAPK-74703INBI...	Business Intelligence Content
SAP_BW	731	0006	SAPKW73106	SAP Business Warehouse
ST-A/PI	01P_731	0000	-	Servicetools for SAP Basis 731

Figure 2 Patch Levels of Different Components

Now you can use the information on the screen to search for SAP Notes or even communicate with the SAP technical support team to troubleshoot technical system issues.

Tip 74

Finding and Deleting Unused Query Elements Using Selection Conditions

You can improve system performance by searching for and deleting unused queries, query elements, and dependent objects in your SAP NetWeaver BW system.

In a typical SAP NetWeaver BW environment, the number of queries, query elements, and dependent objects (workbooks, etc.) heavily increase over time. It's usually difficult to keep track of the objects that are actually being used.

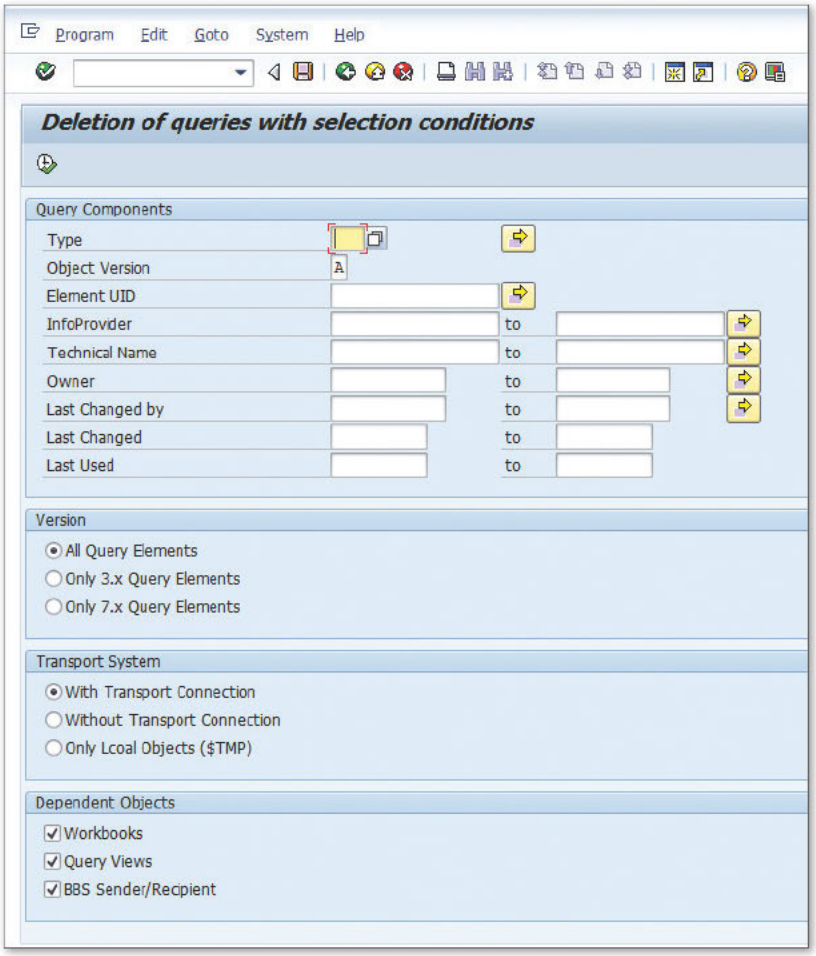
Having a large number of unused objects is a drain on the system and can have an adverse impact on any system-related activities such as conversions, upgrades, and so on. This can become a problem in all of the systems in the landscape: development, acceptance, and production. One solution is to enforce discipline on the users to delete unused queries; however, implementing this approach isn't practical.

This tip describes an SAP standard functionality that can be used to delete unused queries, query elements, and dependent objects such as workbooks, query views, and so on.

✓ And Here's How ...

SAP standard Transaction RSZDELETE enables you to perform an advanced where-used check on queries, query elements, workbooks, and other dependent objects such as query views. It also allows you to delete the unused query objects from the system based on a rich set of selection options. Follow these steps to use the transaction:

1. Execute Transaction RSZDELETE in the Administration Workbench to open the screen shown in Figure 1.



Deletion of queries with selection conditions

Query Components

Type			
Object Version	A		
Element UID			
InfoProvider		to	
Technical Name		to	
Owner		to	
Last Changed by		to	
Last Changed		to	
Last Used		to	

Version

● All Query Elements
○ Only 3.x Query Elements
○ Only 7.x Query Elements

Transport System

● With Transport Connection
○ Without Transport Connection
○ Only Lcoal Objects ($TMP)

Dependent Objects

✓ Workbooks
✓ Query Views
✓ BBS Sender/Recipient

Figure 1 Transaction RSZDELETE Initial Screen

Here, you have four main options for providing selection conditions:

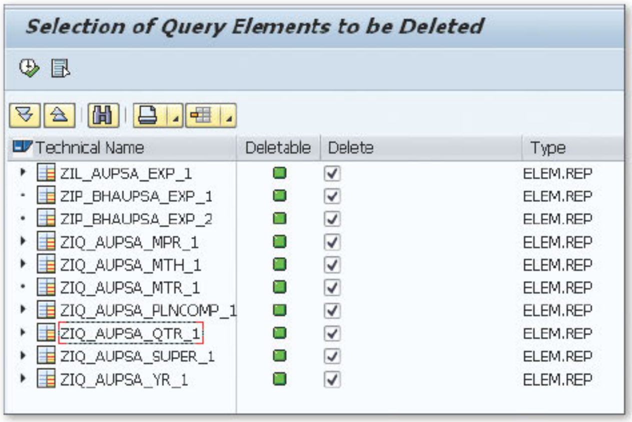
- Query Components
  Define the query components to be selected. Using the Type dropdown box, you can specify the type of query elements (Query, Filter, Structure,

RESTRICTED KEY FIGURE, CALCULATED KEY FIGURE, and VARIABLE) that you want to select. Alternatively, you may select the query components using other options such as the INFOPROVIDER, OWNER, LAST USED, and so on.

- VERSION
  Specify whether you need to select only version-specific query elements 3.x, 7.x, or all versions.
- TRANSPORT SYSTEM
  Specify whether the changes (deletions) need to be transported to another system.
- DEPENDENT OBJECTS
  Specify the type of dependent objects you want to include in the selection.

Where necessary, restrict your search further; for example, to a certain Info-Provider, to certain technical names, or to the last person to make the change.

2. After making all of the selections, click on the EXECUTE button. The next screen displays all of the query and query elements based on the selection conditions specified in the earlier screen. This screen also has a checkbox for selecting the objects to be deleted (see Figure 2).



Selection of Query Elements to be Deleted

Technical Name	Deletable	Delete	Type
ZIL_AUPSA_EXP_1	🟩	✓	ELEM.REP
ZIP_BHAUPSA_EXP_1	🟩	✓	ELEM.REP
ZIP_BHAUPSA_EXP_2	🟩	✓	ELEM.REP
ZIQ_AUPSA_MPR_1	🟩	✓	ELEM.REP
ZIQ_AUPSA_MTH_1	🟩	✓	ELEM.REP
ZIQ_AUPSA_MTR_1	🟩	✓	ELEM.REP
ZIQ_AUPSA_PLNCOMP_1	🟩	✓	ELEM.REP
ZIQ_AUPSA_QTR_1	🟩	✓	ELEM.REP
ZIQ_AUPSA_SUPER_1	🟩	✓	ELEM.REP
ZIQ_AUPSA_YR_1	🟩	✓	ELEM.REP

Figure 2 List of Selected Queries and Query Elements

3. Select the appropriate queries, query elements, and dependent objects that need to be deleted, and click on the EXECUTE button. If there are any dependent objects, the system will pop up a message asking if you want to delete that dependent object as well (e.g., after deleting a query, the system will ask the user if it has to delete any dependent workbook if available).

Tip 75

Accessing the Metadata of an InfoCube for Troubleshooting

If you need to troubleshoot the SAP NetWeaver BW environment, you can get insight into the metadata of InfoCubes by directly accessing their underlying tables.

In an SAP NetWeaver BW support environment, you may need to access the metadata of InfoCubes. This metadata is extremely valuable for housekeeping activities or troubleshooting issues related to the overall system or specific InfoCubes. There is no easy method to access this information using standard SAP screens. The only way to get this information is by navigating several screens and manually noting the details.

This tip lists all of the key tables that you can use to access the metadata of an InfoCube or list of InfoCubes and also describes the method to access the metadata.

✓ And Here's How ...

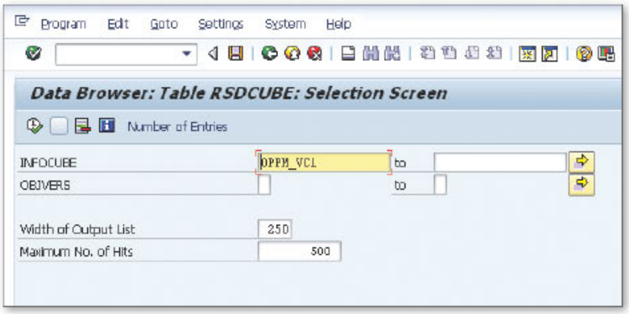
The list of key tables in Table 1 provides you with insight into the metadata for an InfoCube. Each of these tables provides a different type of information on the InfoCube.

You can access these tables using Transaction SE16 (Data Browser). You may also create utility programs using these tables by writing ABAP programs on these tables. Follow these steps to access the table information:

Table Name	Description
RSDCUBE	Directory of InfoCubes
RSDCUBET	Texts on InfoCubes
RSDCUBEIOBJ	Objects per InfoCubes (where-used list)
RSDDIME	Directory of dimensions
RSDDIMET	Texts on dimensions
RSDDIMEIOBJ	InfoObjects for each dimension (where-used list)
RSDCUBEMULTI	InfoCubes involved in a MultiCube
RSDICMULTIIOBJ	MultiProvider: selection/identification of InfoObjects
RSDICHAPRO	Characteristic properties specific to an InfoCube
RSDIKYFPRO	Flag properties specific to an InfoCube
RSDICVALIOBJ	InfoObjects of the stock validity table for the InfoCube

Table 1 List of SAP Tables That Contain Metadata Information on InfoCubes

1. Log on to the SAP NetWeaver BW environment and run Transaction SE16.
2. Enter the table name RSDCUBE and press `Enter`.
3. On the resulting selection screen (see Figure 1), enter the name of a specific InfoCube or a list of InfoCubes for which you want to analyze the metadata. You may also specify an InfoArea in this selection screen. Click on the EXECUTE icon on the screen or press `F8`.

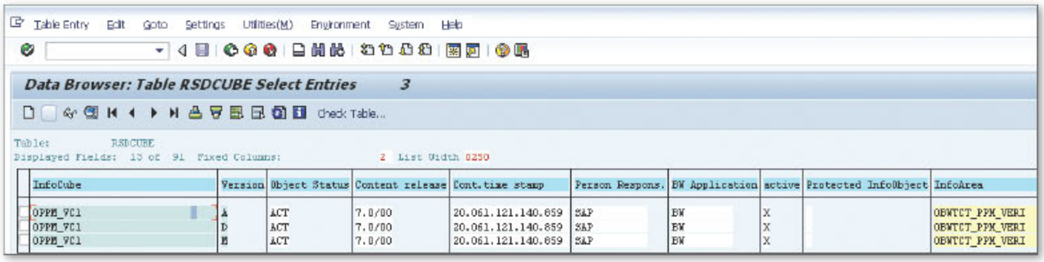


Data Browser: Table RSDCUBE: Selection Screen

INFOCUBE	0PPM_VC1	to	
OBJVERS		to	
Width of Output List	250		
Maximum No. of Hits	500		

Figure 1 Selection Screen for the Data Browser

4. The next screen displays all of the entries for the selected InfoCube(s) (see Figure 2).



Data Browser: Table RSDCUBE Select Entries 3

Table: RSDCUBE
Displayed Fields: 13 of 91 Fixed Columns: 2 List Width 0250

InfoCube	Version	Object Status	Content release	Cont.time stamp	Person Respons.	BW Application	active	Protected InfoObject	InfoArea
0PPM_VC1	A	ACT	7.0/00	20.061.121.140.859	SAP	BW	X		0BWTCT_PPM_VERI
0PPM_VC1	D	ACT	7.0/00	20.061.121.140.859	SAP	BW	X		0BWTCT_PPM_VERI
0PPM_VC1	M	ACT	7.0/00	20.061.121.140.859	SAP	BW	X		0BWTCT_PPM_VERI

Figure 2 Entries for the Selected InfoCube(s)

5. Double-clicking on any one of the records listed in Figure 2 will show you all of the details on the cube, as you can see in Figure 3.

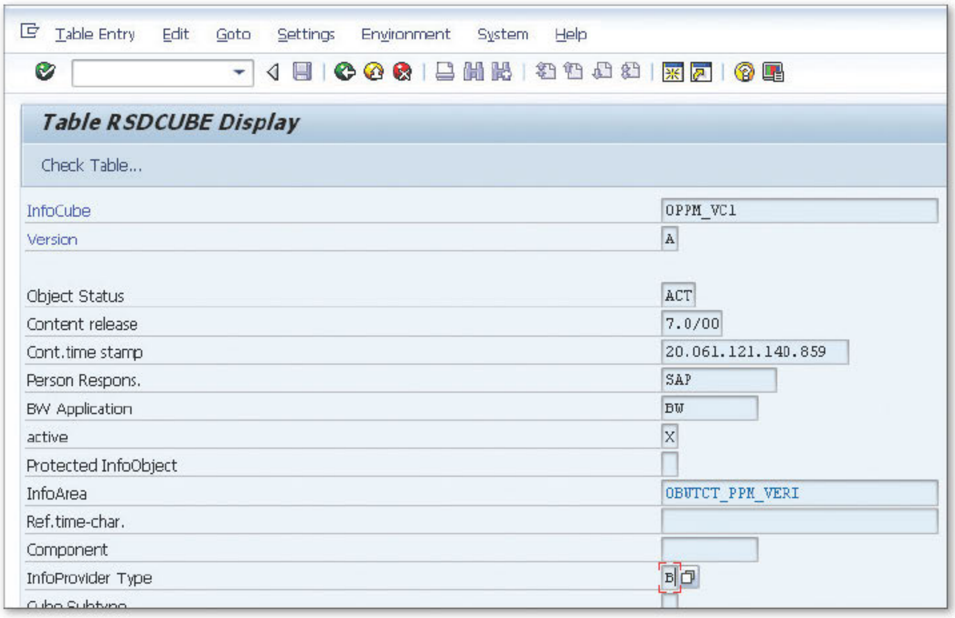


Table RSDCUBE Display

InfoCube	0PPM_VC1
Version	A
Object Status	ACT
Content release	7.0/00
Cont.time stamp	20.061.121.140.859
Person Respons.	SAP
BW Application	BW
active	X
Protected InfoObject	
InfoArea	0BWTCT_PPM_VERI
Ref.time-char.	
Component	
InfoProvider Type	B

Figure 3 Detailed Metadata Information on the InfoCube

Applying the preceding method to the different tables listed in Table 1 will give you access to various types of valuable metadata information on the InfoCubes.

If you need to access this data routinely, you can create views on some of these tables so that you can join two or more tables. Alternatively, you can create a utility program to access the information in a more user-friendly manner.

Tip 76

Tracking Changes to Common SAP NetWeaver BW Objects Using the Rev-Trac Tool

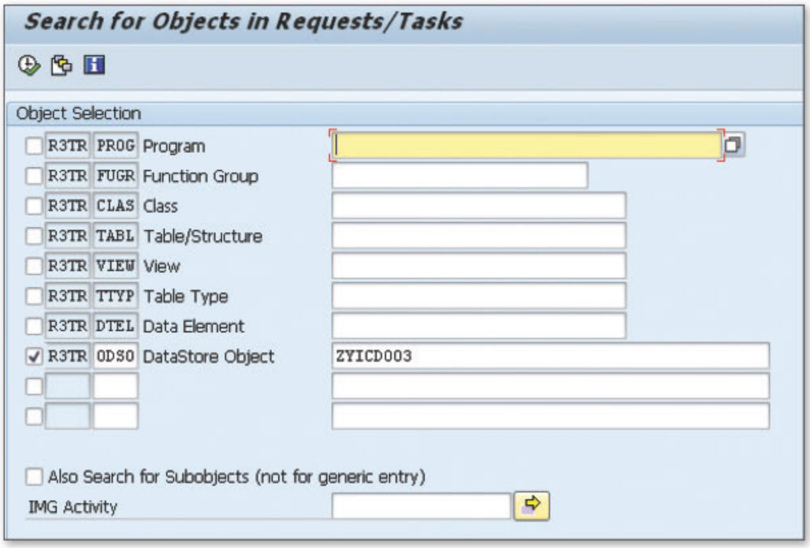
When many people are working on an implementation where changes to objects require permission, you can activate track changes.

When you're working with a large SAP NetWeaver BW implementation, people will hurry to complete their tasks and may cut some corners. This can involve making changes to common objects such as master data or DSOs with General Ledger data without obtaining approval. This can either positively or negatively affect multiple teams that are sharing the common objects, but when there is a problem, nobody claims ownership. To avoid this situation, you can turn on the Rev-Trac tool to track what changes are made to common objects.

✓ And Here's How ...

To access Rev-Trac, go to Transaction SE03 and follow these steps:

1. On the initial screen, double click on Transport Organizer Tools • Objects in Requests • Search for Objects in Requests/Tasks. The screen shown in Figure 1 appears.
2. In this screen, you can track changes for all of the standard SAP NetWeaver BW objects such as programs, function groups, classes, tables, data elements, DSOs, and so on. Select the last checkbox, and enter "ODSO" if you want to track the changes for a DSO, such as "ZYICD003" (see Figure 1).



Search for Objects in Requests/Tasks

Object Selection

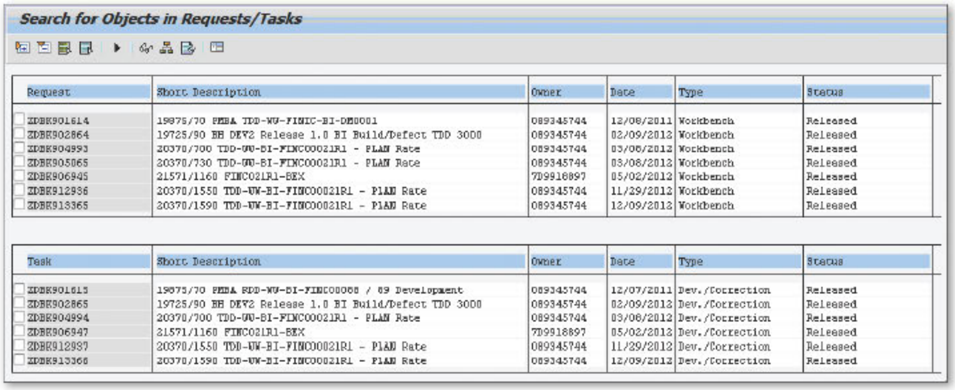
☐	R3TR	PROG	Program	
☐	R3TR	FUGR	Function Group	
☐	R3TR	CLAS	Class	
☐	R3TR	TABL	Table/Structure	
☐	R3TR	VIEW	View	
☐	R3TR	TTYP	Table Type	
☐	R3TR	DTEL	Data Element	
✓	R3TR	ODSO	DataStore Object	ZYICD003
☐				
☐				

☐ Also Search for Subobjects (not for generic entry)

IMG Activity

Figure 1 Search for Objects in Requests/Tasks

3. Click the EXECUTE button. You'll see all of the changes made to this object and the corresponding users who made the changes as shown in Figure 2.



Search for Objects in Requests/Tasks

Request	Short Description	Owner	Date	Type	Status
ZDBK901614	19875/70 PMBA TDD-WU-FINIC-BI-DM0001	089345744	12/08/2011	Workbench	Released
ZDBK902864	19725/90 BH DEV2 Release 1.0 BI Build/Defect TDD 3000	089345744	02/09/2012	Workbench	Released
ZDBK904993	20370/700 TDD-UU-BI-FINC00021R1 - PLAN Rate	089345744	03/06/2012	Workbench	Released
ZDBK905065	20370/730 TDD-UU-BI-FINC00021R1 - PLAN Rate	089345744	03/08/2012	Workbench	Released
ZDBK906945	21571/1160 FINC021R1-BEX	7D9918897	05/02/2012	Workbench	Released
ZDBK912936	20370/1550 TDD-UW-BI-FINC00021R1 - PlAN Rate	089345744	11/29/2012	Workbench	Released
ZDBK913365	20370/1590 TDD-UW-BI-FINC00021R1 - PlAN Rate	089345744	12/09/2012	Workbench	Released

Task	Short Description	Owner	Date	Type	Status
ZDBK901615	19875/70 PMBA RDD-WU-BI-FINC00088 / 89 Development	089345744	12/07/2011	Dev./Correction	Released
ZDBK902865	19725/90 BH DEV2 Release 1.0 BI Build/Defect TDD 3000	089345744	02/09/2012	Dev./Correction	Released
ZDBK904994	20370/700 TDD-UU-BI-FINC00021R1 - PLAN Rate	089345744	03/06/2012	Dev./Correction	Released
ZDBK906947	21571/1160 FINC021R1-BEX	7D9918897	05/02/2012	Dev./Correction	Released
ZDBK912937	20370/1550 TDD-UW-BI-FINC00021R1 - PlAN Rate	089345744	11/29/2012	Dev./Correction	Released
ZDBK913366	20370/1590 TDD-UW-BI-FINC00021R1 - PlAN Rate	089345744	12/09/2012	Dev./Correction	Released

Figure 2 Results of Rev-Trac: Changed Objects and Responsible Users

This log provides details on all the changes to the object with the date and owner of the change. The log also lists the transport request number associated with the change.

Similarly, you can select any other object in the system in Figure 1 and generate the same log for that object.

Tip 77

Suppressing Unwanted Messages in BEx Workbooks or Queries

You can selectively suppress unwanted messages in BEx workbooks or BEx queries to facilitate easy navigation.

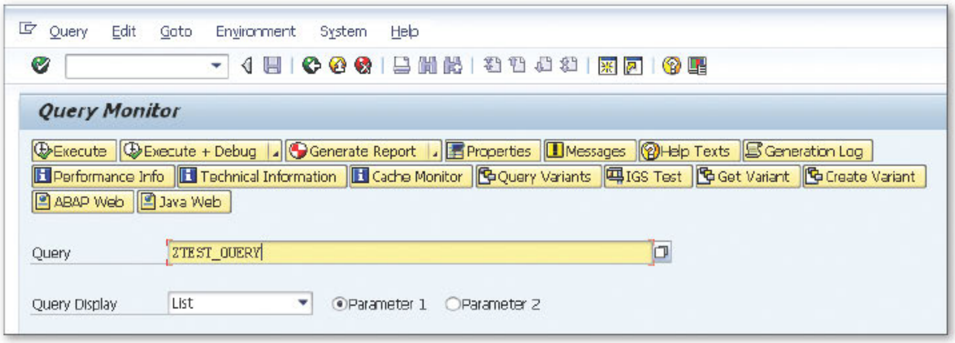
Depending on the design of the reports, BEx workbooks and queries sometimes return informative warning or error messages. While error messages are useful in communicating the status of data retrieval, the informative or warning messages can be annoying to the end user and counterintuitive. Double-clicking on these messages often won't provide any valuable information such as an error message number or a description.

To get rid of these annoying messages for your end users, we'll go over a simple method you can use to selectively suppress BEx query messages.

✓ And Here's How ...

Follow these steps to suppress warning messages:

1. Log in to the SAP NetWeaver BW system, and execute Transaction RSRT. The Query Monitor screen appears as shown in Figure 1.
2. Enter the name of the query for which you want to suppress the messages. In this case, we'll use a test query named ZTEST_QUERY that is already created in the system.
3. Click on the Messages button at the top of the screen.

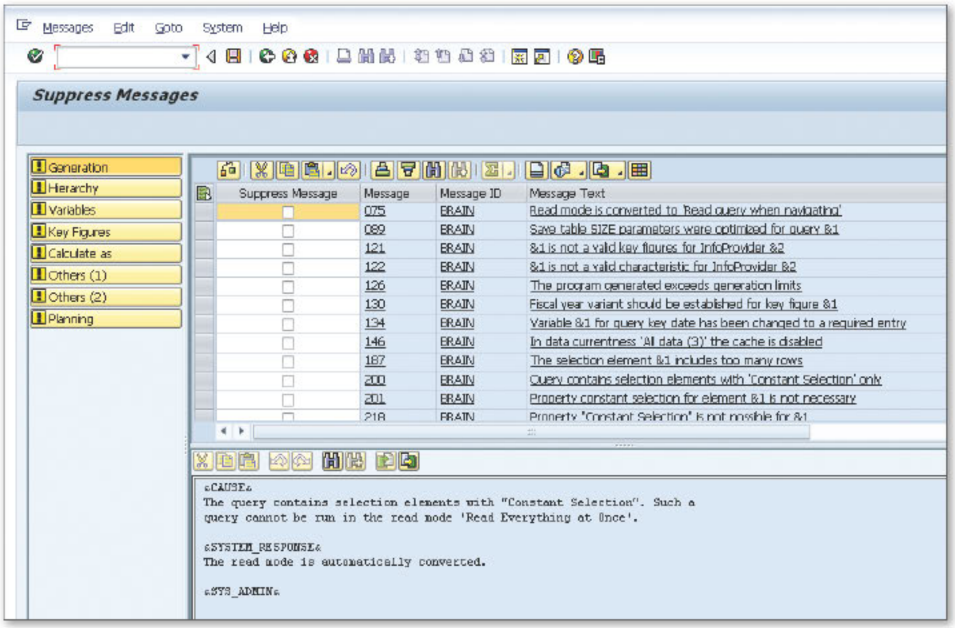


Query ZTEST_QUERY

Query Display List [✓ Parameter 1] Parameter 2

Figure 1 RSRT Screen

The Suppress Message screen appears as shown in Figure 2. This screen displays all available messages. On the left side of the screen, you'll see several options to filter the messages shown in the list on the right side of the screen. You can click on any of the categories to restrict the messages. For instance, if you click on the Variables option, you'll see a list of messages related to variables.



Suppress Message	Message	Message ID	Message Text
☐	075	BRAIN	Read mode is converted to 'Read query when navigating'
☐	089	BRAIN	Save table SIZE parameters were optimized for query &1
☐	121	BRAIN	&1 is not a valid key figures for InfoProvider &2
☐	122	BRAIN	&1 is not a valid characteristic for InfoProvider &2
☐	126	BRAIN	The program generated exceeds generation limits
☐	130	BRAIN	Fiscal year variant should be established for key figure &1
☐	134	BRAIN	Variable &1 for query key date has been changed to a required entry
☐	146	BRAIN	In data currentness 'All data (3)' the cache is disabled
☐	187	BRAIN	The selection element &1 includes too many rows
☐	200	BRAIN	Query contains selection elements with 'Constant Selection' only
☐	201	BRAIN	Property constant selection for element &1 is not necessary
☐	218	BRAIN	Property 'Constant Selection' is not possible for &1

```
&CAUSE&
The query contains selection elements with "Constant Selection". Such a
query cannot be run in the read mode 'Read Everything at Once'.

&SYSTEM_RESPONSE&
The read mode is automatically converted.

&SYS_ADMIN&
```

Figure 2 Suppress Messages

Check the box under Suppress Messages to selectively suppress the messages.

Tip 78

Checking the Last Failed Authorization in a Role

You can troubleshoot analysis authorization issues by accessing the authorization trace using Transaction RSECADMIN.

Often in SAP NetWeaver BW systems, users complain about not being able to fully access certain reports and queries. It can be quite redundant recopying authorizations without a clear idea regarding the source of the issue. To avoid assigning authorizations to users who may already have them, we'll show you a method to use to check the last failed authorization of a user. This will help you take further action in adding the failed authorization to the user.

✓ And Here's How ...

Follow these steps to use Transaction RSECADMIN to check for missing user authorizations:

1. Run Transaction RSECADMIN, and click on the ANALYSIS tab as shown in Figure 1. Click on the EXECUTE AS button.

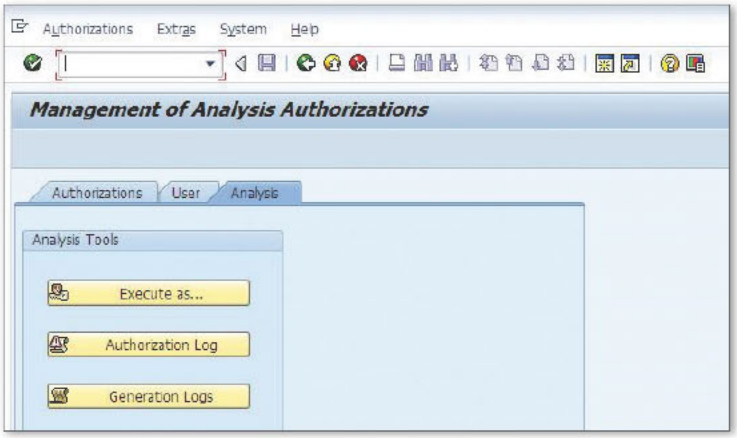
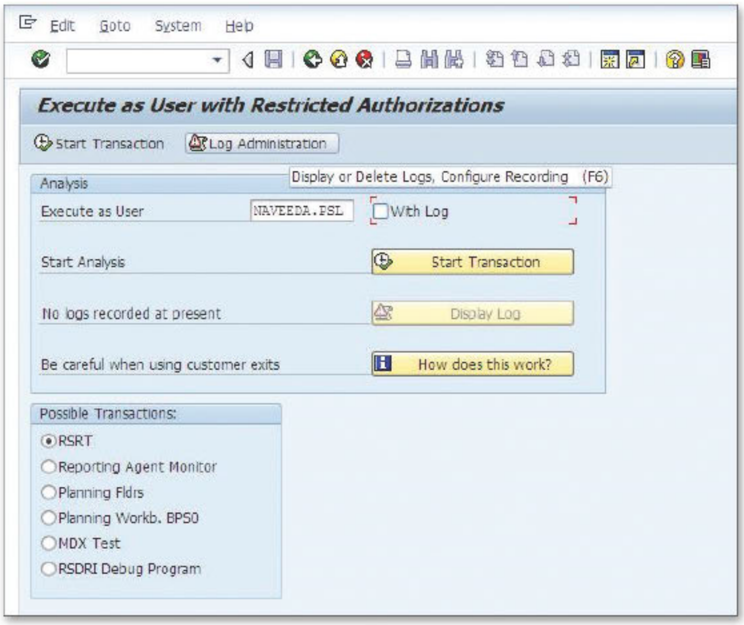


Figure 1 Transaction RSECADMIN

2. In the next screen, click on the Log Administration button (see Figure 2). The options for execution of the trace are provided in the Possible Transactions area. For this tip, choose the RSRT radio button. This option performs the authorization trace based on repot monitoring. Click on the Start Transaction button to generate the log of the trace.

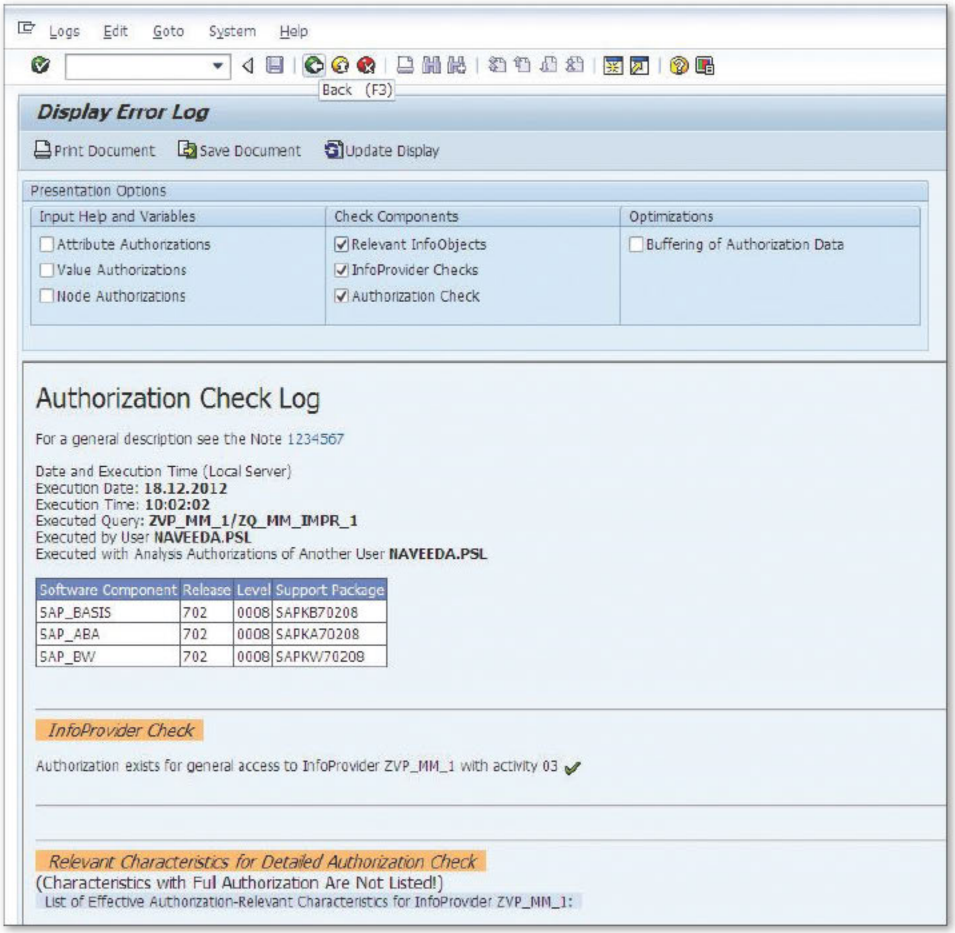


Execute as User NAVEEDA.PSL ☐ With Log
Start Analysis
No logs recorded at present
Be careful when using customer exits

[✓ RSRT]
Reporting Agent Monitor
Planning Fldrs
Planning Workb. BPS0
MDX Test
RSDRI Debug Program

Figure 2 Options for Executing the Trace

3. Now you can view the detail analysis check log as shown in Figure 3. Any missing authorizations are noted in red; otherwise, they'll be displayed in green.



Presentation Options

Input Help and Variables	Check Components	Optimizations
☐ Attribute Authorizations	☑ Relevant InfoObjects	☐ Buffering of Authorization Data
☐ Value Authorizations	☑ InfoProvider Checks	
☐ Node Authorizations	☑ Authorization Check	

Authorization Check Log

For a general description see the Note 1234567

Date and Execution Time (Local Server)
Execution Date: **18.12.2012**
Execution Time: **10:02:02**
Executed Query: **ZVP_MM_1/ZQ_MM_IMPR_1**
Executed by User **NAVEEDA.PSL**
Executed with Analysis Authorizations of Another User **NAVEEDA.PSL**

Software Component	Release	Level	Support Package
SAP_BASIS	702	0008	SAPKB70208
SAP_ABA	702	0008	SAPKA70208
SAP_BW	702	0008	SAPKW70208

InfoProvider Check

Authorization exists for general access to InfoProvider ZVP_MM_1 with activity 03 ✓

Relevant Characteristics for Detailed Authorization Check
(Characteristics with Full Authorization Are Not Listed!)
List of Effective Authorization-Relevant Characteristics for InfoProvider ZVP_MM_1:

Figure 3 Error Log Based on the Trace of the Authorization Checks

The trace first displays the name of the InfoProvider and the query name that the user executed. It also provides you with a list of characteristics in the cube for which the user has non-full (*) access as these need to be checked at a more detailed level. The trace also shows the authorization checks for these characteristics with non-full authorizations. This information can be very helpful in troubleshooting authorization issues.

Tip 79

Transferring SAP NetWeaver BW Authorizations into SAP BusinessObjects

You can save time and reduce errors by transferring SAP NetWeaver BW authorizations into SAP BusinessObjects to maintain both authorizations from a single place.

Report security in SAP BusinessObjects is controlled either using the folder-level security or the object-level security within the folders by giving specific rights at that level. However, if you are reporting on SAP NetWeaver BW data, then you also need to build data-level security using SAP NetWeaver BW authorizations. This can lead to a dual maintenance situation where the security maintenance has to be done in SAP BusinessObjects as well as in the SAP NetWeaver BW system, which might result in errors but definitely requires more time and attention.

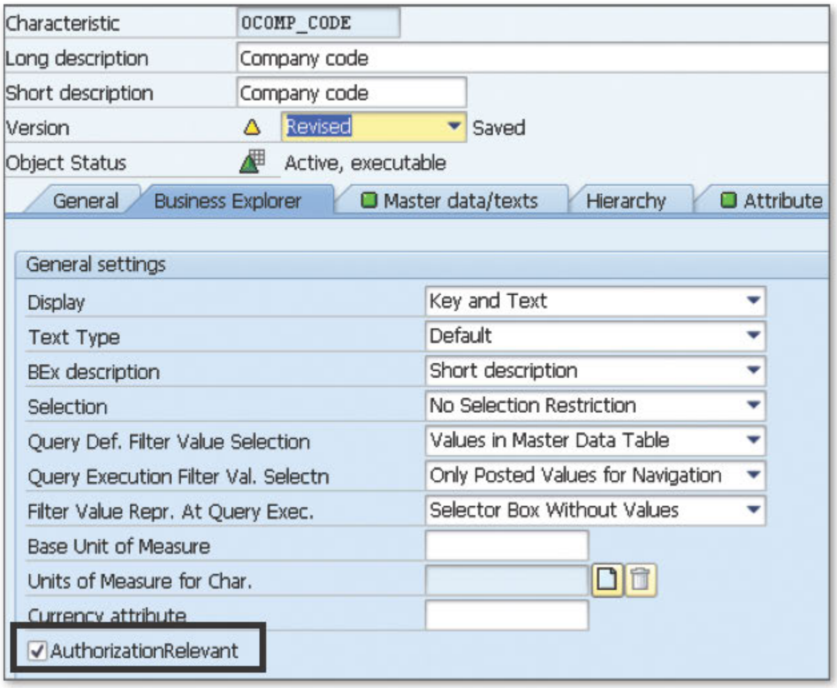
The more efficient approach is to maintain the data-level authorizations in SAP NetWeaver BW and then transfer these authorizations to SAP BusinessObjects. This allows for single maintenance of authorizations in SAP NetWeaver BW, which can then be assigned to groups in the SAP BusinessObjects Business Intelligence (SAP BusinessObjects BI) platform.

In this tip, we'll explain how to set this process up and how to transfer SAP NetWeaver BW authorizations to SAP BusinessObjects.

And Here's How ...

To illustrate this tip, we'll need to perform some steps on the SAP NetWeaver BW side and also on the SAP BusinessObjects system. Follow these steps:

1. Log in to the SAP NetWeaver BW system. We'll use characteristic 0COMP_CODE and make this object authorization relevant. Using Transaction RSA1, open the CHARACTERISTIC 0COMP_CODE (company code) in change mode. Click the AUTHORIZATIONRELEVANT checkbox as shown in Figure 1.



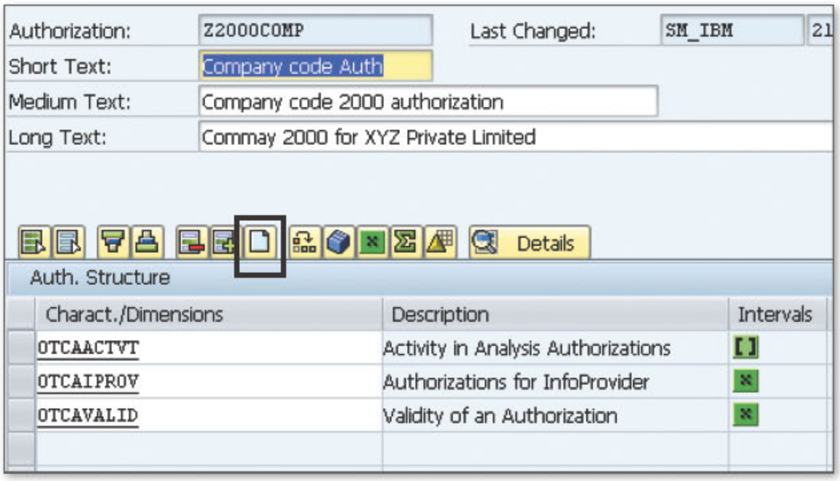
Characteristic	0COMP_CODE
Long description	Company code
Short description	Company code
Version	Revised Saved
Object Status	Active, executable

Display	Key and Text
Text Type	Default
BEx description	Short description
Selection	No Selection Restriction
Query Def. Filter Value Selection	Values in Master Data Table
Query Execution Filter Val. Selectn	Only Posted Values for Navigation
Filter Value Repr. At Query Exec.	Selector Box Without Values
Base Unit of Measure	
Units of Measure for Char.	
Currency attribute	

✓ AuthorizationRelevant

Figure 1 Making the Characteristic Authorization Relevant

2. Go to Transaction RSECADMIN and click the MAINTENANCE button. In the resulting screen, add Z2000COMP in the AUTHORIZATION field. This is the authorization object for the company code.
3. Click the CREATE button, and then click the SPECIAL CHARACTER button. The system will insert three characters/dimensions as shown in Figure 2.



Authorization:	Z2000COMP	Last Changed:	SM_IBM 21
Short Text:	Company code Auth		
Medium Text:	Company code 2000 authorization		
Long Text:	Commay 2000 for XYZ Private Limited		

Auth. Structure

Charact./Dimensions	Description	Intervals
0TCAACTVT	Activity in Analysis Authorizations	[ ]
0TCAIPROV	Authorizations for InfoProvider	🟩
0TCAVALID	Validity of an Authorization	🟩

Figure 2 Maintenance of Authorization Object Z2000COMP

The following entries appear in the CHARACT./DIMENSIONS column:

- 0TCAACTVT
  Grant authorizations to different activities such as change and display. The default value is 03 DISPLAY.
- 0TCAIPROV
  Grant authorizations to particular InfoProviders. The default value is *.
- 0TCAVALID
  This is used to check the validity of the authorization.

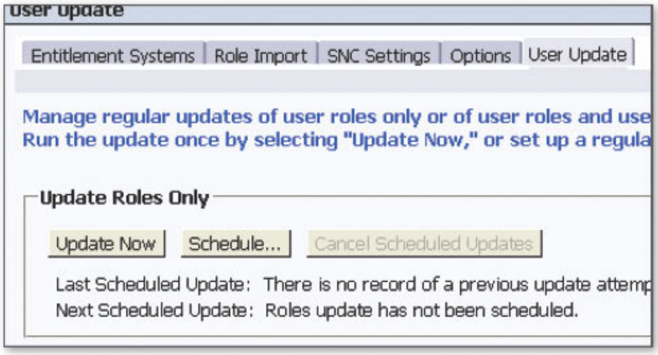
4. Additional characters/dimensions are used to include the characteristic company code for the authorization object:
   - 0TCAKYFNM
     Grant authorization to particular key figures.
   - 0COMP_CODE
     Company code.
5. Now you need to assign users to the authorization object Z2000COMP. In the main screen of Transaction RSECADMIN on the USER tab, click the ASSIGNMENT button.
6. Select a user, and click the CHANGE button.
7. Insert the technical name of your created authorization object Z2000COMP in the AUTH. column and press Enter. Click the SAVE button.

8. In the resulting screen, add the created authorization (Z2000COMP) via Transaction PFCG (role) to authorization object, S_RS_AUTH. Create a role named ZBI_XYZ_2021. When creating the connection in the SAP BusinessObjects Information Design Tool, set the authentication mode to USE SINGLE SIGN ON.

SAP BusinessObjects Steps

Now that you've enabled the transfer from SAP NetWeaver BW, follow these steps:

1. On the SAP BusinessObjects side, go to the Central Management Console, select AUTHENTICATION, and double-click on SAP.
2. Go to the ROLE IMPORT tab, search for the previously created role (ZBI_XYX_2021), and click the MANUALLY ADD button.
3. Click the UPDATE NOW button (see Figure 3).


User Update

Manage regular updates of user roles only or of user roles and use
Run the update once by selecting "Update Now," or set up a regula

Update Roles Only

Last Scheduled Update: There is no record of a previous update attemp
Next Scheduled Update: Roles update has not been scheduled.

Figure 3 Update of User Roles

The user is automatically created in SAP BusinessObjects and assigned the same role that was created in the SAP NetWeaver BW system (ZB_XYZ_2021).

Tip 80

Analyzing Transport Logs to Identify Failing Elements within a Change Request

You can easily identify failing elements within a change request by using detailed analysis on transport logs.

In an SAP NetWeaver BW environment, change requests are used to record all changes to SAP NetWeaver BW objects. The objects in the change requests are then migrated from the development objects to test and production systems using the transport process of the Change and Transport System. However, if the transport fails, finding the reason why can be very difficult because in most cases, the change request and the transport logs use an object ID instead of the actual name of the object. It's therefore difficult to identify the objects that caused the failure of the change request.

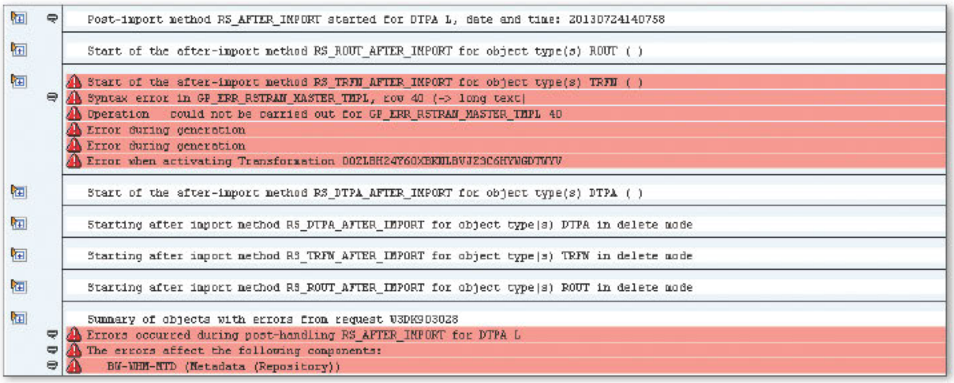
This tip discusses some steps you can use to diagnose the reasons for transport failure of change requests.

✓ And Here's How ...

Typically transports are either successful, end up with a warning (yellow), or end up with an error (red). In most cases, an analysis of the failed transport reveals why the transport failed and possible ways to resolve it.

The process is initiated by viewing the log of the erroneous transport. One of the ways of doing this is as follows:

1. Use Transaction SE10 to identify the erroneous transport.
2. Click on the Transport Logs button (or press Ctrl + Shift + F2), and then click on the log.
3. If you want to expand the log to see which object caused the transport, click on the Details icon. For example, the log details are shown in Figure 1.

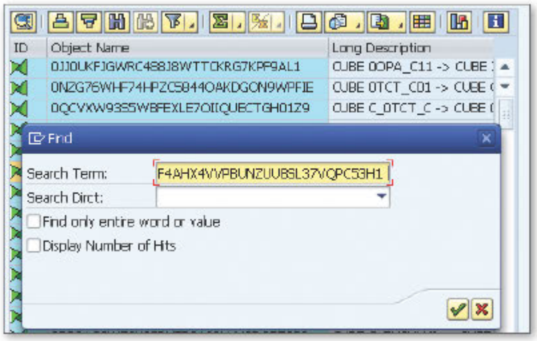


```
Post-import method RS_AFTER_IMPORT started for DTPA L, date and time: 20130724140758
Start of the after-import method RS_ROUT_AFTER_IMPORT for object type(s) ROUT ( )
Start of the after-import method RS_TRFN_AFTER_IMPORT for object type(s) TRFN ( )
Syntax error in GP_ERR_RSTRAN_MASTER_TMPL, row 40 (-> long text)
Operation   could not be carried out for GP_ERR_RSTRAN_MASTER_TMPL 40
Error during generation
Error during generation
Error when activating Transformation 00ZLBH24Y60XBKNLBVJZ3C6HYNGDTWYV
Start of the after-import method RS_DTPA_AFTER_IMPORT for object type(s) DTPA ( )
Starting after import method RS_DTPA_AFTER_IMPORT for object type(s) DTPA in delete mode
Starting after import method RS_TRFN_AFTER_IMPORT for object type(s) TRFN in delete mode
Starting after import method RS_ROUT_AFTER_IMPORT for object type(s) ROUT in delete mode
Summary of objects with errors from request U3DK903028
Errors occurred during post-handling RS_AFTER_IMPORT for DTPA L
The errors affect the following components:
   BW-WHM-MTD (Metadata (Repository))
```

Figure 1 Transport Log Display

The detailed log indicates that there is a syntax error in the transformation routine. However the transformation routine is displayed as an object ID 04ASA9AA9ZL19LG3TVC645A9LPJFPZ3X, so it cannot be searched for or identified in the Administrator Workbench (Transaction RSA1). To find out the details of the transformation, follow these steps:

1. Access Transaction RSA1, choose Transport, and then double-click on Select Objects.
2. In the ensuing screen, enter the search routine ID "04ASA9AA9ZL19LG3TVC645A9LPJFPZ3X" (see Figure 2).



ID	Object Name	Long Description
	0JJ0UKFJGWRC488J8WTTCKRG7KPF9AL1	CUBE 0OPA_C11 -> CUBE
	0NZG76WHF74HPZC5844OAKDGON9WPFIE	CUBE 0TCT_C01 -> CUBE
	0QCVXW93S5WBFEXLE7OIIQUECTGH01Z9	CUBE C_0TCT_C -> CUBE

Search Term: F4AHX4VVPBUNZUUBSL37VQPC53H1
Search Dirct:
☐ Find only entire word or value
☐ Display Number of Hits

Figure 2 Searching for the Transformation Routine

3. Identify the text of the transformation, which makes it easier to identify the object in Transaction RSA1. With these details, look up the routine associated with the transformation, and check in the target system (of the transport) to determine why it resulted in an error.

However, as mentioned previously, there may be multiple reasons for a failed transport. A few of the sample errors are:

- **Example 1**
  The error is in the process chain. Log in to the target system and bring up the process chain to see what the error is.
- **Example 2**
  Determine the query element and analyze why it wasn't transported correctly.
- **Example 3**
  The object was changed in the target system and collected in a transport. Identify the transport in the target system, release it, and then re-import the transport.
- **Example 4**
  Check the InfoObject and the associated tables.

Tip 81

Creating User-Input Required Variables for Use in Key Figure Calculations

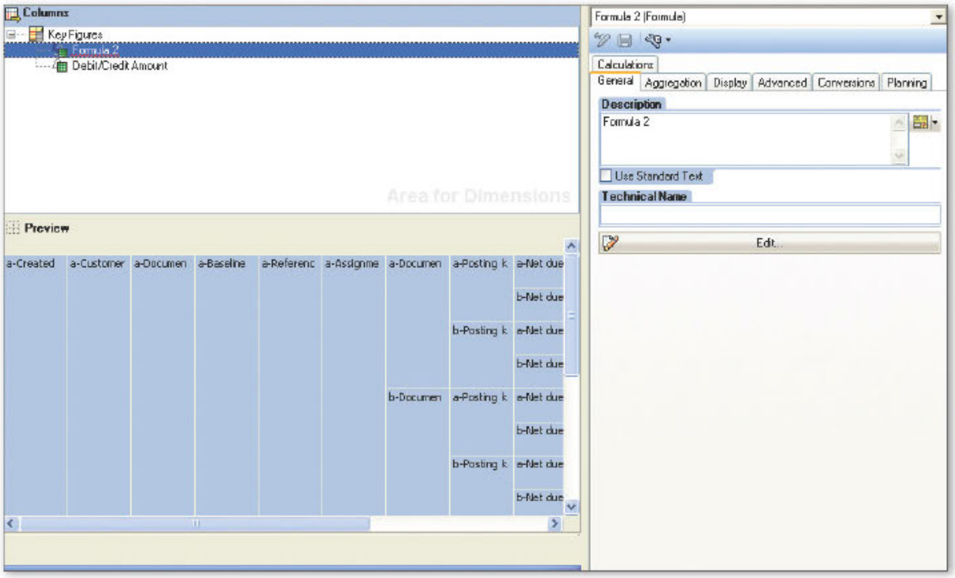
You can create a formula variable to provide additional flexibility to your key figure calculations in a BEx query.

Traditionally, variables in BEx are used to filter a result set. However, there are other instances when a variable will prompt a user to enter specific input that is used to drive calculations on the data set. This functionality adds more flexibility to the BEx query to drive dynamic or on-the-fly key figure calculations. To accomplish this, you can create a formula variable that will be used in the KEY FIGURES section of your query, which will save user input so that it can be used in other key figure calculations.

✓ And Here's How ...

To create a formula variable for end user use, follow these steps:

1. Open your existing BEx query.
2. Go into the KEY FIGURES section as shown in Figure 1, right-click to create a new formula, and click on the EDIT button in the PROPERTIES pane (right side).



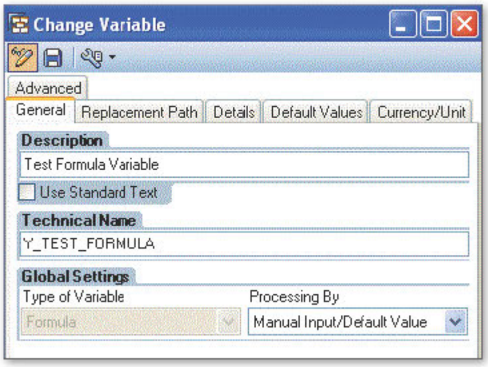
Description: Formula 2

Use Standard Text: ☐

Technical Name:

Figure 1 Create a New Key Figure Formula

3. Right-click the FORMULA VARIABLES folder under the GENERAL tab toward the bottom of the screen, and select NEW VARIABLE.
4. Right-click the new variable, and select EDIT; enter a DESCRIPTION and TECHNICAL NAME as shown in Figure 2.



Description: Test Formula Variable

Use Standard Text: ☐

Technical Name: Y_TEST_FORMULA

Global Settings

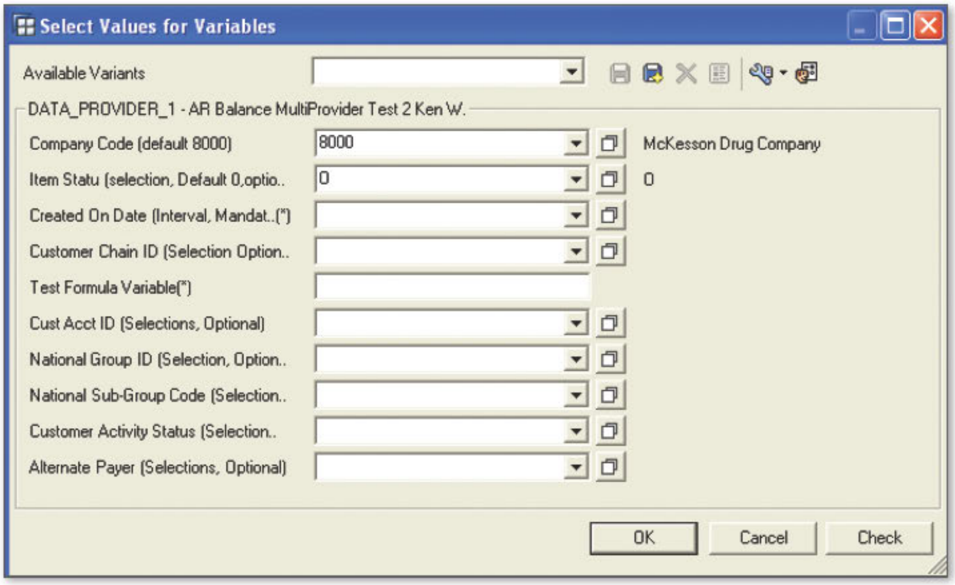
Type of Variable: Formula

Processing By: Manual Input/Default Value

Figure 2 Enter a Description and Technical Name

5. Go to the DETAILS tab, set that the variable is MANDATORY, and click the VARIABLE IS READY FOR INPUT checkbox.

6. Click and drag the newly created formula variable in the DETAIL VIEW pane.
7. Run the query and you'll see an input prompt for that variable (TEST FORMULA VARIABLE) (Figure 3).



Available Variants

DATA_PROVIDER_1 - AR Balance MultiProvider Test 2 Ken W.

Company Code (default 8000)	8000	McKesson Drug Company
Item Statu (selection, Default 0,optio..	0	0
Created On Date (Interval, Mandat..(*)		
Customer Chain ID (Selection Option..		
Test Formula Variable(*)		
Cust Acct ID (Selections, Optional)		
National Group ID (Selection, Option..		
National Sub-Group Code (Selection..		
Customer Activity Status (Selection..		
Alternate Payer (Selections, Optional)		

Figure 3 Test Formula Variable Now Active in the Query

Now you can leverage the TEST FORMULA VARIABLE calculated key figure in other BEx query calculations.

Tip 82

Overriding Default Cell Limits in the BEx Web Analyzer

You can adjust the row limit in the Web Analyzer to its maximum setting to process very large queries.

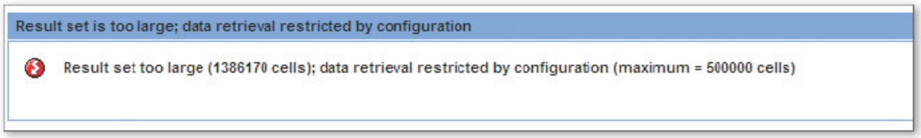
Like the Excel-based Analyzer tool with its row limits, the BEx Web Analyzer also has limits, but it's set at a cell level (rows multiplied by columns). There are two settings in SAP NetWeaver BW to address the cell limit: a default and a maximum limit. Both these limits are defined by your SAP NetWeaver BW Basis administrator. The default limit is the first limit that you'll encounter in Web Analyzer but that limit can be overridden by the maximum limit.

You'll encounter the cell limit error if you're running a query that brings back a lot of columns or rows. With this simple tip, you can set the row limit to its maximum setting to avoid further error messages upon query execution.

✓ And Here's How ...

To adjust the row limit, follow these steps:

1. Log in to the Web Analyzer and execute an existing query that contains more than 500,000 records.
2. When you reach the system default limit, you'll see a warning message informing you of the total number of cells required and the default limit as shown in Figure 1.


Result set is too large; data retrieval restricted by configuration

Result set too large (1386170 cells); data retrieval restricted by configuration (maximum = 500000 cells)

Figure 1 Web Analyzer Default Error Message for Exceeding 500,000 Cells

3. After you see this message, click on the SETTINGS link on the far right of the menu bar in the Web Analyzer.
4. Click over to the DATA PROVIDER tab (see Figure 2).
5. Select the MAXIMUM NO. OF CELLS FOR RESULT SET dropdown, and then set it to MAXIMUM NO. (2500000 CELLS). Then select APPLY.

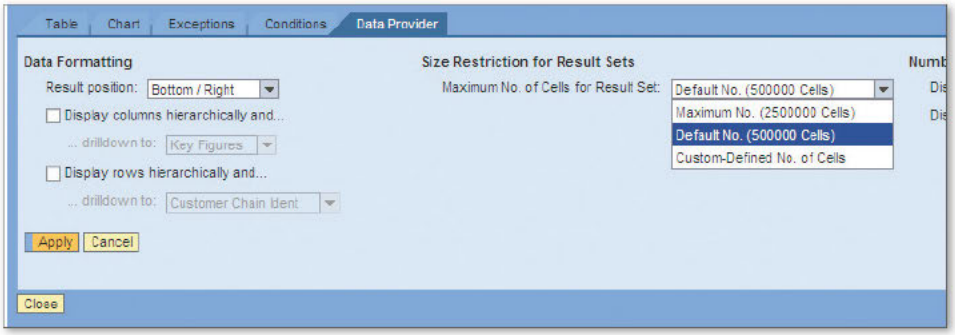


Table
Chart
Exceptions
Conditions
Data Provider
Data Formatting
Result position: Bottom / Right
Display columns hierarchically and...
... drilldown to: Key Figures
Display rows hierarchically and...
... drilldown to: Customer Chain Ident
Apply
Cancel
Size Restriction for Result Sets
Maximum No. of Cells for Result Set: Default No. (500000 Cells)
Maximum No. (2500000 Cells)
Default No. (500000 Cells)
Custom-Defined No. of Cells
Close

Figure 2 Web Analyzer Data Provider Tab with Maximum Number of Cells for Result Set Options

You can now run a query with up to 2,500,000 million cells.

Tip 83

Creating Secondary Indexes on DataStore Objects without a Transport

You can create secondary indexes on DataStore objects directly in production without a time-consuming transport with the use of simple commands.

You can use secondary indexes to optimize query performance on fields of DataStore objects (DSOs) that are not adequately covered by the primary index. Typically, you'll encounter missing secondary indexes on a DSO in a production environment. Missing secondary indexes may severely impact the performance of the associated queries. The best solution is to create a database level index (Oracle, DB2, etc.) on the desired fields. Typically, the issue is fixed by creating the missing index on the DSO in development and then transporting the changes to production. However, this can be a time-consuming process because the transports usually need to go through a testing and approval process.

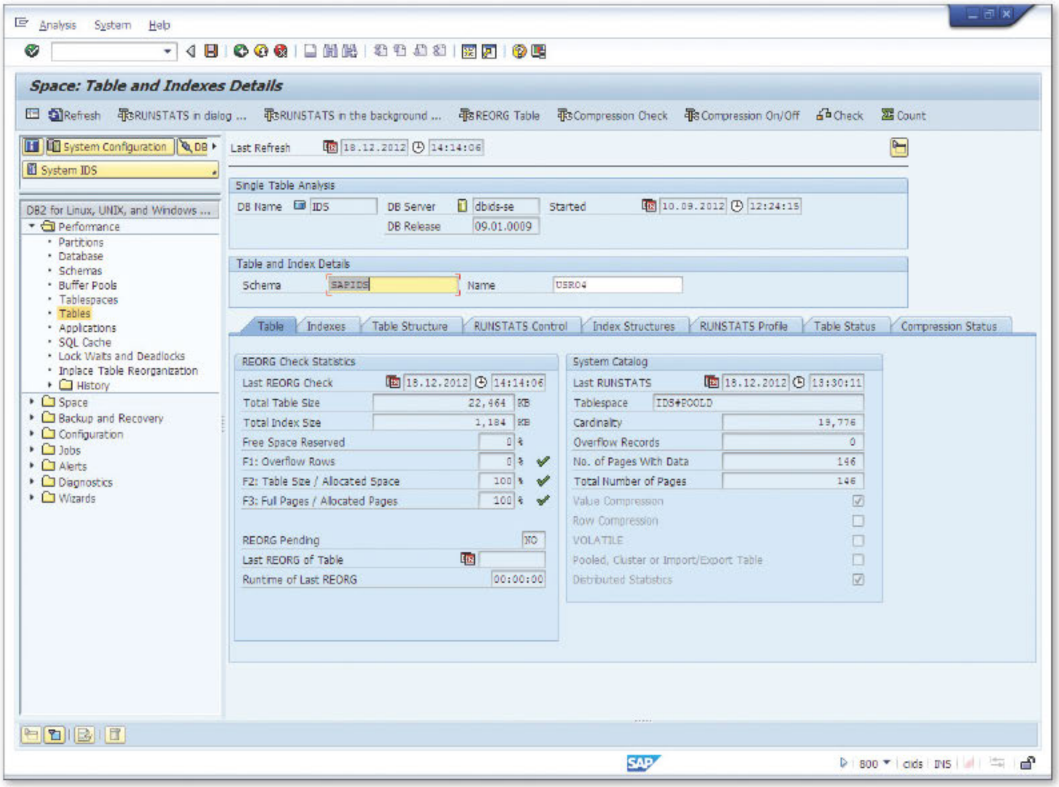
This tip describes a method you can use to create a secondary index on the DSO directly in the production database so that any query performance issues can be resolved immediately. At a later stage, the missing secondary index can be created in development, and then when the DSO is retransported into production, SAP NetWeaver BW will not generate a new secondary index in the database but instead will use the existing secondary DB level index.

✓ And Here's How ...

With SAP NetWeaver BW 7.10 and SAP NetWeaver 7.0 SP12, SAP has provided the new Transaction DBACOCKPIT that can be used for database monitoring and administration of SAP NetWeaver systems. Transaction DBACOCKPIT covers all database platforms that are supported by SAP.

For this tip, we will assume a scenario where SAP NetWeaver BW is installed on a DB2 database. However, you can use this method on any database.

1. Run Transaction DBACOCKPIT (see Figure 1), and choose Performance • Tables. Find USR04 in DBACOCKPIT and double-click on it.



Last Refresh 18.12.2012 14:14:06

Single Table Analysis

DB Name	IDS	DB Server	dbids-se	Started	10.09.2012 12:24:15
		DB Release	09.01.0009		

Table and Index Details

Schema SAPIDS Name USR04

REORG Check Statistics

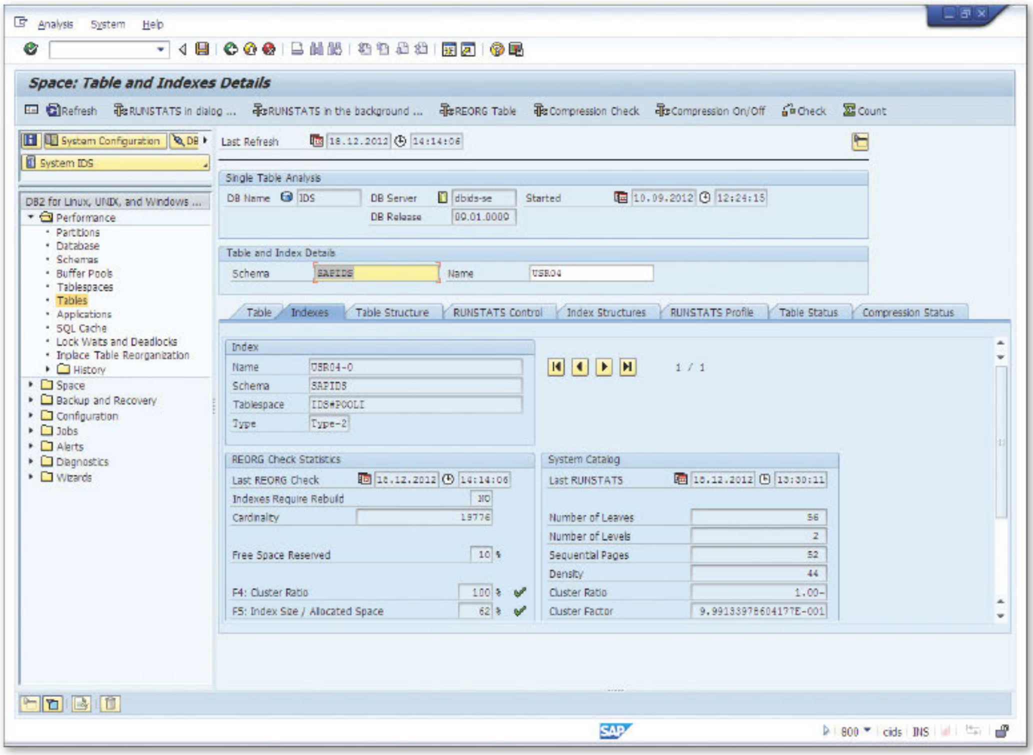
Last REORG Check	18.12.2012 14:14:06	
Total Table Size	22,464	KB
Total Index Size	1,184	KB
Free Space Reserved	0	%
F1: Overflow Rows	0	% ✓
F2: Table Size / Allocated Space	100	% ✓
F3: Full Pages / Allocated Pages	100	% ✓
REORG Pending	NO	
Last REORG of Table		
Runtime of Last REORG	00:00:00	

System Catalog

Last RUNSTATS	18.12.2012 13:30:11
Tablespace	IDS#POOLD
Cardinality	19,776
Overflow Records	0
No. of Pages With Data	146
Total Number of Pages	146
Value Compression	☑
Row Compression	☐
VOLATILE	☐
Pooled, Cluster or Import/Export Table	☐
Distributed Statistics	☑

Figure 1 Transaction DBACOCKPIT Screen

2. Go to the Indexes tab. In Figure 2, you can see index 1 of 1. This means that the DSO currently has only one index. If more than one index is available, you can use the arrows to toggle between them.



Last Refresh 18.12.2012 14:14:06

Single Table Analysis

DB Name	IDS	DB Server	dbids-se	Started	10.09.2012 12:24:15
		DB Release	09.01.0000		

Table and Index Details

Schema SAPIDS Name USR04

Index

Name	USR04-0
Schema	SAPIDS
Tablespace	IDS#POOLI
Type	Type-2

1 / 1

REORG Check Statistics

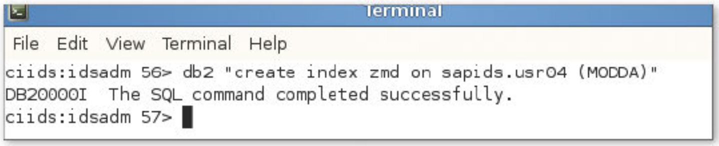
Last REORG Check	18.12.2012 14:14:06
Indexes Require Rebuild	NO
Cardinality	19776
Free Space Reserved	10 %
F4: Cluster Ratio	100 % ✓
F5: Index Size / Allocated Space	62 % ✓

System Catalog

Last RUNSTATS	18.12.2012 13:30:11
Number of Leaves	56
Number of Levels	2
Sequential Pages	52
Density	44
Cluster Ratio	1.00-
Cluster Factor	9.9913397860417 7E-001

Figure 2 Index Tab of Transaction DBACOCKPIT

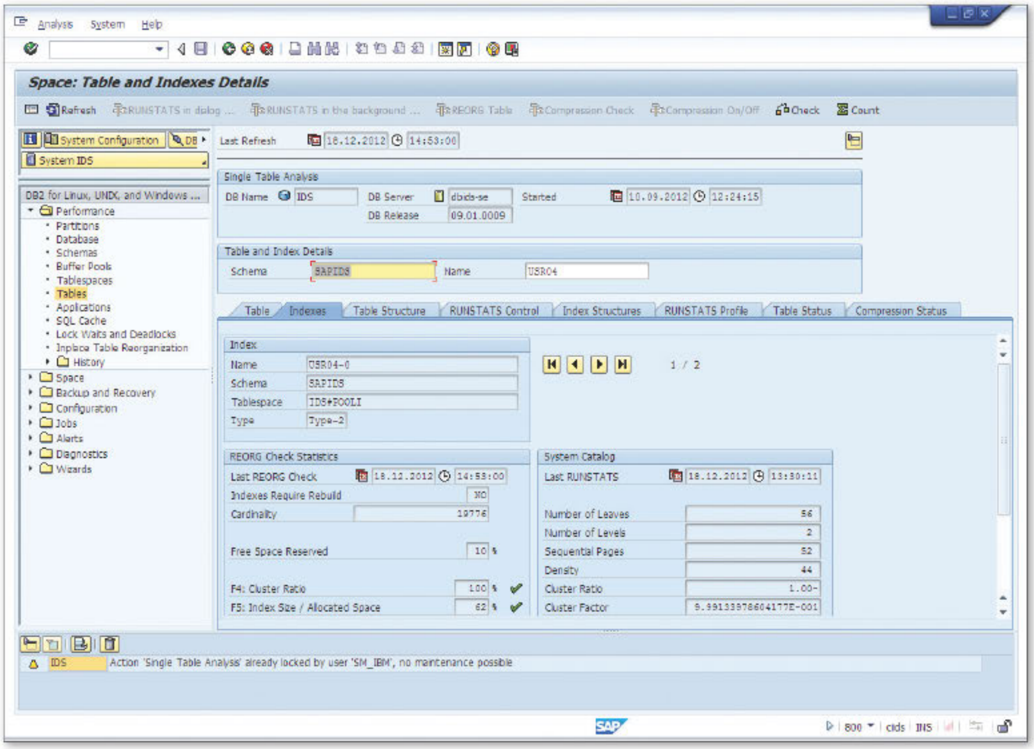
3. Now create the index with the name "ZMD" in Table USR04, column MODDA, by following these steps:
   - Log in to DB2 as <SID>adm.
   - Issue the command to create an index at the database level as shown in Figure 3.



```
ciids:idsadm 56> db2 "create index zmd on sapids.usr04 (MODDA)"
DB20000I  The SQL command completed successfully.
ciids:idsadm 57> █
```

Figure 3 Create Index Command

In the INDEXES tab in Figure 4, you can now see the new index that was created on the database side. The numbers near the arrow show 1 of 2. So the SAP NetWeaver BW system has recognized the new index that was created on the database.



Space: Table and Indexes Details

Last Refresh 18.12.2012 14:53:00

Single Table Analysis

DB Name IDS | DB Server dbids-se | Started 10.09.2012 12:24:15
DB Release 09.01.0009

Table and Index Details

Schema SAPIDS | Name USR04

Index

Name	USR04-0
Schema	SAPIDS
Tablespace	IDS#POOLI
Type	Type-2

1 / 2

REORG Check Statistics

Last REORG Check	18.12.2012 14:53:00
Indexes Require Rebuild	NO
Cardinality	19776
Free Space Reserved	10 %
F4: Cluster Ratio	100 % ✓
F5: Index Size / Allocated Space	62 % ✓

System Catalog

Last RUNSTATS	18.12.2012 13:30:11
Number of Leaves	56
Number of Levels	2
Sequential Pages	52
Density	44
Cluster Ratio	1.00-
Cluster Factor	9.99133978604177E-001

IDS Action 'Single Table Analysis' already locked by user 'SM_IBM', no maintenance possible

Figure 4 DBA Cockpit Showing a Second Index Available in the System

Any queries on this DSO will now use this new index, and the users will be able to see immediate performance benefits with the queries they are executing.

Here are a few other helpful pointers with regard to indexes on DSOs:

- If you're adding an index to a very large DSO, you may want to arrange the transport to be sent during off hours to control when the create index job runs. This can drastically consume system resources and adversely affect response times.
- Another approach is to use DSO partitioning if the DSO size is very large. This helps the database perform parallel processing on the indexing job and reduces the completion time.

Never delete the index in a development environment and transport to resolve a missing index. If the index has already been defined in SAP NetWeaver BW and is missing, resolve the missing index in the production environment.

Tip 84

Optimizing Query Performance by Using Code to Restrict Parallel Access to InfoProviders

You can use a characteristic to keep MultiProviders from accessing unnecessary data from too many InfoProviders, which will help optimize your query performance.

Typical reporting scenarios require extensive use of MultiProviders, which are a data union of basic InfoProviders. One of the main advantages of using a MultiProvider is that it allows parallel access to underlying basic InfoProviders, load balancing, resource utilization, and query pruning. However, you may find that certain reports only need data from one or two basic InfoProviders. In this situation, parallel access of all the basic InfoProviders that are included in the MultiProvider could be a drag on the performance of the report.

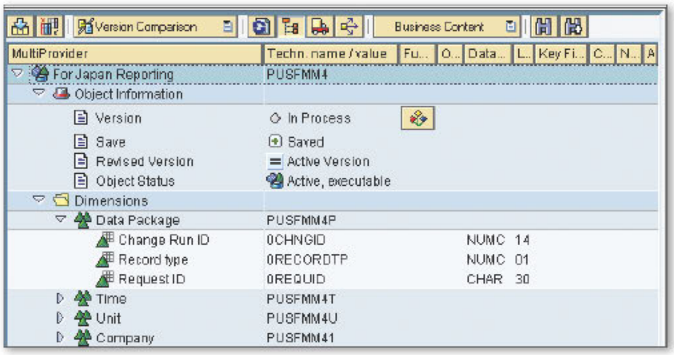
In this tip, we show you how to use characteristic 0INFOPROV when designing a query on the MultiProvider to ensure that a MultiProvider only retrieves data from relevant basic InfoProviders at query runtime. This tip assumes that you have a good understanding of BEx query definition, variable creation process, and the customer exit logic available for variables.

✓ And Here's How ...

As a refresher, the enhancement RSR00001 (BW: Enhancements for Global Variables in Reporting) is called up several times during execution of the report. The parameter `I__STEP` in the enhancement specifies when the enhancement is called up. For this tip, we will use `I_STEP = 2` that is called up directly after variable entry.

The InfoObject 0INFOPROV provides details of the source of data in a query based on a MultiProvider. It allows you to determine the basic InfoProvider(s) that is the source of data.

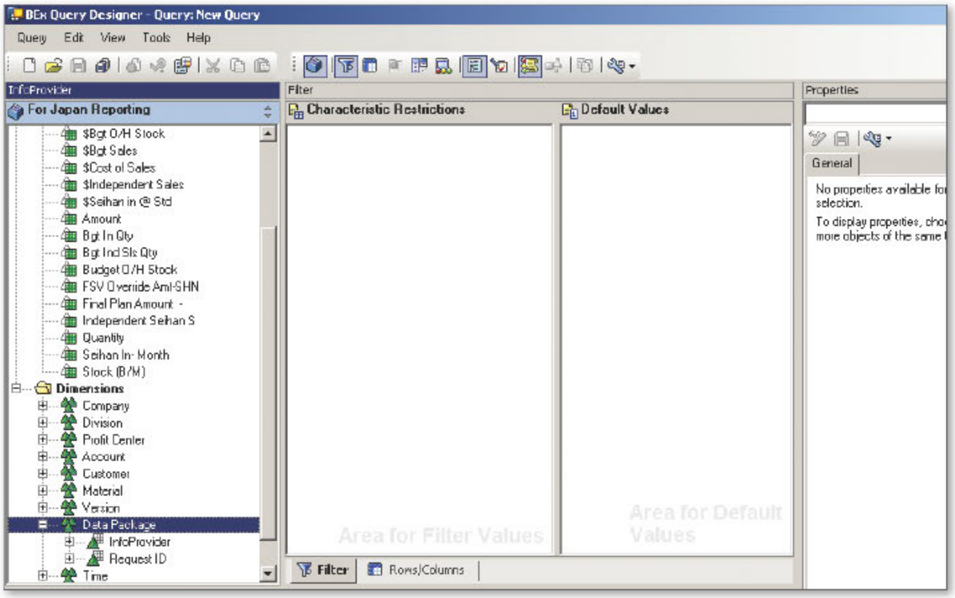
The InfoObject is not available when defining the MultiProvider (see Dimensions • Data Package in Figure 1).



MultiProvider	Techn. name / value	Fu...	O...	Data...	L...	Key Fi...	C...	N...	A
For Japan Reporting	PUSFMM4								
Object Information									
Version	In Process								
Save	Saved								
Revised Version	Active Version								
Object Status	Active, executable								
Dimensions									
Data Package	PUSFMM4P								
Change Run ID	0CHNGID			NUMC	14				
Record type	0RECORDTP			NUMC	01				
Request ID	0REQUID			CHAR	30				
Time	PUSFMM4T								
Unit	PUSFMM4U								
Company	PUSFMM41								

Figure 1 InfoObject 0INFOPROV Not Displayed in the Model

However, the InfoObject is available under Dimensions • Data Package on the left-side column when you use BEx Query Designer (see Figure 2).



For Japan Reporting
- $Bgt O/H Stock
- $Bgt Sales
- $Cost of Sales
- $Independent Sales
- $Seihan in @ Std
- Amount
- Bgt In Qty
- Bgt Ind Sls Qty
- Budget O/H Stock
- FSV Override Amt-SHN
- Final Plan Amount -
- Independent Seihan S
- Quantity
- Seihan In- Month
- Stock (B/M)
- Dimensions
  - Company
  - Division
  - Profit Center
  - Account
  - Customer
  - Material
  - Version
  - Data Package
    - InfoProvider
    - Request ID
  - Time

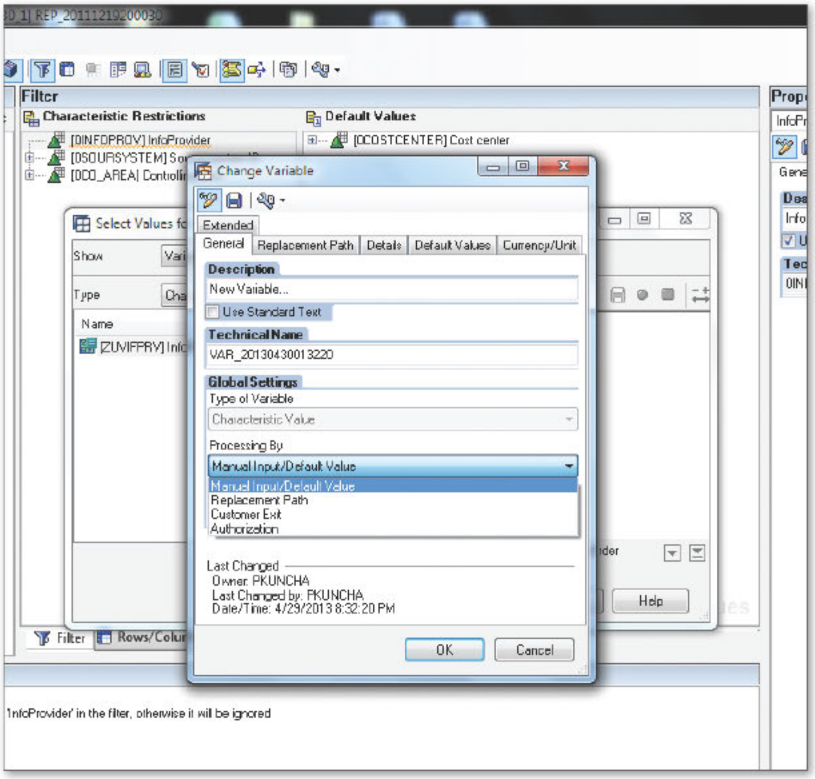
No properties available for selection.
To display properties, cho more objects of the same

Figure 2 BEx Query Screen for the MultiProvider and the InfoObject 0INFOPROV in the Data Package Dimension

This tip involves three basic steps—create a variable, populate the value, and write the logic—as explained in the following sections.

Create a Variable for the Characteristic 0INFOPROV

1. Add characteristic 0INFOPROV to all of the queries that are built on the MultiProvider. The InfoObject 0INFOPROV is available in the Data Package dimension of the MultiProvider, and you can drag and drop this InfoObject into your query definition.
2. Create a variable on the characteristic 0INFOPROV using the following steps:
   - Right-click on the InfoObject 0INFOPROV in the Characteristic Restrictions area, and then choose the Restrict option.
   - In the resulting screen, choose the Variables option from the dropdown box.
3. Create a variable on the characteristic 0INFOPROV with the following properties (see Figure 3):



[0INFOPROV] InfoProvider
[0SOURSYSTEM] So
[0CO_AREA] Controllin
[0COSTCENTER] Cost center
[ZUVIFPRV] Info

Description: New Variable...
Use Standard Text: ☐
Technical Name: VAR_20130430013220
Global Settings
Type of Variable: Characteristic Value
Processing By: Manual Input/Default Value
Manual Input/Default Value
Replacement Path
Customer Exit
Authorization
Last Changed
Owner: PKUNCHA
Last Changed by: PKUNCHA
Date/Time: 4/29/2013 8:32:20 PM

'InfoProvider' in the filter, otherwise it will be ignored

Figure 3 Creating a New Variable

- Technical name: "ZINFOP1"
- Description : "Customer Exit Variable for 0INFOPROV"
- Processing By: Customer Exit
- Mandatory: Yes
- Ready for Input: No

  Any variables that are mandatory and not ready for input are executed during the `I_STEP = 2` of the execution of the customer exit.

Populate the Value of Variable for the Characteristic 0INFOPROV with the Name of the Basic InfoProvider
Insert the variable into the filter section of the query.

Derive and populate the value of the customer exit variable ZINFOP1 defined in the earlier step by identifying another ready for input variable in the query. In the example shown in Figure 2, variables on characteristics such as Company, Profit Center, Division, or Customer could be a good candidate for variable selection entry. For this scenario, we assume that you have already created a data entry variable ZVCOMP for the characteristic Company.

Write Logic
In this step, you identify the basic InfoProvider(s) based on the value of other variables that are input by the query executor. Write code to derive the basic InfoProvider based on the selection value entered by the user. Following is the logic of the pseudo-code:

```
 Data: V_INFOPR type 0INFOPROV
    CASE i_vnam.
WHEN 'ZINFOP1'.
   IF i_step = 2.
      LOOP AT i_t_var_range INTO loc_var_range
           WHERE vnam = 'ZVCOMP'.
                      CASE
              When '100'.          "Company code "
                                V_INFOPR   =  'ZSALES1'
"Name of basic infoprovider
              When '101'.               "Company code "
                                V_INFOPR   =  'ZSALES2'
"Name of basic infoprovider
```

```
                                    ………
          …… . .
          …… . .
                         ENDCASE.
                  ENDLOOP.
                            ENDIF.
```

Note that this code is a simple pseudo-code for the logic; the actual code would most definitely require more extensive and complex logic to derive the basic InfoProviders.

With these steps, any MultiProvider queries that include the characteristic 0INFO-PROV will be optimized to access only the basic InfoProvider(s) that are really required to source the data requested by the report user.

Tip 85

Narrowing Down Data Selection on MultiProviders with Query Pruning

You can improve system performance for reporting in SAP NetWeaver BW by using the query pruning functionality to optimize reading access.

Generally, when queries are executed on a MultiProvider, the system performs a data selection on all InfoProviders that are part of the MultiProvider, even though the data selection made is applicable for only one InfoProvider. This data selection approach can slow the query execution time, particularly if the query needs data from only specific InfoProviders within the MultiProvider.

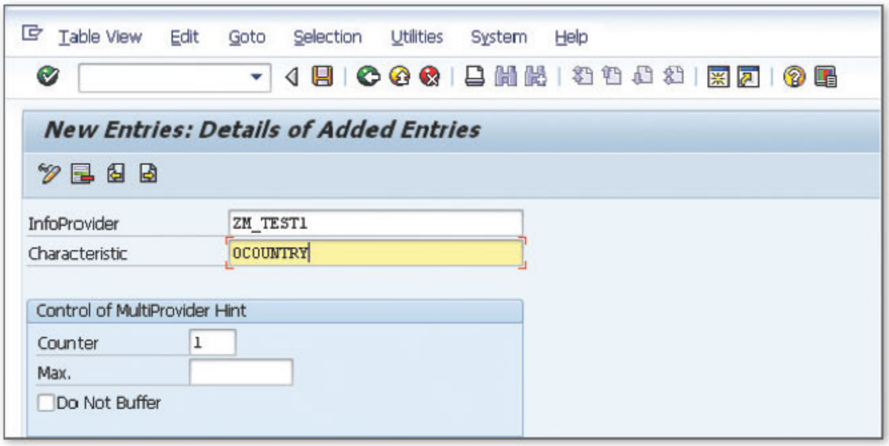
This tip describes a query pruning functionality that helps to identify which InfoProvider the query has to fetch data from. This is based on the selection made during execution of the query, and retrieves data from the relevant InfoProvider in the MultiProvider. With this approach, the end user gets vastly improved performance on the SAP Business Explorer (BEx) queries that are built on MultiProviders. This is a simple technique that is easy to set up and, if used for the right scenarios, can help improve the performance of the queries.

✓ And Here's How ...

This method of query pruning automatically determines the base InfoProvider(s) in a MultiProvider that needs to be restricted for querying based on a partitioning criteria that is maintained in a table on the SAP NetWeaver BW side. This solution will work only on MultiProviders that only have InfoCubes as base InfoProviders and will not work on DataStore Object (DSO) InfoProviders.

To do this, you first need to maintain the partitioning criteria in the customizing table RRKMULTIPROVHINT on the SAP NetWeaver BW system. Follow these steps:

1. Go to Transaction SE16, enter the table name as "RRKMULTIPROVHINT", and click on the CREATE icon. Access Table RRKMULTIPROVHINT, and specify the MULTIPROVIDER and a CHARACTERISTIC for partitioning.
2. The CHANGE VIEW screen appears. Click on the NEW ENTRIES icon.
3. Specify the name of the MultiProvider that you are trying to optimize and also characteristic(s) that can be used for partitioning. In this case, we will use an existing MultiProvider "ZM_TEST1" and use the characteristic "0COUNTRY" as the partitioning criteria. 0COUNTRY will now be set as a global filter for all queries on this MultiProvider (see Figure 1).



New Entries: Details of Added Entries

InfoProvider: ZM_TEST1
Characteristic: 0COUNTRY

Control of MultiProvider Hint
Counter: 1
Max.:
☐ Do Not Buffer

Figure 1 Creating a New Entry for the MultiProvider and Partitioning Criteria

When the query is executed, based on the selection made on 0COUNTRY, the system will fetch the data from the relevant PartProvider.

During the query execution time, the PartProviders that aren't needed are automatically filtered out. Because we have specified 0COUNTRY as the partitioning criteria for the MultiProvider, the OLAP processor will determine which InfoCubes in the MultiProvider can return data for the value of the country that is selected in the query. Based on this, the data manager will completely ignore the remaining InfoCubes that do not have the value.

Tip 86

Quickly Finding a BEx Query Definition in the SAP NetWeaver BW Backend

You can use an SAP-provided standard program to get a complete BEx query definition for analysis, documentation, or download purposes.

Query definition analysis is useful to understand the query in detail and help support the team to fix issues quickly. However, analyzing or documenting the query definition using BEx Query Designer is always a challenging and time-consuming process as it isn't easy to get all the query elements in one screen. Additionally, you can't download the query definition to an Excel document if you need to do further analysis.

This tip describes an easy way to quickly get a query definition for analysis or documentation. It also provides details on how to get only required parts of a query definition.

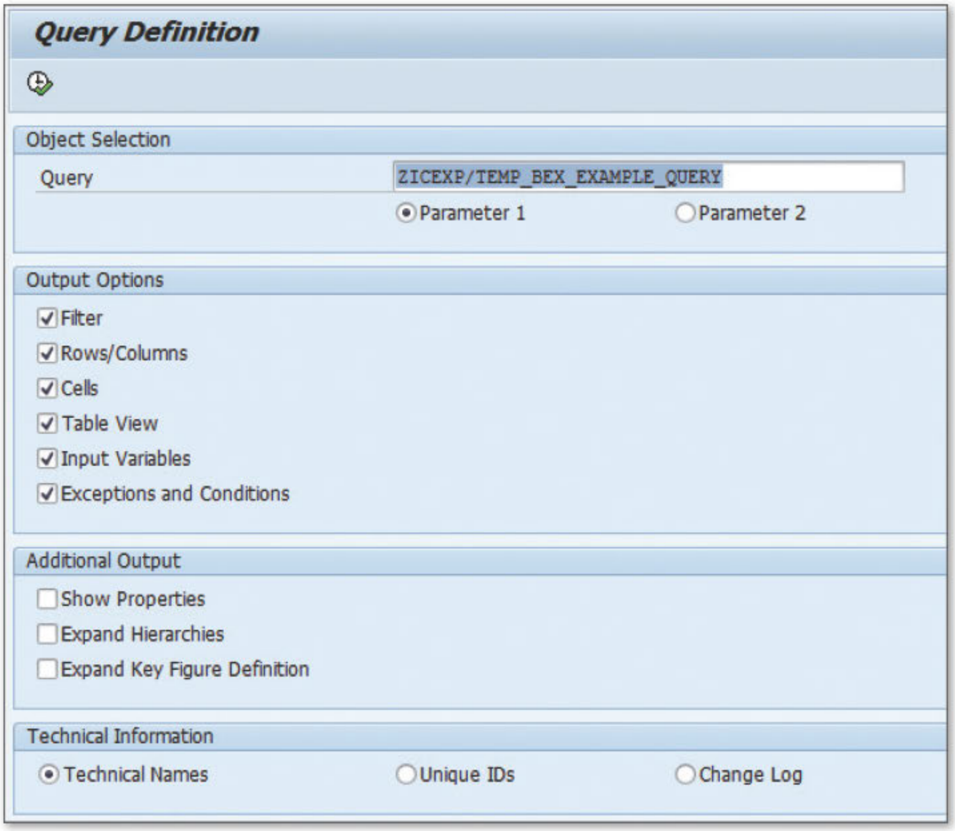
✓ And Here's How ...

To find a query definition in the SAP NetWeaver BW backend, follow these steps:

1. Execute Transaction SE38 to invoke the ABAP Editor.
2. Enter the program name "RSRQ_QUERYDEFINITION", and click on EXECUTE (or press F8).
3. In the next selection screen, enter the technical name of the query for which the query definition is needed. Also, choose the required OUTPUT OPTIONS (choose

only the required part of the query) and ADDITIONAL OUTPUT as needed (see Figure 1). On the same screen, select the type of TECHNICAL INFORMATION needed.

4. Click the EXECUTE button after completing the selection.



Object Selection

Query: ZICEXP/TEMP_BEX_EXAMPLE_QUERY

[✓ Parameter 1] Parameter 2

Output Options

- ✓ Filter
- ✓ Rows/Columns
- ✓ Cells
- ✓ Table View
- ✓ Input Variables
- ✓ Exceptions and Conditions

Additional Output

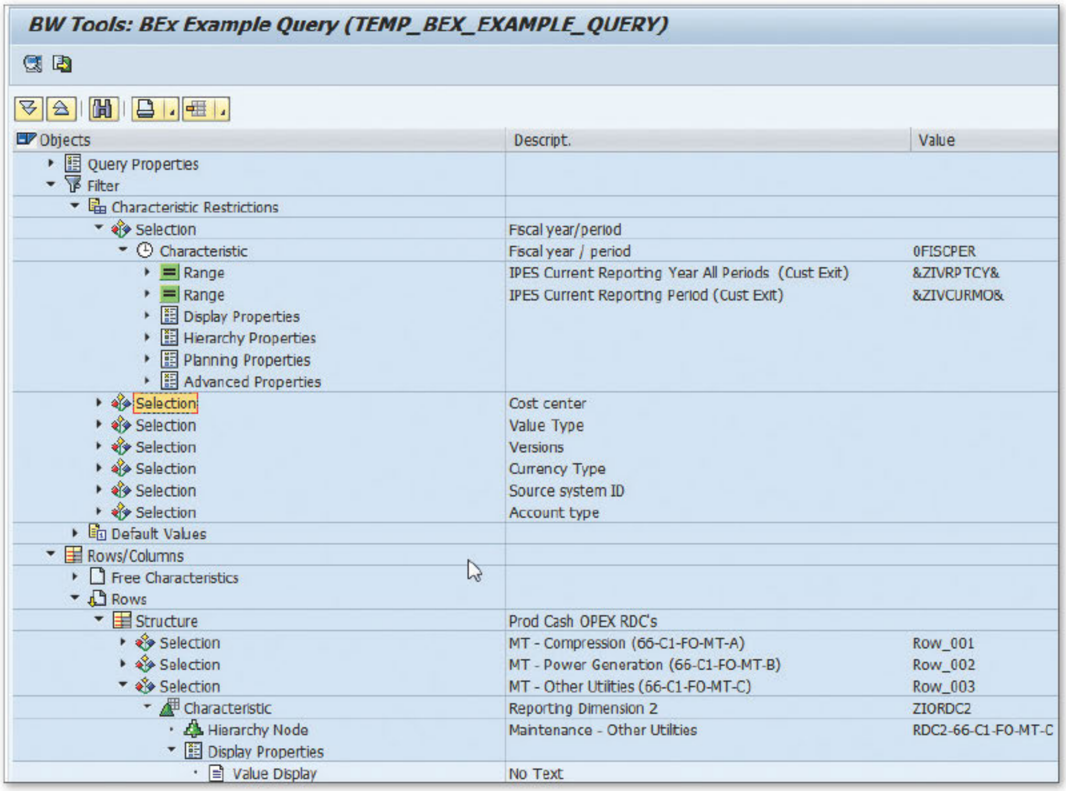
- ☐ Show Properties
- ☐ Expand Hierarchies
- ☐ Expand Key Figure Definition

Technical Information

[✓ Technical Names] Unique IDs Change Log

Figure 1 Selecting Output Options for the Query Definition

5. Based on options chosen in the selection screen, the program generates the query definition upon execution. Expand the various sections of the query definition for detailed analysis as shown in Figure 2.



BW Tools: BEx Example Query (TEMP_BEX_EXAMPLE_QUERY)

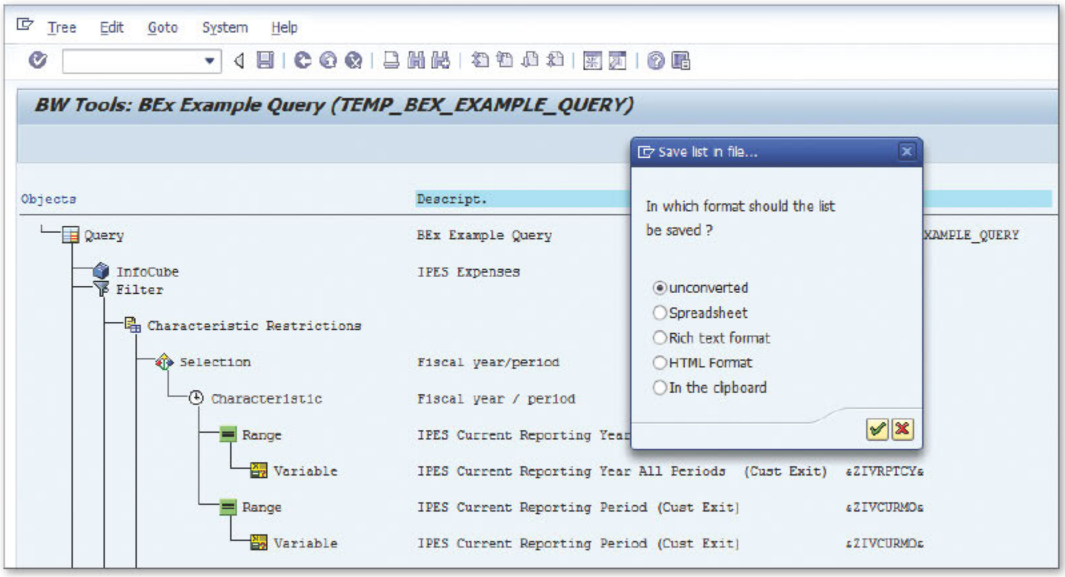
Objects	Descript.	Value
Query Properties		
Filter		
Characteristic Restrictions		
Selection	Fiscal year/period	
Characteristic	Fiscal year / period	0FISCPER
Range	IPES Current Reporting Year All Periods (Cust Exit)	&ZIVRPTCY&
Range	IPES Current Reporting Period (Cust Exit)	&ZIVCURMO&
Display Properties		
Hierarchy Properties		
Planning Properties		
Advanced Properties		
Selection	Cost center	
Selection	Value Type	
Selection	Versions	
Selection	Currency Type	
Selection	Source system ID	
Selection	Account type	
Default Values		
Rows/Columns		
Free Characteristics		
Rows		
Structure	Prod Cash OPEX RDC's	
Selection	MT - Compression (66-C1-FO-MT-A)	Row_001
Selection	MT - Power Generation (66-C1-FO-MT-B)	Row_002
Selection	MT - Other Utilities (66-C1-FO-MT-C)	Row_003
Characteristic	Reporting Dimension 2	ZIORDC2
Hierarchy Node	Maintenance - Other Utilities	RDC2-66-C1-FO-MT-C
Display Properties		
Value Display	No Text	

Figure 2 Expanded View of the Query Definition

Query Definition Download

Sometimes, you might be required to download the query definition to provide details to users or to create detailed documentation. To do this, follow these steps:

1. From the query definition output screen, click Download (F9), which generates a list view.
2. Choose System • List • Save • Local File to open the screen shown in Figure 3 for choosing the required file format to download the query definition.



BW Tools: BEx Example Query (TEMP_BEX_EXAMPLE_QUERY)

```
Objects                                Descript.
└─Query                                BEx Example Query                          XAMPLE_QUERY
  ├─InfoCube                           IPES Expenses
  ├─Filter
  │ ├─Characteristic Restrictions
  │ │ ├─Selection                      Fiscal year/period
  │ │ │ └─Characteristic               Fiscal year / period
  │ │ │   ├─Range                      IPES Current Reporting Year
  │ │ │   │ └─Variable                 IPES Current Reporting Year All Periods  (Cust Exit)  &ZIVRPTCY&
  │ │ │   └─Range                      IPES Current Reporting Period (Cust Exit)             &ZIVCURMO&
  │ │ │     └─Variable                 IPES Current Reporting Period (Cust Exit)             &ZIVCURMO&
```

Save list in file...

In which format should the list be saved ?

- [✓ unconverted]
- Spreadsheet
- Rich text format
- HTML Format
- In the clipboard

Figure 3 Download Options for the Query Definition

Part 5

Integration

Things You'll Learn in this Section

This part of the book will cover tips and tricks that you can use in areas of SAP NetWeaver BW integration such as SAP Data Services, SAP BusinessObjects Business Intelligence tools, Integrated Planning, and SAP HANA. Integrating SAP NetWeaver BW with tools in other systems, or even with other platforms provides business experts with the ability to add more flexibility and functionality to modeling and planning scenarios, as well as with other tasks.

We also touch on the integration of SAP NetWeaver BW models with SAP HANA models, which may be uncharted territory for you.

Tip 87

Making SAP NetWeaver BW Integrated Planning Formulas in Queries Input-Ready

You can create an input-ready formula by using the Inverse formula option in the BEx Query Designer.

In SAP NetWeaver BW Integrated Planning, it's common to use formulas for a lot of the key figures that are used in the planning process. The most common examples are any key figures involving calculations with prices (Price * Sales Quantity). Typically, the planner changes the basic key figures and checks the formula results. However, in some cases the planner may want to change the results of formula calculations to update the basic key figures. To make it possible to enter values for formulas, you can make the formulas input-ready.

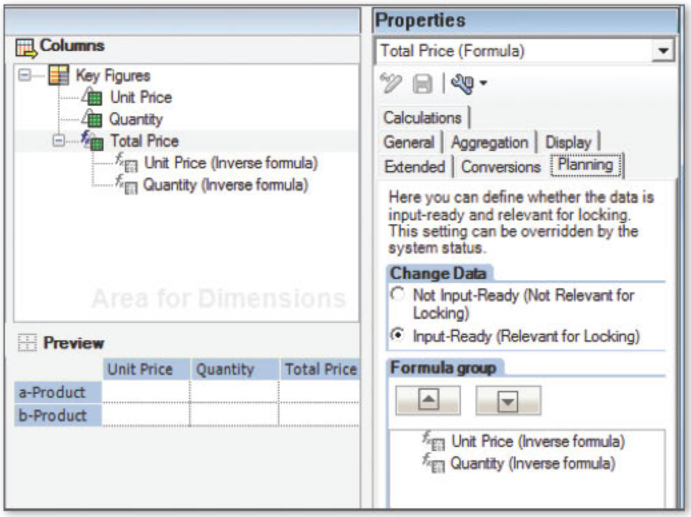
This tip shows how to create an input-ready formula by using the INVERSE FORMULA option in the Query Designer. This is particularly useful when there are scenarios where planning in calculated key figures (i.e., percentage, average price, etc.) is preferred.

✓ And Here's How ...

In this tip, the underlying basic key figures (the one found in the cube) are unit price and quantity. The product of the two provides the total price for each category of product. The requirement is to be able to plan on the result of the calculation (formula); that is, the total price.

Follow these simple steps to make the formula input-ready:

1. Create a formula (Total Price) and make it input ready by clicking the Input-Ready (Relevant for Locking) radio button in the Change Data area (see Figure 1).



	Unit Price	Quantity	Total Price
a-Product			
b-Product			

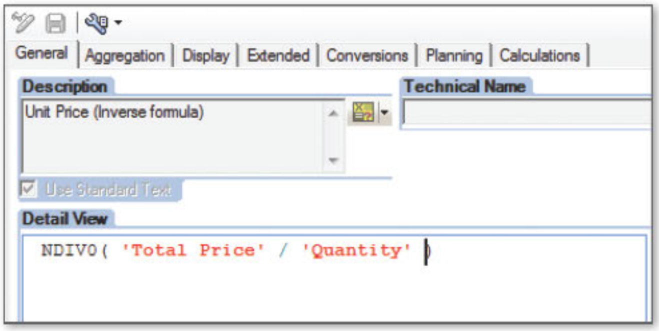
Here you can define whether the data is input-ready and relevant for locking. This setting can be overridden by the system status.

Change Data
Not Input-Ready (Not Relevant for Locking) [✓ Input-Ready (Relevant for Locking)]

Formula group
Unit Price (Inverse formula)
Quantity (Inverse formula)

Figure 1 Creating the Formula for Total Price

2. Define how the basic key figures will be calculated out of this formula by going to the PROPERTIES area of each basic key figure (see Figure 2). The order with which you place the subformulas Unit Price and Quantity determines which basic key figure will be calculated first as a result of a change in the input-ready formula (Total Price). For Unit Price, enter "NDIVO" (Total Price/Quantity) and for Quantity, enter "NDIVO" (Total Price/Unit Price).



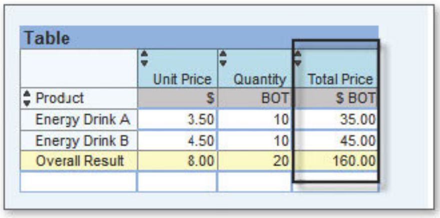
Description: Unit Price (Inverse formula)

Detail View

```
NDIV0( 'Total Price' / 'Quantity' )
```

Figure 2 Defining the Inverse Formula for Unit Price

3. Executing the query in BEx shows you the formula calculation as a result of the values in the UNIT PRICE and QUANTITY key figures. Note that the formula TOTAL PRICE is input ready (see Figure 3).

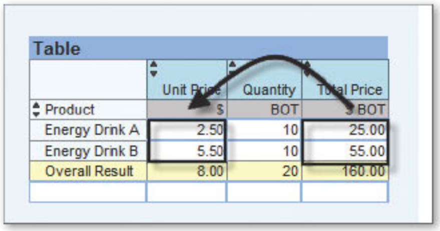


Table

Product	Unit Price	Quantity	Total Price
	$	BOT	$ BOT
Energy Drink A	3.50	10	35.00
Energy Drink B	4.50	10	45.00
Overall Result	8.00	20	160.00

Figure 3 Query Result Showing the Input-Ready Formula Total Price

4. Change the values of the TOTAL PRICE and refresh the page to trigger the change of the UNIT PRICE (see Figure 4).



Table

Product	Unit Price	Quantity	Total Price
	$	BOT	$ BOT
Energy Drink A	2.50	10	25.00
Energy Drink B	5.50	10	55.00
Overall Result	8.00	20	160.00

Figure 4 Changing the Total Price Changes the Unit Price

Tip 88

Creating a Postable Node Hierarchy in SAP NetWeaver BW Integrated Planning

You can create a postable node to allow a user to post at any level of a hierarchy for Integrated Planning purposes or for use as a buffer for top-down distribution.

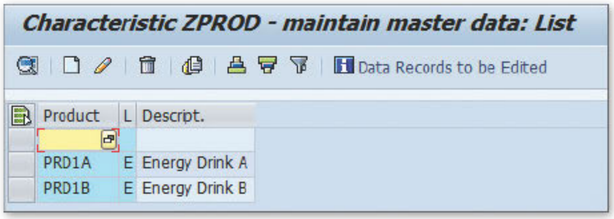
In SAP NetWeaver BW Integrated Planning implementations, it's common to perform hierarchy-based planning such as product planning on a product group hierarchy, expense planning on a cost center hierarchy, or sales planning on a geographical hierarchy. In some of these cases, the requirement is to post the plan numbers at the hierarchy node level instead of the characteristic itself. For instance, if you are doing expense planning based on a cost center hierarchy, you'll be able to enter plan numbers only at the cost center level by default, which is the lowest level of the hierarchy. However, some business scenarios require flexibility to post the plan numbers at any level of the hierarchy (any of the cost center hierarchy nodes). Also, in some scenarios, top-down distribution doesn't require the distribution of the whole amount that a node has, so the postable node can serve as the buffer for the undistributed amount.

This tip shows how to create a hierarchy with postable nodes to support planning scenarios in Integrated Planning that require posting to hierarchy nodes.

✓ And Here's How ...

In this tip, we'll use a characteristic called product (technical name = ZPROD) and a simple hierarchy based on this characteristic. First we'll show how the hierarchy would behave before we create postable nodes so that we can do a before and after comparison of a nonpostable hierarchy and a postable one.

Figure 1 shows a master data screenshot of the product before the creation of a postable hierarchy.

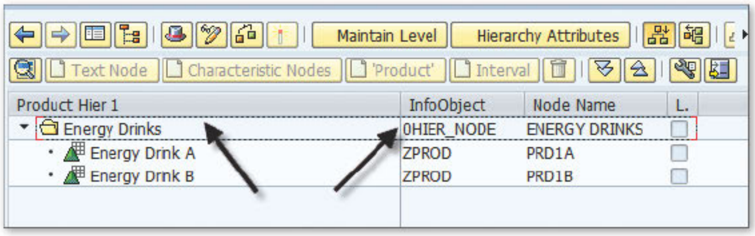

Characteristic ZPROD - maintain master data: List

Data Records to be Edited

Product	L	Descript.
PRD1A	E	Energy Drink A
PRD1B	E	Energy Drink B

« Figure 1 Master Data before Creating a Postable Hierarchy

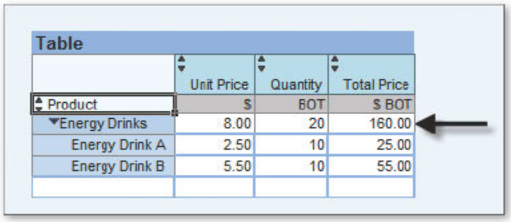
Figure 2 shows a screenshot of the nonpostable hierarchy from the backend of SAP NetWeaver BW. Note that the ENERGY DRINKS node is a text node, and the INFOOBJECT value is OHIER_NODE.



Product Hier 1	InfoObject	Node Name	L.
Energy Drinks	0HIER_NODE	ENERGY DRINKS	☐
Energy Drnk A	ZPROD	PRD1A	☐
Energy Drnk B	ZPROD	PRD1B	☐

⮝ Figure 2 Simple Hierarchy on the Characteristic ZPROD

Figure 3 shows a screenshot of the query running out of the nonpostable hierarchy. Note that the ENERGY DRINKS line is not input-ready.



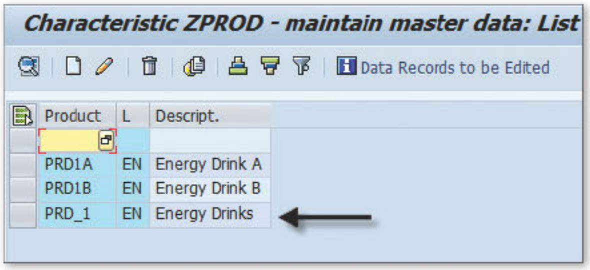
Table

	Unit Price	Quantity	Total Price
Product	$	BOT	$ BOT
Energy Drinks	8.00	20	160.00
Energy Drink A	2.50	10	25.00
Energy Drink B	5.50	10	55.00

« Figure 3 The Not Input-Ready Energy Drinks Node

With this scenario, you can execute the following simple steps to make the hierarchy node ENERGY DRINKS input-ready:

1. Using the master data maintenance screen in the SAP NetWeaver BW system, add the postable nodes (i.e., ENERGY DRINKS) as a regular value in the master data table of the characteristic product (see Figure 4).

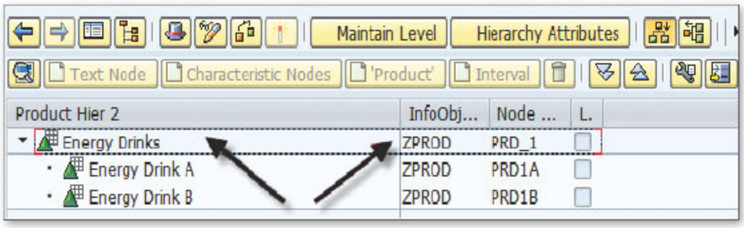


Characteristic ZPROD - maintain master data: List

Product	L	Descript.
PRD1A	EN	Energy Drink A
PRD1B	EN	Energy Drink B
PRD_1	EN	Energy Drinks

Figure 4 Master Data Maintenance Screen

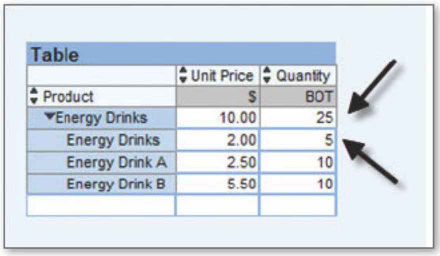
2. Create a new postable node hierarchy by creating the parent-child relationships between the various master data values of PRODUCT. Note that node ENERGY DRINKS no longer has the symbol of a text node (see Figure 5).



Product Hier 2	InfoObj...	Node ...	L.
Energy Drinks	ZPROD	PRD_1	☐
Energy Drink A	ZPROD	PRD1A	☐
Energy Drink B	ZPROD	PRD1B	☐

Figure 5 Creating a Hierarchy Node

3. Executing the query in BEx reveals one more line below the ENERGY DRINKS text node that has the same name and is input-ready (see Figure 6).



Table

	Unit Price	Quantity
Product	$	BOT
Energy Drinks	10.00	25
Energy Drinks	2.00	5
Energy Drink A	2.50	10
Energy Drink B	5.50	10

Figure 6 Input-Ready Hierarchy Node

Using these simple steps, you can make any hierarchy node input-ready.

Tip 89

Triggering a Report to Create Dynamic Measures for BEx Key Figures in OLAP Universes

You can use dynamic measure objects in an SAP BusinessObjects OLAP universe to dynamically select BEx key figures in SAP BusinessObjects Web Intelligence.

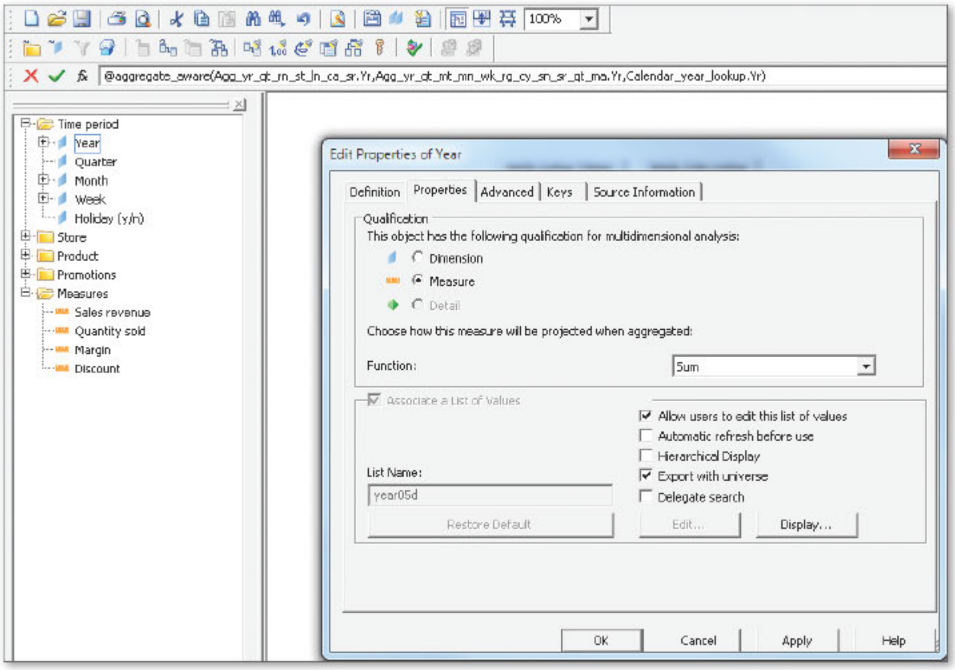
When creating SAP BusinessObjects Web Intelligence reports from an SAP BusinessObjects OLAP universe, a user may want to dynamically select BEx key figures during report runtime. This can be very useful in the reporting requirements scenarios where there is a need to create many identical reports in which the only difference is the measure. For example, you might want to build reports on measures such as gross expense, net expense, gross margin, net margin, and so on, but the report layout is the same. In this case, you can create multiple reports for these measures. The approach we discuss in this tip will help minimize the number of reports that you're required to build and maintain.

In this tip, we'll explain how you can set up the system so that a user can dynamically select a BEx key figure via a dynamic measure object in an SAP BusinessObjects OLAP universe, triggering a report runtime prompt in the SAP BusinessObjects Web Intelligence report.

✓ And Here's How ...

To trigger a runtime report to produce a list of key figures, follow these steps:

1. First, within the SAP BusinessObjects OLAP universe, create a new measure object as shown in Figure 1.



```
@aggregate_aware(Agg_yr_qt_rn_st_ln_ca_sr.Yr,Agg_yr_qt_mt_mn_wk_rg_cy_sn_sr_qt_ma.Yr,Calendar_year_lookup.Yr)
```

- Time period
  - Year
  - Quarter
  - Month
  - Week
  - Holiday (y/n)
- Store
- Product
- Promotions
- Measures
  - Sales revenue
  - Quantity sold
  - Margin
  - Discount

Edit Properties of Year

Definition | Properties | Advanced | Keys | Source Information

Qualification

This object has the following qualification for multidimensional analysis:

( ) Dimension [✓ Measure] ( ) Detail

Choose how this measure will be projected when aggregated:

Function: Sum

[✓] Associate a List of Values

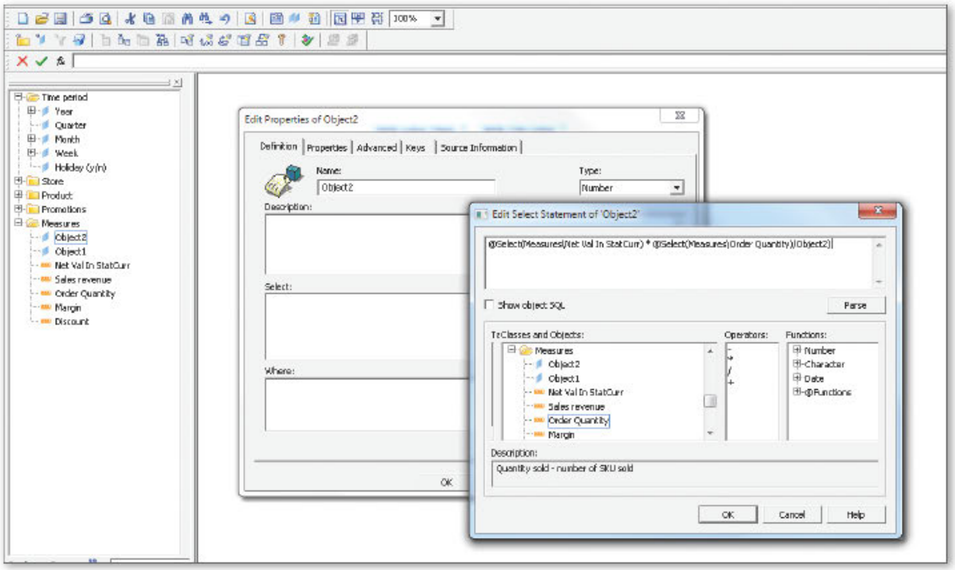
List Name: year05d

[✓] Allow users to edit this list of values
[ ] Automatic refresh before use
[ ] Hierarchical Display
[✓] Export with universe
[ ] Delegate search

Figure 1 SAP BusinessObjects OLAP Universe Screen

2. You can define the following properties for the measure object:
   - Define the aggregation FUNCTION as SUM.
   - Check the boxes for ASSOCIATE A LIST OF VALUES and ALLOW USERS TO EDIT THIS LIST OF VALUES. These settings will allow users to dynamically change the value of the measure.

3. Modify the SELECT object property with the script syntax shown in Figure 2.



Name: Object2

Type: Number

```
@Select(Measures\Net Val In StatCurr) * @Select(Measures\Order Quantity)/Object2)
```

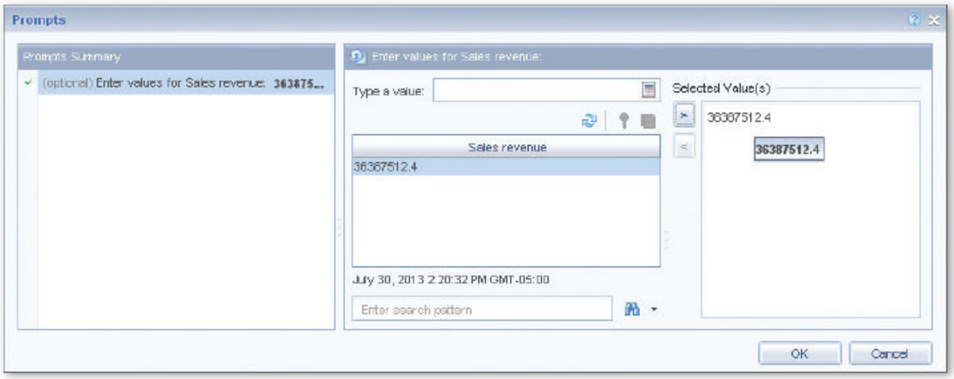
Description: Quantity sold - number of SKU sold

Figure 2 Entering Code for the Objects

4. Enter the following code in the EDIT SELECT STATEMENT OF OBJECT1 box:

```
<EXPRESSION>
(@Prompt('Enter Period','A:A',{'QTD':'[Measures].
[006EINA83OW2SPU1QR9CA3R4M]','Q1 To Date':'[Measures].
[006EINA83OW2SPU1QR9CA3XG6]','Q2 To Date':'[Measures].
[006EINA83OW2SPU1QR9CA43RQ]','Q3 To Date':'[Measures].
[006EINA83OW2SPU1QR9CA4A3A]','Q4 To Date':'[Measures].
[006EINA83OW2SPU1QR9CA4GEU]'},Mono,Primary_Key))
</EXPRESSION>
```

5. Export the SAP BusinessObjects OLAP universe to the central repository.
6. Create a new WebI report with a dynamic measure object as shown in Figure 3.



(optional) Enter values for Sales revenue: 363875...

Type a value:

Selected Value(s): 36387512.4

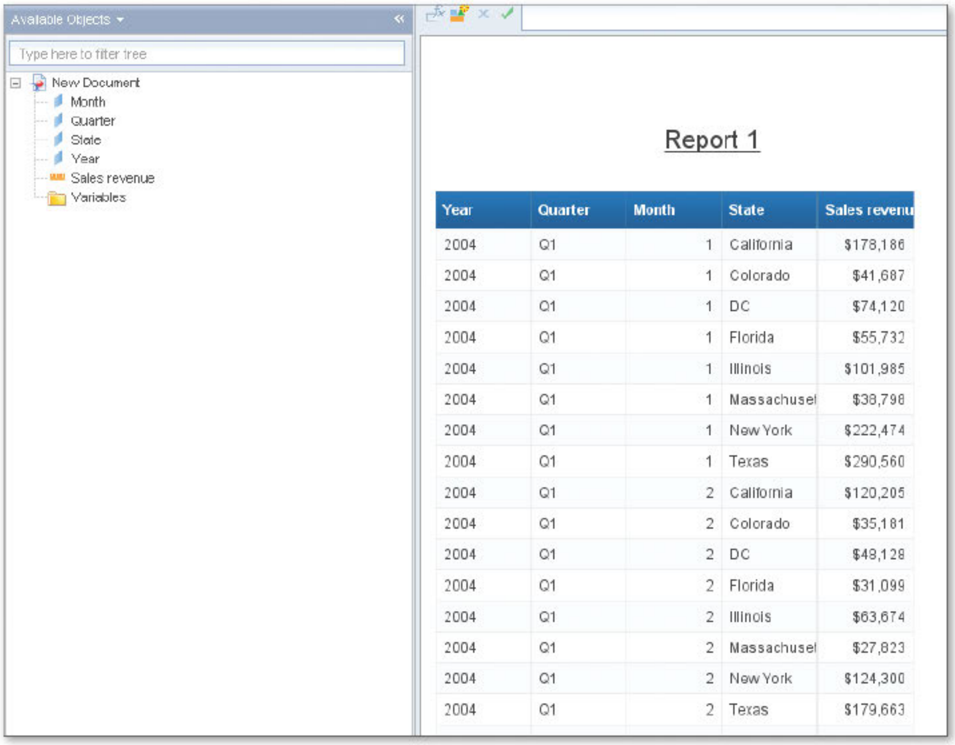
Sales revenue
36387512.4

July 30, 2013 2:20:32 PM GMT-05:00

Enter search pattern

Figure 3 Web Intelligence Report Creation

7. Execute the report and select the desired key figure from the dynamic prompt. The report is generated with key figure selection as shown in Figure 4.



Report 1

Year	Quarter	Month	State	Sales revenu
2004	Q1	1	California	$178,186
2004	Q1	1	Colorado	$41,687
2004	Q1	1	DC	$74,120
2004	Q1	1	Florida	$55,732
2004	Q1	1	Illinois	$101,985
2004	Q1	1	Massachuset	$38,798
2004	Q1	1	New York	$222,474
2004	Q1	1	Texas	$290,560
2004	Q1	2	California	$120,205
2004	Q1	2	Colorado	$35,181
2004	Q1	2	DC	$49,128
2004	Q1	2	Florida	$31,099
2004	Q1	2	Illinois	$63,674
2004	Q1	2	Massachuset	$27,823
2004	Q1	2	New York	$124,300
2004	Q1	2	Texas	$179,663

Figure 4 Report Results

Tip 90

Invoking an SAP Data Services Job Using Process Chains in SAP NetWeaver BW

You can load non-SAP data into SAP NetWeaver BW via mass data loading jobs by integrating SAP Data Services into process chains.

If you need to load non-SAP data into SAP NetWeaver BW, you can use the SAP Data Services tool. However, integrating the SAP Data Services job into the overall SAP NetWeaver BW data-loading process (process chains) can be difficult to do correctly. In SAP NetWeaver BW, process chains enable the automation of the entire data-loading process, as well as perform central control and monitor the data-loading processes. If you don't integrate the SAP Data Services job into the process chains, then you'll have to monitor and manage all of the SAP Data Services data-loading jobs separately, which can be a major overhead on the SAP NetWeaver BW support team.

The key for integration of the SAP Data Services job to the process chains is to have the ability to invoke the SAP Data Services data load job within SAP NetWeaver BW via the InfoPackage, and then the InfoPackage can be included in any process chain. This tip describes how to trigger an SAP Data Services data load job using an InfoPackage in SAP NetWeaver BW for a non-SAP source.

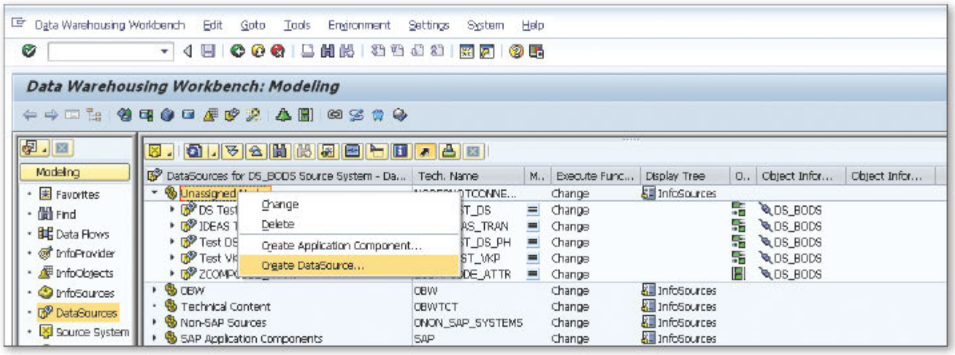
And Here's How ...

In this tip, we assume that you already have an RFC connection created between SAP NetWeaver BW and SAP Data Services. The entire process involves four steps, which you can view in the following sections.

Step 1

Create a DataSource in SAP NetWeaver BW's Data Warehousing Workbench (DWB), and use SAP Data Services as the source system. Follow these steps:

1. Log in to the SAP NetWeaver BW system and in the Administrator Workbench, go to the DATASOURCES option. Right-click on the application component (here we choose UNASSIGNED), and click on CREATE DATASOURCE as shown in Figure 1.



DataSources for DS_BODS Source System - Da...	Tech. Name	M..	Execute Func...	Display Tree	O..	Object Infor...	Object Infor...
Unassigned	TCONNE...		Change	InfoSources			
DS Test	T_DS		Change			DS_BODS	
IDEAS T	AS_TRAN		Change			DS_BODS	
Test DS	T_DS_PH		Change			DS_BODS	
Test VK	ST_VKP		Change			DS_BODS	
ZCOMP	DE_ATTR		Change			DS_BODS	
OBW	OBW		Change	InfoSources			
Technical Content	OBWTCT		Change	InfoSources			
Non-SAP Sources	ONON_SAP_SYSTEMS		Change	InfoSources			
SAP Application Components	SAP		Change	InfoSources			

Figure 1 Creating a DataSource in SAP NetWeaver BW with SAP Data Services as the Source System

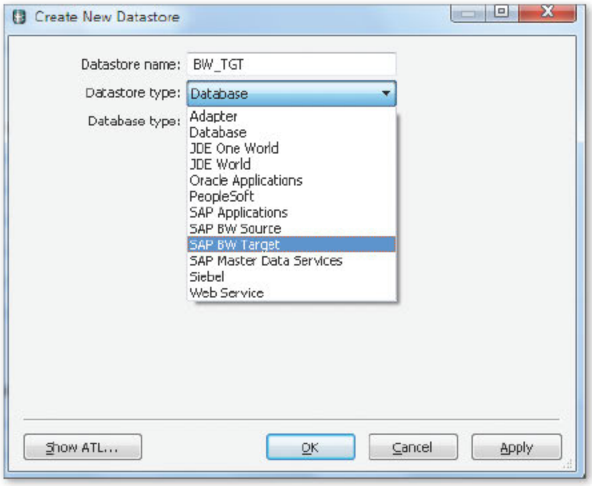
2. In the CREATE DATASOURCE popup, enter the DataSource name, source system, and data type, and then click CONTINUE.
3. In the next screen, input the InfoObject names and then save and activate the DataSource.

Step 2

Import the DataSource created in SAP NetWeaver BW into SAP Data Services. Follow these steps:

1. Log in to SAP Data Services, and create a connection DataStore with SAP NetWeaver BW as the target. Right-click in the DATASTORE pane, and click NEW.

2. The Create New DataStore popup appears requesting the Datastore name, Datastore type, and so on. Input a DataStore name, and select SAP BW Target from the DataStore Type dropdown as shown in Figure 2.



Create New Datastore

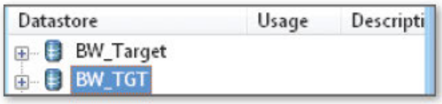
Datastore name: BW_TGT

Datastore type: Database

Database type:

« *Figure 2 Entering theDatastore Name and Selecting SAP BW Target from the Datastore Type Dropdown*

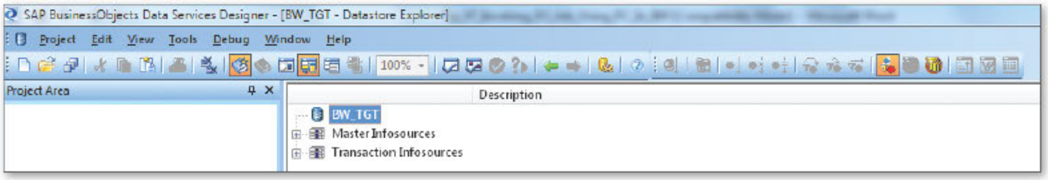
3. On the next popup screen, enter the database server name, user name, and password.
4. Click on the Advanced button on the screen, enter the client number of the SAP NetWeaver BW system, and click OK.
5. A new DataStore is created in the local object library as shown in Figure 3.



Datastore	Usage	Descripti
BW_Target		
BW_TGT		

« *Figure 3 New Datastore in the Library*

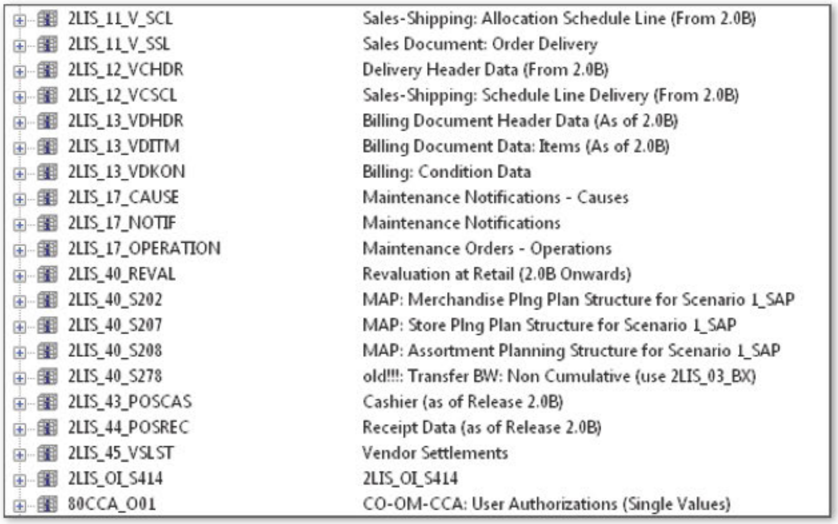
6. Import the DataSource created in SAP NetWeaver BW to SAP Data Services. Double-click on the Datastore created, and a screen opens on the right side pane as shown in Figure 4.



SAP BusinessObjects Data Services Designer - [BW_TGT - Datastore Explorer]

Figure 4 Displaying the Transaction InfoSources Tree

7. Expand the transaction InfoSources tree and search for the DataSource that you created in step 1.
8. Click on the + sign beside the DataSource, and click on IMPORT (see Figure 5).



2LIS_11_V_SCL	Sales-Shipping: Allocation Schedule Line (From 2.0B)
2LIS_11_V_SSL	Sales Document: Order Delivery
2LIS_12_VCHDR	Delivery Header Data (From 2.0B)
2LIS_12_VCSCL	Sales-Shipping: Schedule Line Delivery (From 2.0B)
2LIS_13_VDHDR	Billing Document Header Data (As of 2.0B)
2LIS_13_VDITM	Billing Document Data: Items (As of 2.0B)
2LIS_13_VDKON	Billing: Condition Data
2LIS_17_CAUSE	Maintenance Notifications - Causes
2LIS_17_NOTIF	Maintenance Notifications
2LIS_17_OPERATION	Maintenance Orders - Operations
2LIS_40_REVAL	Revaluation at Retail (2.0B Onwards)
2LIS_40_S202	MAP: Merchandise Plng Plan Structure for Scenario 1_SAP
2LIS_40_S207	MAP: Store Plng Plan Structure for Scenario 1_SAP
2LIS_40_S208	MAP: Assortment Planning Structure for Scenario 1_SAP
2LIS_40_S278	old!!!: Transfer BW: Non Cumulative (use 2LIS_03_BX)
2LIS_43_POSCAS	Cashier (as of Release 2.0B)
2LIS_44_POSREC	Receipt Data (as of Release 2.0B)
2LIS_45_VSLST	Vendor Settlements
2LIS_OI_S414	2LIS_OI_S414
80CCA_O01	CO-OM-CCA: User Authorizations (Single Values)

Figure 5 Importing DataSource

9. The DataSource is now imported to the SAP NetWeaver BW target DATASTORE in SAP Data Services (you can check by expanding the transaction tree structure under DATASTORE.

Step 3

Create the SAP Data Services job. Create a job in SAP Data Services with the preceding DataSource as the target.

Step 4

Implement the InfoPackage in SAP NetWeaver BW for the the preceding job created in SAP Data Services. Log into SAP NetWeaver BW and create an InfoPackage for the DataSource created in step 1.

1. In the THIRD-PARTY SELECTIONS tab of the InfoPackage, enter SAP Data Services parameter details such as REPOSITORY, JOBSERVER, and JOBNAME.
2. Save and activate the InfoPackage.
3. Create a process chain with the previously created InfoPackage as shown in Figure 6 and execute the process chain.

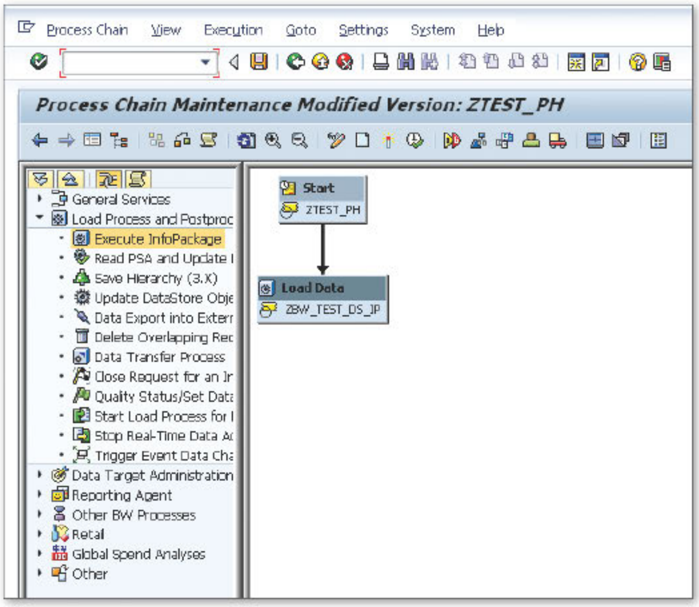


Figure 6 Process Chain for Use with the InfoPackage for the SAP Data Services Job

After successful completion of the process chain, data will be available in the PSA (Persistent Staging Area). This way, you can invoke the SAP Data Services data-loading job using an InfoPackage in SAP NetWeaver BW and include the InfoPackage in the process chains for integrated control and monitoring of data loads.

Tip 91

Importing SAP NetWeaver BW BEx Queries to an SAP HANA Modeling Environment

To enable end users to search for business information in a combined SAP NetWeaver BW/SAP HANA environment, you can import SAP NetWeaver BW BEx queries into SAP HANA.

End users can use SAP BusinessObjects Explorer to search for and explore business information without needing to create specific reports. In a traditional SAP NetWeaver BW environment, the SAP BusinessObjects Explorer tool is able to report on BW data via a universe created on the SAP NetWeaver BW InfoProviders.

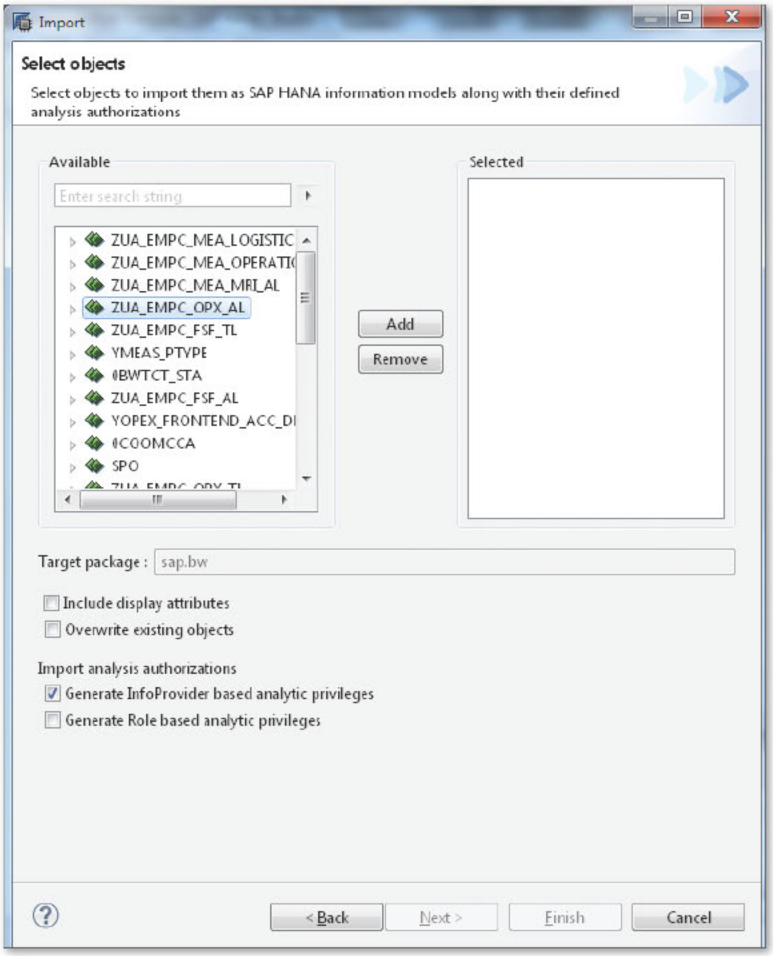
With the advent of SAP HANA, however, SAP has released SAP BusinessObjects Explorer for SAP NetWeaver BW on HANA, which is based on SAP HANA architecture, and can't connect to SAP NetWeaver BW InfoProviders and access the data directly. This is a major pain point because an SAP BusinessObjects Explorer business user will not be able to leverage the SAP NetWeaver BW functionality completely if the MultiProviders or BEx queries cannot be imported into the SAP HANA modeling environment.

This tip describes an approach to import an SAP NetWeaver BW BEx query to the SAP HANA modeling environment. Because MultiProviders cannot be imported directly into the SAP HANA modeling environment, the best approach is to create a BEx query of the MultiProvider with all of the required data and key figure fields included in the query.

✓ And Here's How ...

For the purpose of this tip, it is assumed that the BEx query is already created and available to import into SAP HANA. The major task on the SAP NetWeaver BW side is to create a query snapshot in SAP NetWeaver BW and create an SAP HANA index on the query snapshot. Follow these steps:

1. Execute Transaction RSDDB in the SAP NetWeaver BW system. On the resulting screen, enter the query name and click on RELEASE QUERY AS INFOPROVIDER
2. Create an index by entering the query name and clicking the CREATE button. This launches the index maintenance screen where you select the table and click on ACTIVATE AND FILL INDEX. After activating and filling the index is completed, INDEX STATUS will appear as green.
3. Log on to SAP HANA Studio to import the query as a snapshot that we created in the previous step.
4. Import the SAP NetWeaver BW model by clicking FILE • IMPORT. On the IMPORT screen, click the SAP HANA CONTENT folder, and choose the IMPORT SAP NETWEAVER BW MODELS option. Then click NEXT.
5. Select a source system from where SAP NetWeaver BW models will be imported by following these steps:
   - Create a connection to SAP NetWeaver BW on HANA application server (ABAP based) by providing all relevant information such as SERVER NAME, HOST NAME, INSTANCE NUMBER, CLIENT NUMBER, USER NAME, and PASSWORD. Click NEXT.
   - If a connection already exists, then select a connection and provide a USER NAME and PASSWORD to login to the SAP NetWeaver BW system to import an SAP NetWeaver BW model.
   - Click NEXT.
6. Select the TARGET folder where you want to import SAP NetWeaver BW models by entering the TARGET PACKAGE on the screen. In this case, we will use the TARGET SAP.BW.
7. As shown in Figure 1, navigate to the relevant info area and move left to right by selecting the SAP NetWeaver BW model and clicking on the ADD button. Click NEXT and FINISH. Note that imported models will be created as an SAP HANA model (analytic or calculation view) under the TARGET PACKAGE SAP.BW.



Target package : sap.bw

☐ Include display attributes
☐ Overwrite existing objects

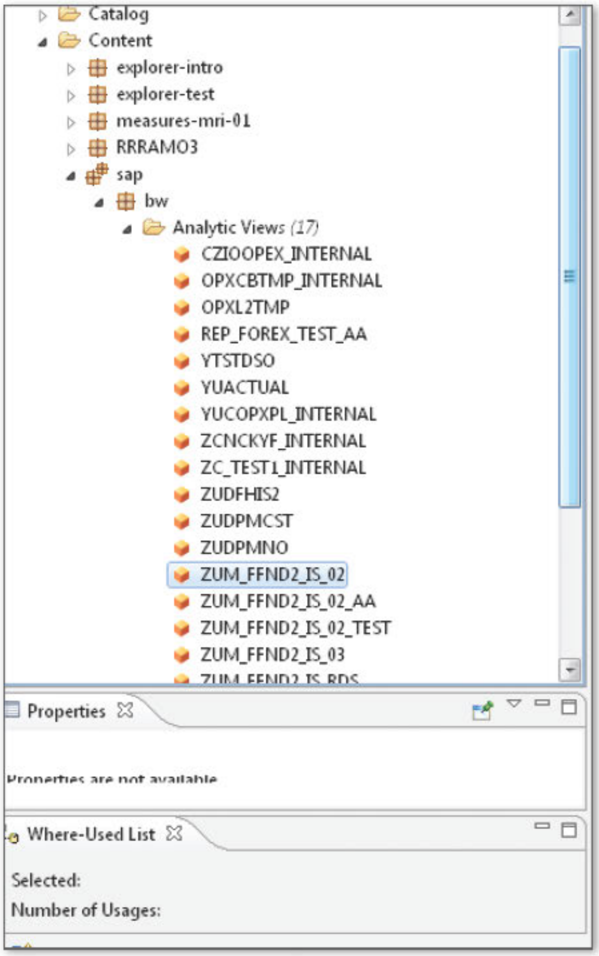
Import analysis authorizations

☑ Generate InfoProvider based analytic privileges
☐ Generate Role based analytic privileges

Figure 1 Select the InfoProvider from the InfoArea

8. Select the OVERWRITE EXISTING OBJECTS checkbox if this model was previously imported and has changes to overwrite. (If the previously imported model, which is an SAP HANA model, has any local changes, then those changes are removed and a new SAP HANA model is recreated.)
9. The SAP HANA model is created and activated. Figure 2 on the SAP HANA environment displays the analytical views that are included in the folder SAP.BW. In the earlier step, we indicated that the TARGET PACKAGE should be SAP.BW, so based on all of the preceding steps, the analytical view will be generated for the

query snapshot that we imported from the SAP NetWeaver BW system, and the analytical views are organized under this folder.



```
▷ Catalog
◢ Content
    ▷ explorer-intro
    ▷ explorer-test
    ▷ measures-mri-01
    ▷ RRRAMO3
    ◢ sap
        ◢ bw
            ◢ Analytic Views (17)
                CZIOOPEX_INTERNAL
                OPXCBTMP_INTERNAL
                OPXL2TMP
                REP_FOREX_TEST_AA
                YTSTDSO
                YUACTUAL
                YUCOPXPL_INTERNAL
                ZCNCKYF_INTERNAL
                ZC_TEST1_INTERNAL
                ZUDFHIS2
                ZUDPMCST
                ZUDPMNO
                ZUM_FFND2_IS_02
                ZUM_FFND2_IS_02_AA
                ZUM_FFND2_IS_02_TEST
                ZUM_FFND2_IS_03
                ZUM_FFND2_IS_RDS
```

Properties

Properties are not available

Where-Used List

Selected:
Number of Usages:

Figure 2 Imported BEx Query Displayed as an SAP HANA View

After the import of the query as a snapshot as SAP HANA views, these views can be exposed to SAP BusinessObjects Explorer for business users to search and explore on the SAP NetWeaver data.

Tip 92

Enabling the Use of SAP BusinessObjects Explorer in an SAP NetWeaver BW on SAP HANA Environment

You can expose SAP NetWeaver BW InfoProviders to SAP BusinessObjects Explorer to enable search and exploration functionality for end users to look for data that's relevant to their business.

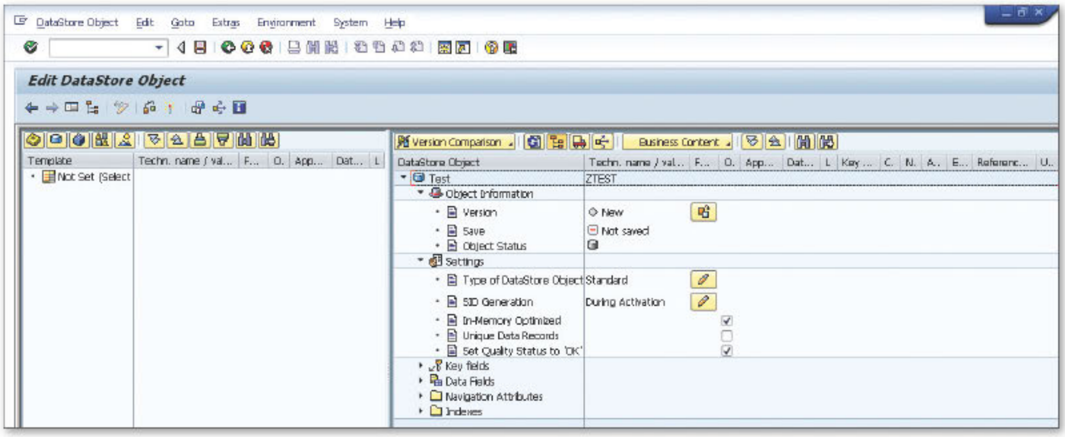
Business users often use SAP BusinessObjects Explorer to search for business information (based on SAP NetWeaver BW InfoProviders) without creating specific reports. However, in an SAP NetWeaver BW on SAP HANA scenario, SAP BusinessObjects Explorer is currently (at the time of this book's publication) not able to connect to SAP NetWeaver BW InfoProviders directly, so the only way to access the data is by using SAP HANA views.

This tip describes the steps you need to follow to expose SAP NetWeaver BW InfoProviders in an SAP NetWeaver BW on HANA scenario to SAP HANA views to enable reporting using SAP BusinessObjects Explorer.

✓ And Here's How ...

To illustrate this tip, we'll use the InfoProvider DSO as an example. The following steps are applicable to InfoCubes as well. Follow these steps to expose DSOs created in SAP NetWeaver BW to SAP BusinessObjects Explorer by creating an SAP HANA view on the DSO:

1. Execute Transaction RSA1 in the SAP NetWeaver BW system.
2. Create a DSO in SAP NetWeaver BW, and make sure that the In-Memory Optimized box is checked as shown in Figure 1. Note that you need to follow all of the other basic steps involved in creating a DSO such as defining key fields, data fields, indexes, and so on.

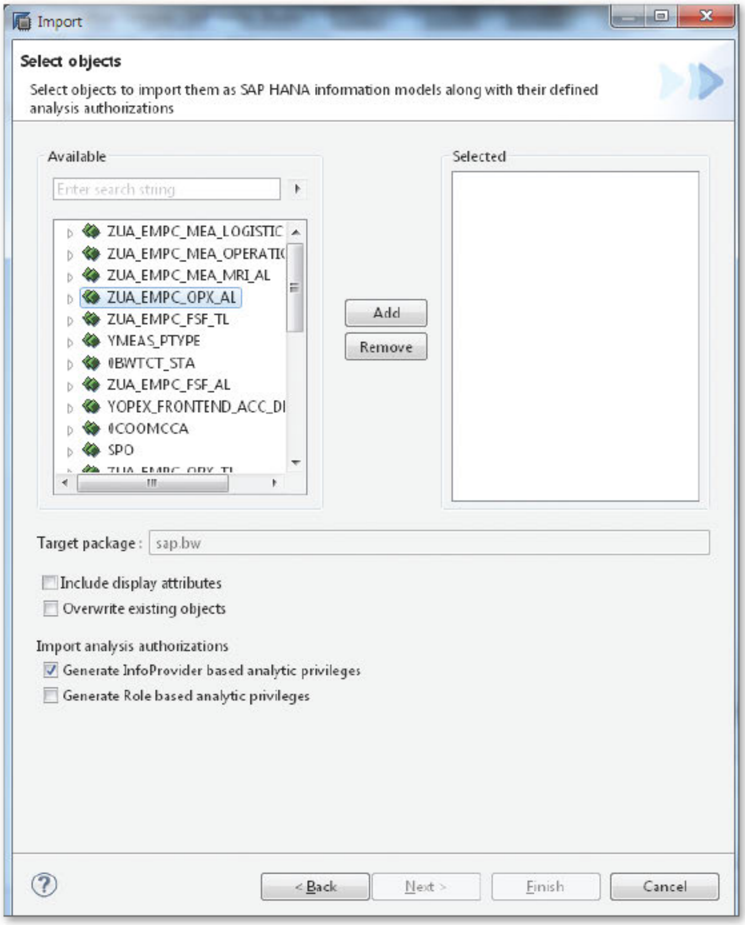


DataStore Object	Techn. name / val...
Test	ZTEST
Object Information	
Version	New
Save	Not saved
Object Status	
Settings	
Type of DataStore Object	Standard
SID Generation	During Activation
In-Memory Optimized	✓
Unique Data Records	☐
Set Quality Status to 'OK'	✓
Key fields	
Data Fields	
Navigation Attributes	
Indexes	

Figure 1 Creating In-Memory Optimized DSO

3. Log on to SAP HANA Studio.
4. Import the DSO created in the previous step by clicking File • Import. Select the SAP HANA Content folder, and then choose Import SAP NetWeaver BW Models. Click Next.
5. Select a source system from where the SAP NetWeaver BW DSO will be imported with the following options:
   - Create a connection to the SAP NetWeaver BW on SAP HANA application server (ABAP-based) by providing all relevant information, such as server name, host name, instance number, client number, user name, and password. Click Next.
   - If a connection already exists, then select a connection and provide a User Name and Password to log in to the SAP NetWeaver BW system to import the SAP NetWeaver BW DSO.
6. Navigate to the relevant InfoArea, and move the relevant available object from left to right by selecting the SAP NetWeaver BW model and clicking on the Add button. Click Next and Finish (see Figure 2). Note that imported models will

be created as an SAP HANA model (analytic or calculation view) under TARGET PACKAGE SAP.BW.


Import

Select objects

Select objects to import them as SAP HANA information models along with their defined analysis authorizations

Available

Enter search string

- ZUA_EMPC_MEA_LOGISTIC
- ZUA_EMPC_MEA_OPERATI(
- ZUA_EMPC_MEA_MRI_AL
- ZUA_EMPC_OPX_AL
- ZUA_EMPC_FSF_TL
- YMEAS_PTYPE
- 0BWTCT_STA
- ZUA_EMPC_FSF_AL
- YOPEX_FRONTEND_ACC_DI
- 0COOMCCA
- SPO

Selected

Target package : sap.bw

☐ Include display attributes

☐ Overwrite existing objects

Import analysis authorizations

☑ Generate InfoProvider based analytic privileges

☐ Generate Role based analytic privileges

Figure 2 Select the SAP NetWeaver BW DSO from the InfoArea

7. Select the OVERWRITE EXISTING OBJECTS checkbox if this model was previously imported and has changes to overwrite. (If the previously imported model, which is an SAP HANA model, has any local changes, then those changes are removed and a new SAP HANA model is recreated.)
8. The SAP HANA model will be created and activated. You can view the SAP HANA model by going to the SAP HANA system, choosing the content folder, and then clicking on SAP • BW and then the package (see Figure 3).

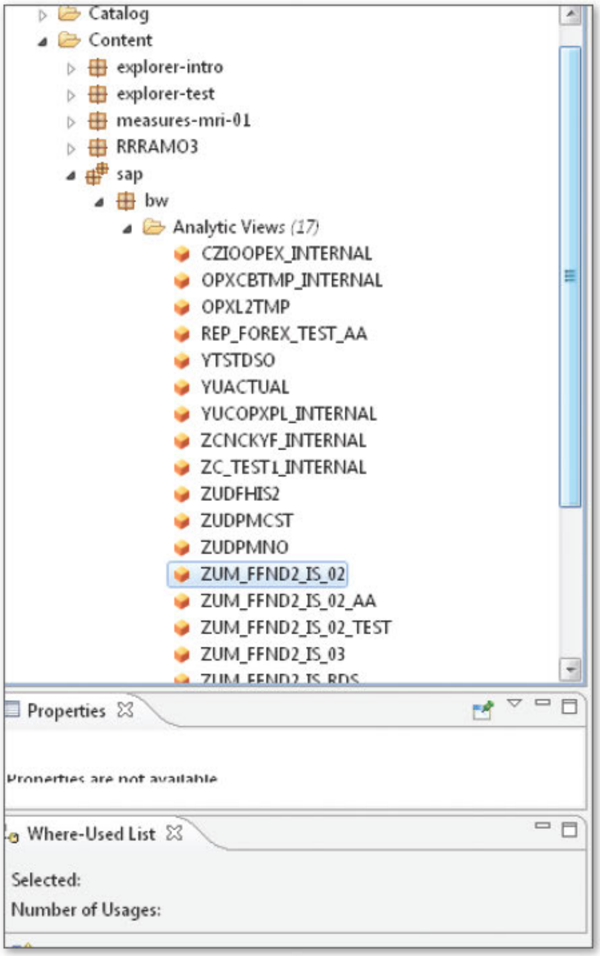


Figure 3 SAP HANA Model Analytical View Created for the DSO

You can create information spaces on the analytical view that we created in the previous step and make them available to a report using SAP BusinessObjects Explorer. Views that are created are available in BEx.

This method can be used to import an InfoCube or a DSO to the SAP HANA Studio as SAP HANA views and expose these SAP NetWeaver BW InfoProviders (InfoCubes and DSOs) to SAP BusinessObjects Explorer for searching and exploring business information.

Tip 93

Using a MultiProvider as a Source for Open Hub Destination

You can use the new Open Hub Destination functionality in SAP NetWeaver BW 7.3 to source data en masse from a MultiProvider.

When you have data from an SAP NetWeaver BW system that needs to be distributed to non-SAP data marts, you'll use the Open Hub Destination. However, one of the limitations of the Open Hub Destination is that it cannot source data from MultiProviders. This is a major drawback in scenarios where the requirement is to extract the data from multiple InfoProviders (InfoCubes or DSOs) that are part of a MultiProvider. The only way to extract data in this case is to source data separately for each of these InfoProviders. This requires creating multiple Open Hub Destination objects and the corresponding transformations and DTPs, resulting in additional development effort and also the need to maintain all of these objects. Additionally, the receiving system has to handle multiple data files and implement further logic to merge the data from these files.

This tip describes the functionality available with SAP NetWeaver BW 7.3 to use a MultiProvider as the source for Open Hub Destination, and thus source data from multiple InfoProviders using a single Open Hub Destination object.

And Here's How ...

For this scenario, we'll use an existing simple MultiProvider: ZTSTMP. Follow these steps:

1. Log in to SAP NetWeaver BW 7.3. Execute Transaction RSA1.
2. In the resulting Data Warehousing Workbench screen, double-click on Open Hub Destination.
3. Create a test Open Hub Destination object with the name "ZTSTOHB".
4. Select the Open Hub Destination object ZTSTOHB, right-click, and choose Create Transformation.
5. On the resulting screen, select MultiProvider from the Object Type dropdown in the Source of the Transformation section (see Figure 1).

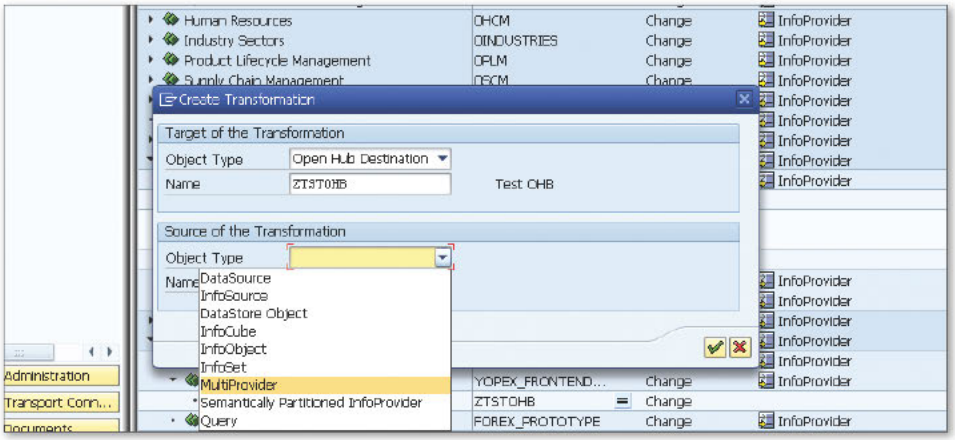
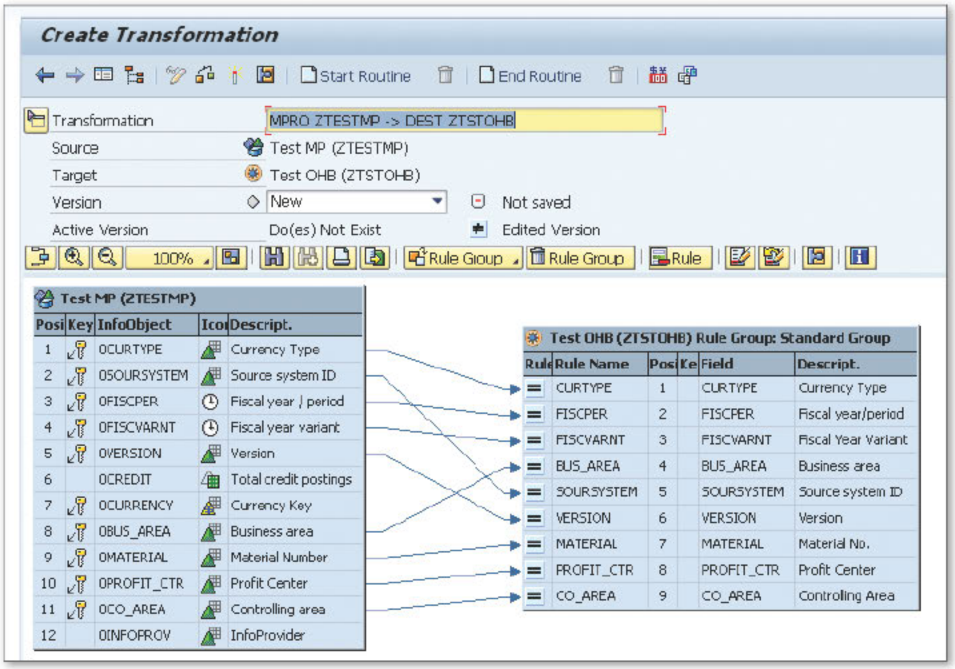


Figure 1 Select MultiProvider

6. Input the MultiProvider name ("ZTSTMP"), and click Continue.
7. Map the appropriate fields in the transformation and activate the transformation (see Figure 2). Create the DTP for the transformation.



Create Transformation

Transformation	MPRO ZTESTMP -> DEST ZTSTOHB
Source	Test MP (ZTESTMP)
Target	Test OHB (ZTSTOHB)
Version	New — Not saved
Active Version	Do(es) Not Exist — Edited Version

Test MP (ZTESTMP)

Posi	Key	InfoObject	Ico	Descript.
1	✓	0CURTYPE		Currency Type
2	✓	0SOURSYSTEM		Source system ID
3	✓	0FISCPER		Fiscal year / period
4	✓	0FISCVARNT		Fiscal year variant
5	✓	0VERSION		Version
6		0CREDIT		Total credit postings
7	✓	0CURRENCY		Currency Key
8	✓	0BUS_AREA		Business area
9	✓	0MATERIAL		Material Number
10	✓	0PROFIT_CTR		Profit Center
11	✓	0CO_AREA		Controlling area
12		0INFOPROV		InfoProvider

Test OHB (ZTSTOHB) Rule Group: Standard Group

Rul	Rule Name	Posi	Ke	Field	Descript.
=	CURTYPE	1		CURTYPE	Currency Type
=	FISCPER	2		FISCPER	Fiscal year/period
=	FISCVARNT	3		FISCVARNT	Fiscal Year Variant
=	BUS_AREA	4		BUS_AREA	Business area
=	SOURSYSTEM	5		SOURSYSTEM	Source system ID
=	VERSION	6		VERSION	Version
=	MATERIAL	7		MATERIAL	Material No.
=	PROFIT_CTR	8		PROFIT_CTR	Profit Center
=	CO_AREA	9		CO_AREA	Controlling Area

Figure 2 Defining Transformation Rules for the MultiProvider to Open Hub Destination

Using this approach, you can use any MultiProvider as a source for the Open Hub Destination. Using this Open Hub Destination, you can distribute the data in the underlying InfoProviders of the MultiProviders to other non-SAP systems.

One drawback of this approach is that only full data loads are supported at this time, so you have to use a time characteristic such as fiscal period or calendar month to restrict the data that is extracted from the MultiProvider.

Tip 94

Reporting on SAP HANA Models from SAP NetWeaver BW with VirtualProviders

You can easily report on SAP HANA with all of the existing SAP NetWeaver BW metadata in your system when you develop BEx reports on SAP HANA models using VirtualProviders.

If you're using SAP NetWeaver BW 7.3 on an SAP HANA platform, then in addition to building your reporting InfoProviders in a traditional SAP NetWeaver BW modeler, you also have the option to build SAP HANA analytical and attribute views using SAP HANA Studio. In some business scenarios, you may want to report the SAP HANA models using BEx queries or combine the SAP NetWeaver BW InfoProviders with the SAP HANA views for reporting, using BEx. Transient InfoProviders provide a good option in exposing SAP HANA models to SAP NetWeaver BW in an agile manner, but the major drawback with that option is that you can't use the SAP NetWeaver BW metadata such as InfoObjects, navigational attributes, and the analysis authorizations.

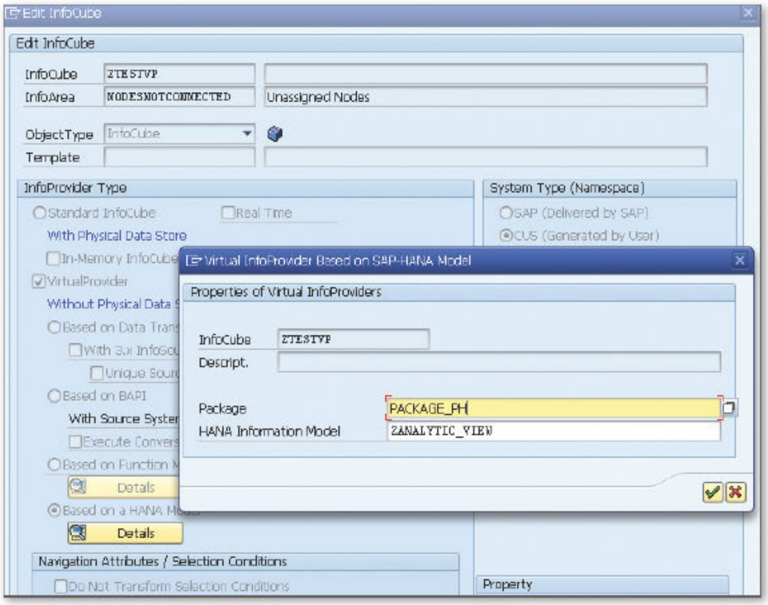
This tip describes an alternative method of exposing SAP HANA models in SAP NetWeaver BW using VirtualProviders on top of the SAP HANA models. The advantage of a VirtualProvider is that it's treated as any other InfoProvider in SAP NetWeaver BW, so it uses SAP NetWeaver BW metadata and can be combined with other InfoProviders using MultiProviders for combined reporting.

✓ And Here's How ...

For the purpose of this tip, it's assumed that you've already created an SAP HANA model (analytical view) in SAP HANA Studio named ZANALYTIC_VIEW, which is available for mapping to the VirtualProvider. This tip only covers steps required in SAP NetWeaver BW to model the analytical view in a VirtualProvider.

1. Log in to the SAP NetWeaver BW 7.3 system and execute Transaction RSA1. Double-click on the INFOPROVIDER option on the left side of the screen.
2. Right-click on any of the InfoAreas on the right side of the screen, and choose the CREATE VIRTUALPROVIDER option.
3. In the next screen shown in Figure 1, you have four main options for creating the VirtualProvider.
   - BASED ON DATA TRANSFER PROCESS (for direct access)
   - BASED ON BAPI (for reporting on tables in external systems)
   - BASED ON FUNCTION MODULE (you can use a function module or class as the source for the VirtualProvider).
   - BASED ON A HANA MODEL

   In this example we are, of course, choosing BASED ON A HANA MODEL.



Edit InfoCube

InfoCube	ZTESTVP	
InfoArea	NODESNOTCONNECTED	Unassigned Nodes
ObjectType	InfoCube	
Template		

InfoProvider Type: Standard InfoCube [ ] Real Time; With Physical Data Store; [ ] In-Memory InfoCube; [✓ VirtualProvider]; Without Physical Data S; Based on Data Trans; [ ] With 3.x InfoSo; [ ] Unique Sour; Based on BAPI; With Source Syster; [ ] Execute Convers; Based on Function M; [✓ Based on a HANA Model]

System Type (Namespace): SAP (Delivered by SAP) [✓ CUS (Generated by User)]

[ ] Do Not Transform Selection Conditions

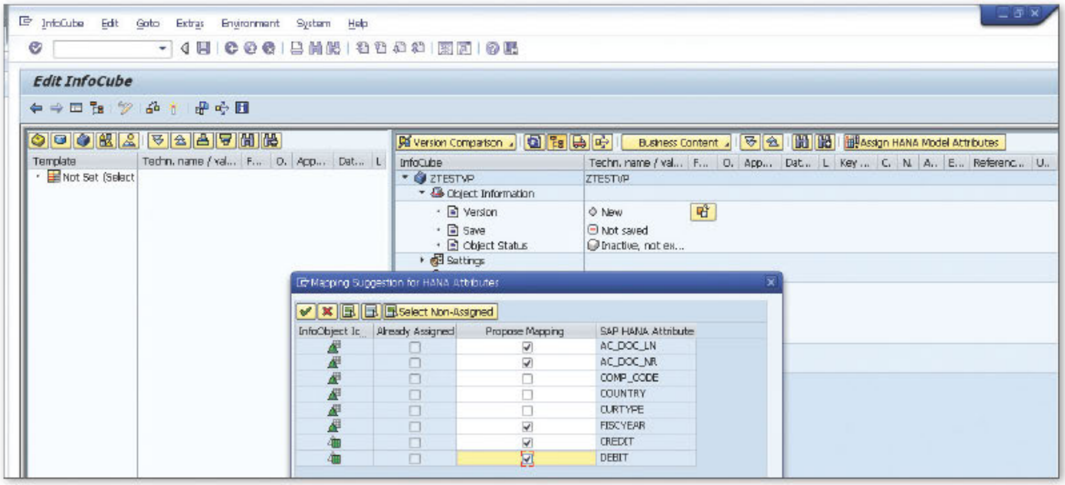
Virtual InfoProvider Based on SAP-HANA Model

Properties of Virtual InfoProviders

InfoCube	ZTESTVP
Descript.	
Package	PACKAGE_PH
HANA Information Model	ZANALYTIC_VIEW

Figure 1 Creating a VirtualProvider Based on the SAP HANA Analytical View

4. Enter the name "ZTESTVP" for the VirtualProvider, and choose the Based on a HANA Model option. Click on the Details button below the Based on a HANA Model radio button.
5. In the next screen that pops up, enter the package name and SAP HANA model name "ZANALYTIC_VIEW". Click on Continue.
6. In the next screen that appears, click on the Assign HANA Model Attributes button at the top-right side of the screen. In the screen that pops up, you can select the required fields from the SAP HANA analytical view that you plan to use in the VirtualProvider (see Figure 2).

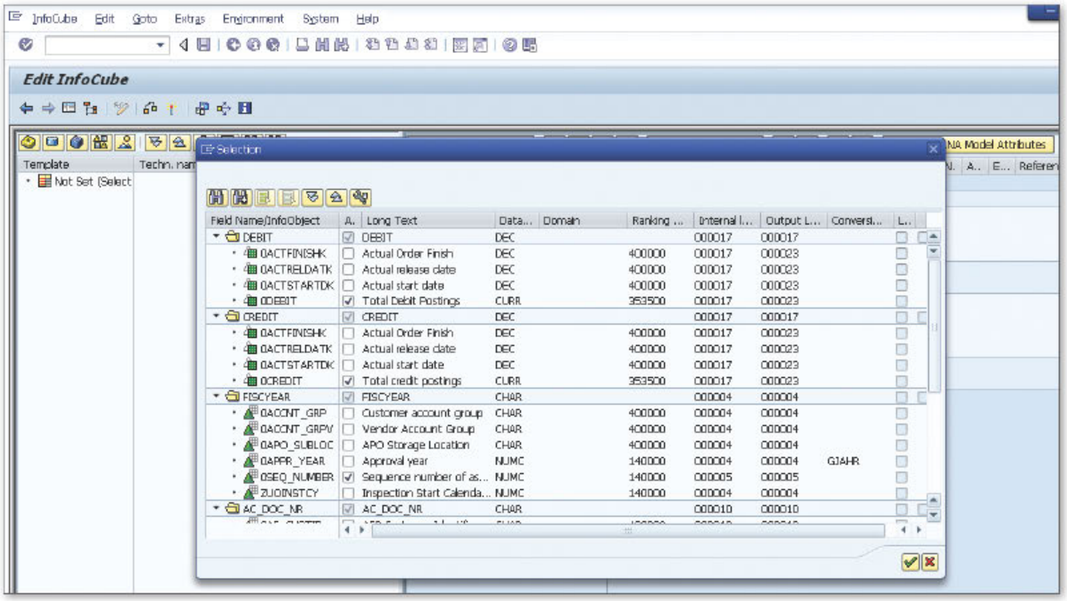


InfoObject Ic...	Already Assigned	Propose Mapping	SAP HANA Attribute
	☐	✓	AC_DOC_LN
	☐	✓	AC_DOC_NR
	☐	☐	COMP_CODE
	☐	☐	COUNTRY
	☐	☐	CURTYPE
	☐	✓	FISCYEAR
	☐	✓	CREDIT
	☐	✓	DEBIT

Figure 2 Selecting the SAP HANA Analytical View Fields for Mapping

For this, select the fields under Propose Mapping, and click on Continue.

7. In the next screen, shown in Figure 3, identify the SAP NetWeaver BW InfoObjects that you are going to use to map the SAP HANA analytical view fields.
8. Select the checkboxes for the fields, and click Continue.
9. All of the InfoObjects (both characteristics and key figuress) will appear in the dimensions and key figures of the InfoCube. Save and activate.



Field Name/InfoObject	A.	Long Text	Data...	Domain	Ranking ...	Internal l...	Output L...	Conversi...	L...
DEBIT	✓	DEBIT	DEC			000017	000017		☐
0ACTFINISHK	☐	Actual Order Finish	DEC		400000	000017	000023		☐
0ACTRELDATK	☐	Actual release date	DEC		400000	000017	000023		☐
0ACTSTARTDK	☐	Actual start date	DEC		400000	000017	000023		☐
0DEBIT	✓	Total Debit Postings	CURR		353500	000017	000023		☐
CREDIT	✓	CREDIT	DEC			000017	000017		☐
0ACTFINISHK	☐	Actual Order Finish	DEC		400000	000017	000023		☐
0ACTRELDATK	☐	Actual release date	DEC		400000	000017	000023		☐
0ACTSTARTDK	☐	Actual start date	DEC		400000	000017	000023		☐
0CREDIT	✓	Total credit postings	CURR		353500	000017	000023		☐
FISCYEAR	✓	FISCYEAR	CHAR			000004	000004		☐
0ACCNT_GRP	☐	Customer account group	CHAR		400000	000004	000004		☐
0ACCNT_GRPV	☐	Vendor Account Group	CHAR		400000	000004	000004		☐
0APO_SUBLOC	☐	APO Storage Location	CHAR		400000	000004	000004		☐
0APPR_YEAR	☐	Approval year	NUMC		140000	000004	000004	GJAHR	☐
0SEQ_NUMBER	✓	Sequence number of as...	NUMC		140000	000005	000005		☐
ZUOINSTCY	☐	Inspection Start Calenda...	NUMC		140000	000004	000004		☐
AC_DOC_NR	✓	AC_DOC_NR	CHAR			000010	000010		☐

Figure 3 Identifying the SAP NetWeaver BW InfoObjects to Map the SAP HANA Fields

You can now build BEx reports of this VirtualProvider using the BEx Query Designer. Alternatively, you can combine this VirtualProvider with other Info-Providers within SAP NetWeaver BW using a MultiProvider for reporting on the combined data.

Tip 95

Exposing BEx Queries to SAP BusinessObjects en Masse

You can expose several BEx queries to SAP BusinessObjects tools using table updates in SAP NetWeaver BW without logging into BEx Designer Studio.

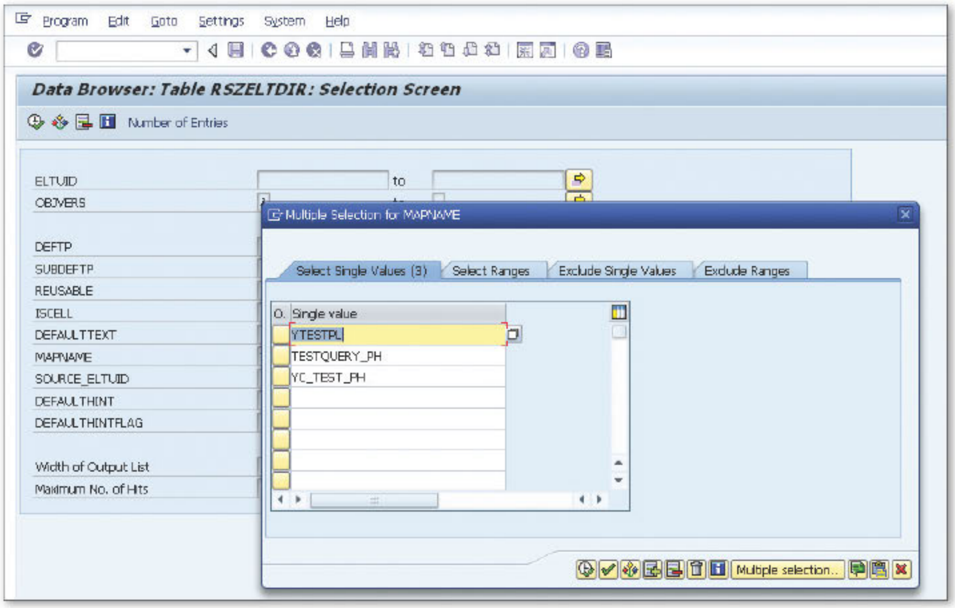
In a mature SAP NetWeaver BW environment, you'll have hundreds of BEx queries that are built over time. So when you move to an SAP BusinessObjects reporting platform, most likely you'll want to leverage these BEx queries that you have built over time and are quality tested. Typically, you'll expose these BEx queries for external access by checking the By OLE DB for OLAP option in the query properties. However, you can do this only one query at a time, and for a large number of queries, this would be a time-consuming process.

This tip describes how to make this setting for several queries at once on the SAP NetWeaver BW backend side by updating table entries directly.

✓ And Here's How ...

Follow these steps:

1. Log in to the SAP NetWeaver BW system and execute Transaction SE11. Enter table name "RSZELTDIR" and go to the table contents.
2. In the selection screen for the table contents, you can use the MAPNAME field to select specific queries. So if you have a specific list of queries that you need to expose externally, you can enter the query names in this field as shown in Figure 1. Table RSZELTDIR will give the technical names of the BEx queries, which is required for making the setting change.



Data Browser: Table RSZELTDIR: Selection Screen

ELTUID		to
OBJVERS		
DEFTP		
SUBDEFTP		
REUSABLE		
ISCELL		
DEFAULTTEXT		
MAPNAME		
SOURCE_ELTUID		
DEFAULTHINT		
DEFAULTHINTFLAG		
Width of Output List		
Maximum No. of Hits		

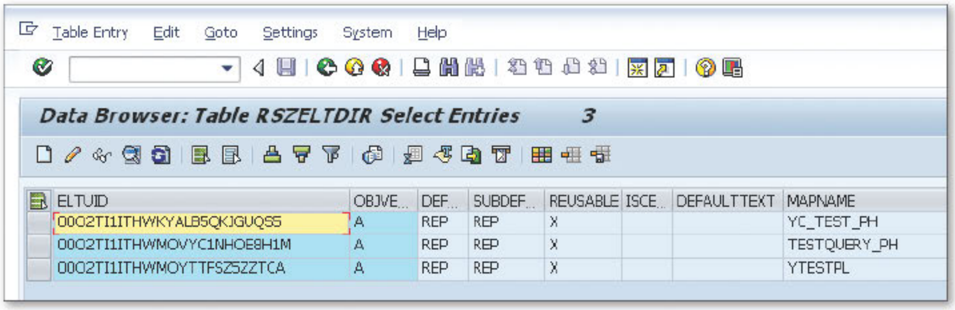
Multiple Selection for MAPNAME

O.	Single value
	YTESTPL
	TESTQUERY_PH
	YC_TEST_PH

Figure 1 Entering Query Names for Selection

If you need to update the External Access setting for all queries, then you can leave this selection wide open.

3. The screen shown in Figure 2 appears, showing the Query Technical names in the ELTUID column. Prepare the list of queries that need to be checked for By OLE DB for OLAP by copying the ELTUIDs from the table contents into notepad.

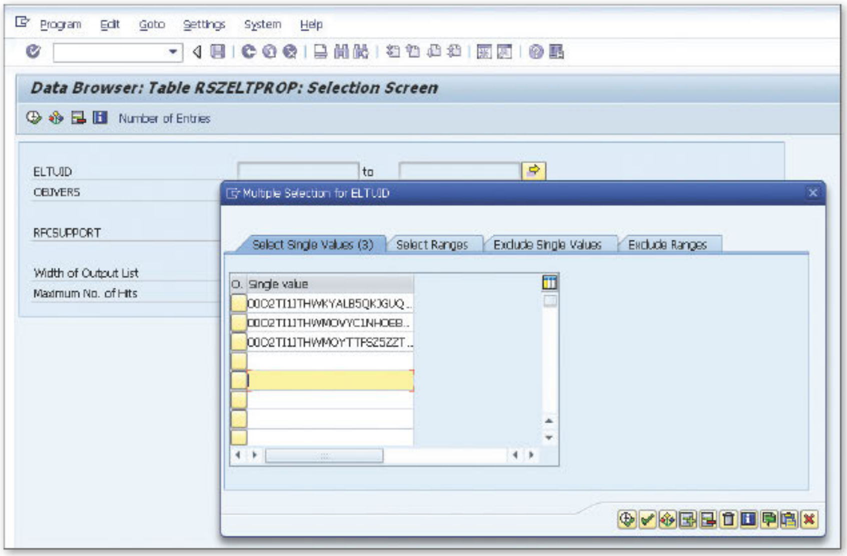


Data Browser: Table RSZELTDIR Select Entries 3

ELTUID	OBJVE..	DEF..	SUBDEF..	REUSABLE	ISCE..	DEFAULTTEXT	MAPNAME
0002TI1ITHWKYALB5QKJGUQSS	A	REP	REP	X			YC_TEST_PH
0002TI1ITHWMOVYC1NHOE8H1M	A	REP	REP	X			TESTQUERY_PH
0002TI1ITHWMOYTTFSZ5ZZTCA	A	REP	REP	X			YTESTPL

Figure 2 List of Query Technical Names

4. Go to Transaction SE11, and go to the contents of Table RSZELTPROP. Enter the values of the ELTUIDs copied in the previous step into the ELTUID field of the table contents of this table and execute (see Figure 3).



Data Browser: Table RSZELTPROP: Selection Screen

ELTUID to
OBJVERS
RFCSUPPORT
Width of Output List
Maximum No. of Hits

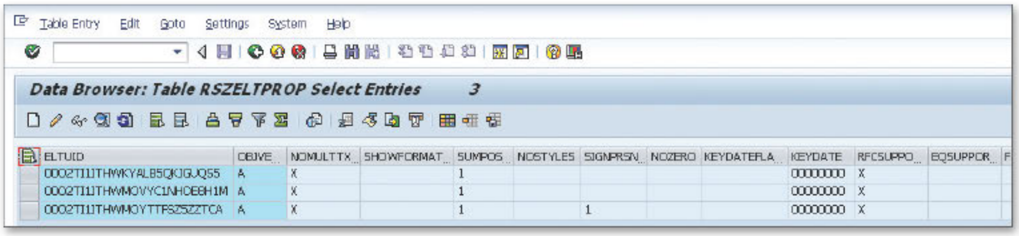
Multiple Selection for ELTUID

O.	Single value
	0002TI1JTHWKYALB5QKJGUQ...
	0002TI1JTHWMOVYC1NHOEB...
	0002TI1JTHWMOYTTFSZ5ZZT...

Figure 3 Entering the Query Technical Names for Selection

5. In the next screen, select all of the lines in the output, and click on the CHANGE button at the top of the screen.

6. Figure 4 now appears. Enter "X" in the RFCSUPPORT field for the rows in the list, and click on the SAVE button. You'll see that X is populated in the RFCSUPPORT field, which means the BEx query can be used for SAP BusinessObjects reporting tools.



Data Browser: Table RSZELTPROP Select Entries 3

ELTUID	OBJVE...	NOMULTTX...	SHOWFORMAT...	SUMPOS...	NOSTYLES	SIGNPRSN...	NOZERO	KEYDATEFLA...	KEYDATE	RFCSUPPO...	EQSUPPOR...	F
0002TI1JTHWKYALB5QKJGUQ55	A	X		1					0000000	X		
0002TI1JTHWMOVYC1NHOEBH1M	A	X		1					0000000	X		
0002TI1JTHWMOYTTFSZ5ZZTCA	A	X		1		1			0000000	X		

Figure 4 RFCSUPPORT with Value "X"

With this approach, you can expose BEx queries for external access, including the SAP BusinessObjects reporting tools.

Tip 96

Suppressing an End User's Unwanted and Excess Messages in Integrated Planning

You can implement a simple logic in an SAP NetWeaver BW class to suppress unwanted messages in Integrated Planning.

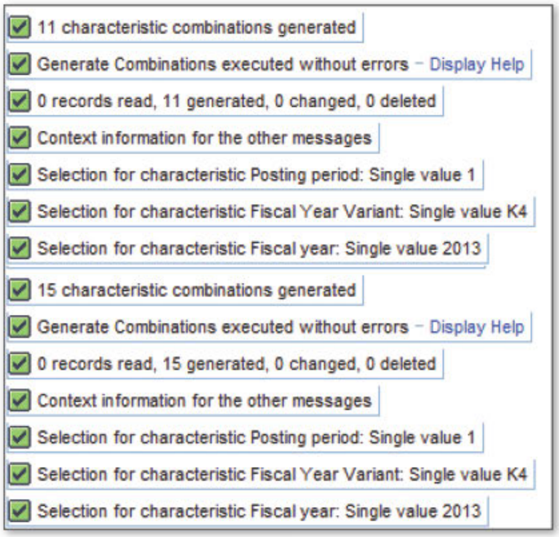
When you use Integrated Planning in SAP NetWeaver BW for implementing planning applications, you'll probably implement planning functions and sequences to execute specific planning tasks. When you execute these planning functions and sequences, typically the system will generate a lot of messages after the execution of the functions. Mostly these messages are very generic and not understandable to the end user. It's also possible that the users might miss out on really critical messages due to all these generic messages that appear on the screen.

This tip describes a method to suppress unwanted Integrated Planning messages in SAP NetWeaver BW after a planning function is executed.

✓ And Here's How ...

To describe this tip, we'll use an existing Integrated Planning data entry screen with planning functions already created. It's assumed that the reader understands basic SAP NetWeaver BW–Integrated Planning functionality so the basic steps of creating a plan data entry screen, creating planning functions, sequences, and so on, aren't described here.

We've included the screen shown in Figure 1 to show the type of messages that are generated after a planning function is executed successfully.

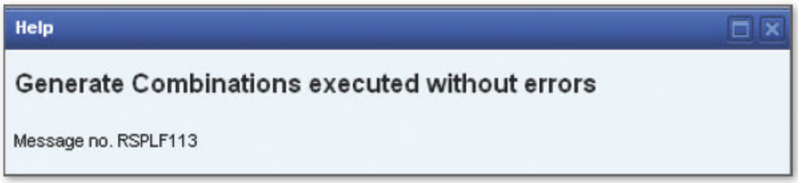


✓ 11 characteristic combinations generated
✓ Generate Combinations executed without errors – Display Help
✓ 0 records read, 11 generated, 0 changed, 0 deleted
✓ Context information for the other messages
✓ Selection for characteristic Posting period: Single value 1
✓ Selection for characteristic Fiscal Year Variant: Single value K4
✓ Selection for characteristic Fiscal year: Single value 2013
✓ 15 characteristic combinations generated
✓ Generate Combinations executed without errors – Display Help
✓ 0 records read, 15 generated, 0 changed, 0 deleted
✓ Context information for the other messages
✓ Selection for characteristic Posting period: Single value 1
✓ Selection for characteristic Fiscal Year Variant: Single value K4
✓ Selection for characteristic Fiscal year: Single value 2013

Figure 1 Planning Function Messages to Suppress

To suppress the unwanted messages in Integrated Planning, follow these steps:

1. In the messages screen for Integrated Planning, click on the Display Help link at the end of the message. A popup will appear as shown in Figure 2 displaying a message ID. (Make a note of these message IDs.)



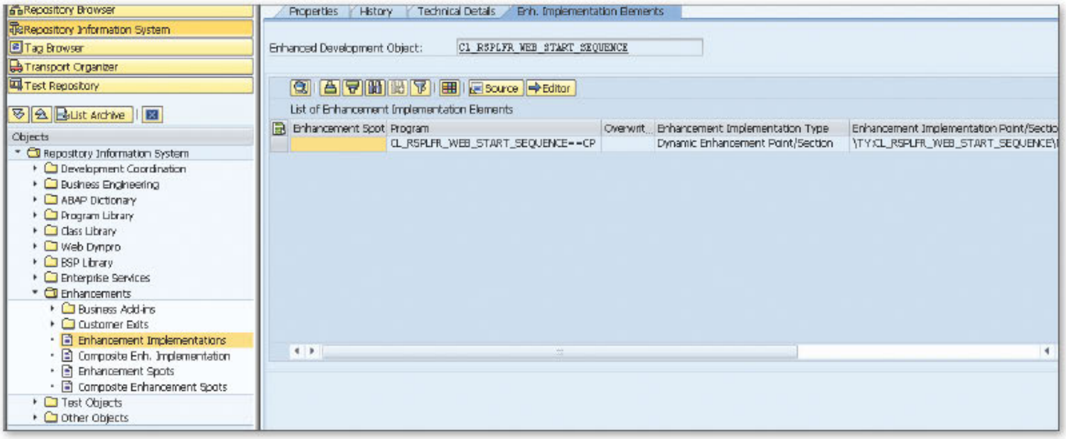
Help

Generate Combinations executed without errors

Message no. RSPLF113

Figure 2 Message Indicating Message IDs to Suppress

2. Log in to the SAP NetWeaver BW system. Go to Transaction SE80 and choose Repository Information System • Enhancements • Enhancement Implementations, and look for the object CL_RSPLFR_WEB_START_SEQUENCE (see Figure 3).



Enhanced Development Object: CL_RSPLFR_WEB_START_SEQUENCE

List of Enhancement Implementation Elements

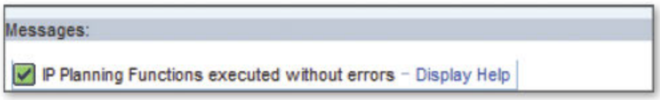
Enhancement Spot	Program	Overwrit...	Enhancement Implementation Type	Enhancement Implementation Point/Sectio
	CL_RSPLFR_WEB_START_SEQUENCE==CP		Dynamic Enhancement Point/Section	\TY:CL_RSPLFR_WEB_START_SEQUENCE\

Figure 3 Enhanced Development Object CL_RSPLFR_WEB_START_SEQUENCE

3. Double-click on the program, and enhance the code as shown here. In this code, you can include the message IDs from the Integrated Planning message screen so that these message IDs are filtered out. Save and activate.

```
CASE N_SEQUENCE_NAME.
    WHEN  'Planning Function Name' .
      DELETE e_t_mesg WHERE ARBGB = 'RSPLF'
                      AND ( TXTNR = '113' or
                            TXTNR = '019' or
                            TXTNR = '410').
  ENDCASE.
```

4. Now execute the planning function. Messages will be suppressed, and the result should appear as shown in Figure 4.



Messages:

✓ IP Planning Functions executed without errors – Display Help

Figure 4 Message Box Suppressing all Unwanted Warning Messages

Tip 97

Leveraging SAP NetWeaver BW Analysis Authorizations to Support Row-Level Security in SAP BusinessObjects Explorer

You can avoid duplicate maintenance of authorizations between SAP NetWeaver BW on SAP HANA and the SAP BusinessObjects platform by maintaining user groups in SAP NetWeaver BW to enable row-level security on SAP NetWeaver BW InfoProviders.

When business users are using SAP BusinessObjects Explorer on top of SAP NetWeaver BW on SAP HANA for search and exploration, this process can leverage SAP NetWeaver BW row-level security by importing the SAP NetWeaver BW analysis authorizations, which are converted as analytical privileges in SAP HANA. However, a major drawback in SAP HANA is that the authorizations are assigned in SAP BusinessObjects for every user individually. This will result in massive intensive security maintenance, and most importantly duplicate security maintenance in SAP NetWeaver BW and SAP BusinessObjects, resulting in SAP NetWeaver BW and SAP HANA becoming out of sync.

This tip describes a method to perform single maintenance of a user in SAP NetWeaver BW and their assignments to user groups in SAP BusinessObjects. The advantage of this method is that you can avoid dual maintenance by using SAP NetWeaver BW as the source system for your users and groups.

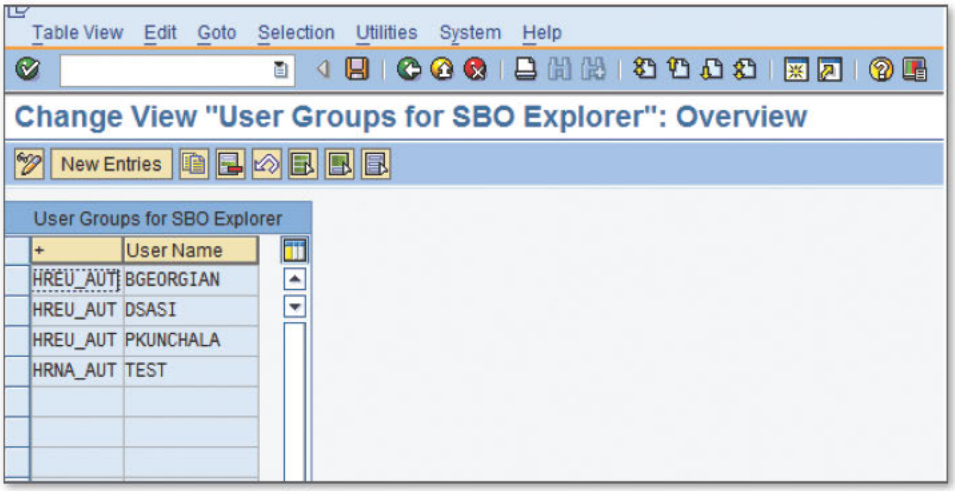
✓ And Here's How ...

In this tip, we'll use the SAP BusinessObjects Explorer authorization maintenance transaction in SAP NetWeaver BW to create an SAP BusinessObjects user group and assign the analysis authorization objects to the user group. These user groups can then be leveraged by BEx.

It's assumed that the reader is familiar with the analysis authorization concept in SAP NetWeaver BW and also understands the authorization user groups on the SAP BusinessObjects platform.

Follow these steps:

1. Log in to SAP NetWeaver BW and go to Transaction RSDDTPS.
2. Enter the SAP NetWeaver BW object (InfoProvider) for which you want to maintain the authorization. In this case, we'll use the existing InfoProvider 0TCT_MC11.
3. Choose the menu path Extras • Maintain Authorization Group.
4. You'll now see a screen (Figure 1) that allows you to edit user groups for SAP BusinessObjects Explorer (table view maintenance Transaction SM30 for Table RSDDTPS_AUTHGRP). Assign the users to the respective groups.



Change View "User Groups for SBO Explorer": Overview

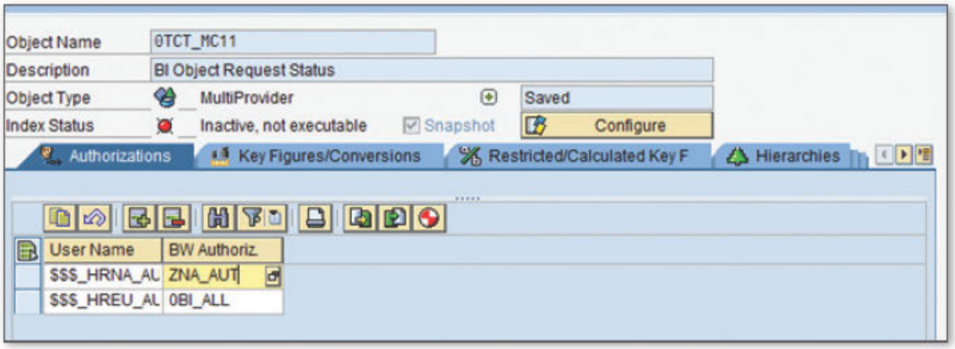
User Groups for SBO Explorer

+	User Name
HREU_AUT	BGEORGIAN
HREU_AUT	DSASI
HREU_AUT	PKUNCHALA
HRNA_AUT	TEST

Figure 1 Editing User Groups for SAP BusinessObjects Explorer

5. Close the user group maintenance screen, and you'll be taken back to the SAP BusinessObjects Explorer maintenance screen.

6. Go to the AUTHORIZATIONS tab as shown in Figure 2, and map the user group with the appropriate authorization object. Enter the SAP BusinessObjects Explorer user groups (created in step 1) using the pattern $$$_Group ID, for example, "$$$_HREU_AUT", and assign the authorization object to the group. All users in this group will automatically have the selected authorization.



Object Name	0TCT_MC11
Description	BI Object Request Status
Object Type	MultiProvider — Saved
Index Status	Inactive, not executable — Snapshot — Configure

User Name	BW Authoriz.
$$$_HRNA_AU	ZNA_AUT
$$$_HREU_AU	0BI_ALL

Figure 2 Mapping of Authorization Objects to SAP BusinessObjects Explorer User Groups

Table RSDDTPS_AUTH now has the information on the mapping between the SAP NetWeaver BW InfoProvider and the user authorization.

You can also use ABAP class `CL_RSDDTPS à SET_USERS_FOR_IPRO` for bulk uploads of user authorizations. Using this approach, you can create user groups for as many SAP NetWeaver BW objects as you like for SAP NetWeaver BW objects that have been activated for display in SAP BusinessObjects Explorer. You can then combine them with various authorization objects that are created on the SAP NetWeaver BW objects.

Tip 98

Connecting SAP NetWeaver BW to External Systems through DB Connect

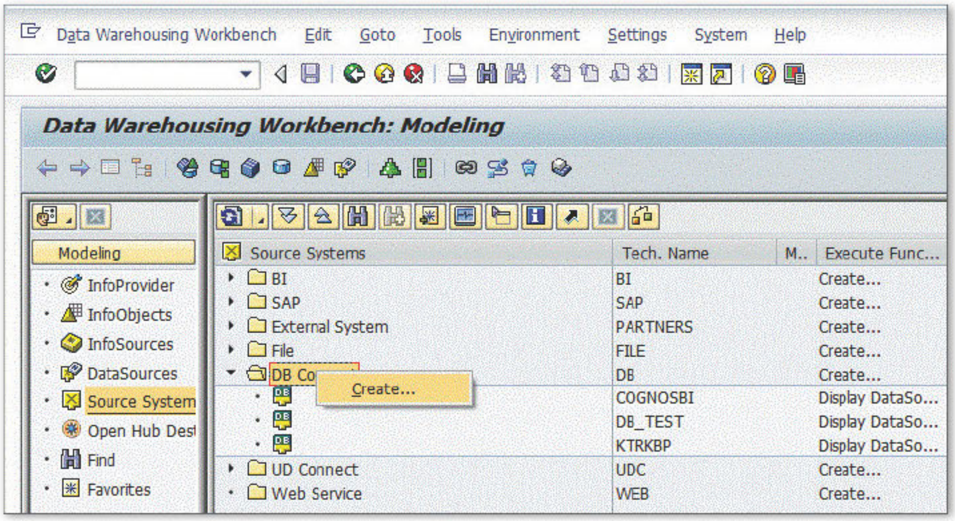
You can easily speed up the process of integrating non-SAP data into SAP NetWeaver BW with the use of the DB Connect protocol.

In many cases, businesses need to extract data from external databases such as SQL Server and Oracle and load the information directly into SAP NetWeaver BW. To address these requirements, you can use the SAP NetWeaver BW DB Connect source system connection. Without this protocol, integrating non-SAP data into SAP NetWeaver BW would be very time-consuming, using flat file extracts and data conversion processes. Through this tip, we'll demonstrate how to connect to SQL Server, but you should note that other connections are available and quite easy to use as well.

✓ And Here's How ...

To set up connectivity to non-SAP databases with the SAP NetWeaver BW DB Connect protocol, follow these steps:

1. Enter Transaction RSA1, and click on the SOURCE SYSTEM option (left pane). Then right-click on DB CONNECT, and choose CREATE (Figure 1) to add a new source system.

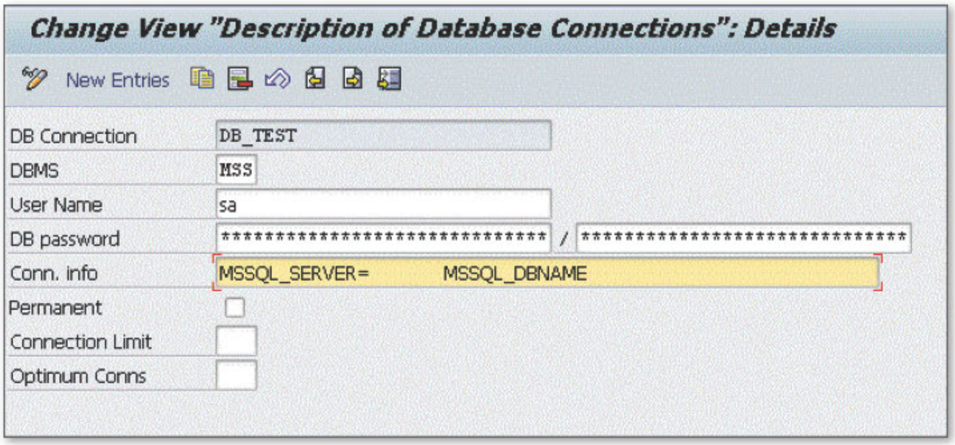


Source Systems	Tech. Name	M..	Execute Func...
BI	BI		Create...
SAP	SAP		Create...
External System	PARTNERS		Create...
File	FILE		Create...
DB Co	DB		Create...
	COGNOSBI		Display DataSo...
	DB_TEST		Display DataSo...
	KTRKBP		Display DataSo...
UD Connect	UDC		Create...
Web Service	WEB		Create...

Figure 1 Choosing Create from the Context Menu

2. On the next screen, define the Logical System Name, which is the ID of the new system, and the Source System Name, which is the description.
3. Choose the Database Management System (DBMS), which is the database platform that you are connecting to. In this tip, we choose ADA, which is the SAP database.
4. Enter your User Name and DB Password as shown in Figure 2 (note the password should be entered twice). Then enter the connection info string (in the Conn. info box). Follow this example and modify it to your needs: "MSSQL_SERVER=xxxxxx MSSQL_DBNAME=yyyyyy". In this example, note the following:
   - "xxxxxx" stands for the SQL Server ID.
   - "yyyyyy" stands for a specific database.
5. You can read more about the last three checkboxes in the dialog box by checking them and pressing F1; then you should consult the DBA of the attached database on whether and how to use them.

Now the new source system is ready for use in SAP NetWeaver BW like any other source system.

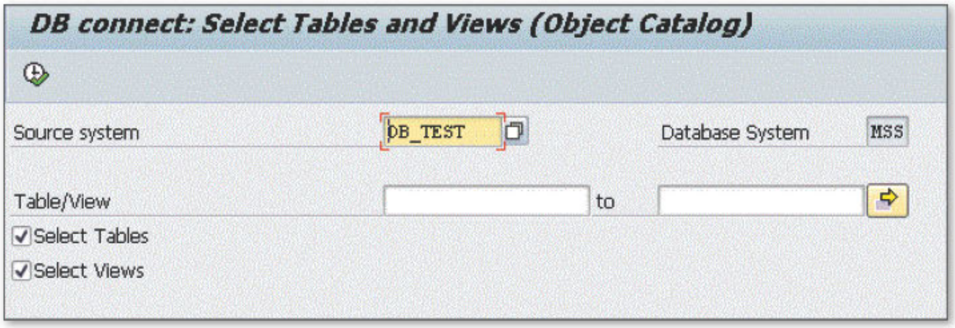

Change View "Description of Database Connections": Details

DB Connection	DB_TEST
DBMS	MSS
User Name	sa
DB password	******************************** / ********************************
Conn. info	MSSQL_SERVER= MSSQL_DBNAME
Permanent	☐
Connection Limit	
Optimum Conns	

Figure 2 Enter User Name, Database Password, and Connection Info

The following steps are optional if you'd like go deeper into the system to look at database tables and views:

1. View the database tables by right-clicking on the new source system and choosing Additional functions • Select database tables.
2. Search for tables and views by entering a search range as shown in Figure 3.


DB connect: Select Tables and Views (Object Catalog)

Source system: DB_TEST Database System: MSS

Table/View to

☑ Select Tables

☑ Select Views

Figure 3 Search for Tables and Views by Entering a Search Range

Tip 99

Reporting on SAP ERP Data from SAP NetWeaver BW with VirtualProviders

You can use VirtualProviders to report directly on data that exists in SAP ERP systems with SAP NetWeaver BW tools, without staging the data in SAP NetWeaver BW or needing custom ABAP development.

There are several scenarios in which you may require access to data in SAP ERP for reporting, in SAP NetWeaver BW, or may even need to combine both sets of data. In some cases, you may need to combine SAP NetWeaver BW data with SAP ERP data for reporting without staging all of the data in SAP NetWeaver BW. Normally, the only option available is to hire a small team of ABAP developers to develop these reports on the SAP ERP side.

This tip describes how to use a little-known functionality in SAP NetWeaver BW called *VirtualProviders*, which can be used to enable reporting on SAP ERP data on the SAP NetWeaver BW side. This makes operational reporting possible, without incurring a massive ABAP development overhead. This functionality also enables agile reporting in SAP NetWeaver BW wherein the SAP NetWeaver BW data can be combined with SAP ERP data in an agile manner and can provide a consistent reporting platform to the end users.

✓ And Here's How ...

VirtualProviders are cubes that don't store any data in SAP NetWeaver BW. When the user requests the data, a VirtualProvider queries the SAP ERP system for the data. There are three steps in using a VirtualProvider to enable SAP ERP reporting from SAP NetWeaver BW, which we'll explain in the following sections.

Step 1: Model the Data

One of the main concerns about VirtualProviders is performance due to the draw on SAP ERP system resources. You need to limit the number of VirtualProviders with efficient data modeling. It's critical for the design team to study all of the user reports and come up with a data model that brings in as much SAP ERP data without slowing down the SAP ERP system.

To illustrate, in Production Planning (PP), you would use one VirtualProvider for repetitive manufacturing and another for discrete manufacturing (consolidating VirtualProviders across module boundaries can get too complicated). These two VirtualProviders will cater 15 to 20 ABAP reports requested by the client. After these VirtualProviders are developed, the functional teams (with a little training) can easily develop the operational reports required along with the development team.

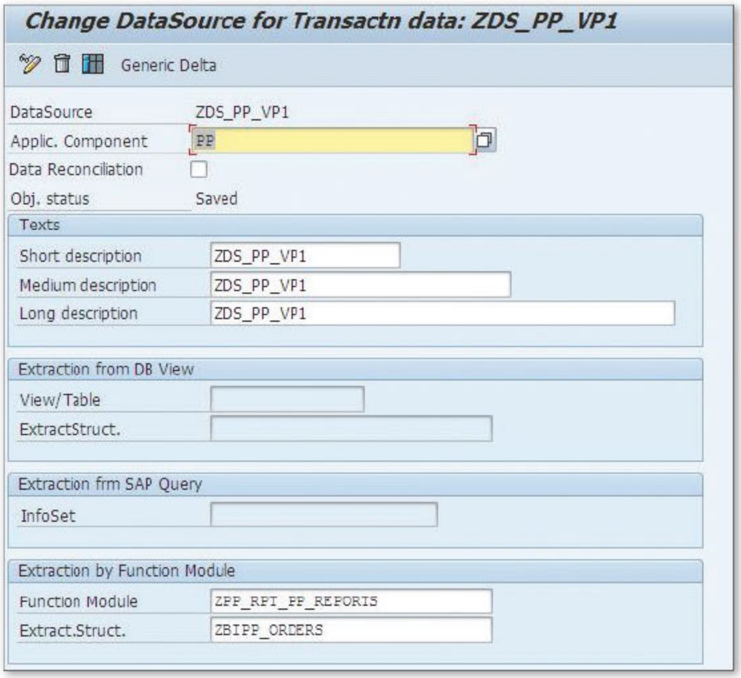
Step 2: Create the Source of Data

VirtualProviders in SAP NetWeaver BW can access SAP ERP data using table views in Transaction SE11 or ABAP function modules. For this scenario, it's assumed that the data modeling is completed, the source of data identified, and the function module for sourcing the data is already developed. Use standard function module `RSAX_BIW_GET_DATA_SIMPLE` and make a copy of it into the Z namespace. Add the import, export, and table parameters needed. The two main parameters for the function module relevant to this tip are:

- `I_T_SELECT` contains the selection conditions to be received from the report being run on the provider.
- `E_T_DATA` contains the data that the function module will send back to SAP NetWeaver BW after the query.

Now follow these steps:

1. Create a generic DataSource using the function module you just created as shown in Figure 1.



Change DataSource for Transactn data: ZDS_PP_VP1

Generic Delta

DataSource	ZDS_PP_VP1
Applic. Component	PP
Data Reconciliation	☐
Obj. status	Saved

Texts

Short description	ZDS_PP_VP1
Medium description	ZDS_PP_VP1
Long description	ZDS_PP_VP1

Extraction from DB View

View/Table	
ExtractStruct.	

Extraction frm SAP Query

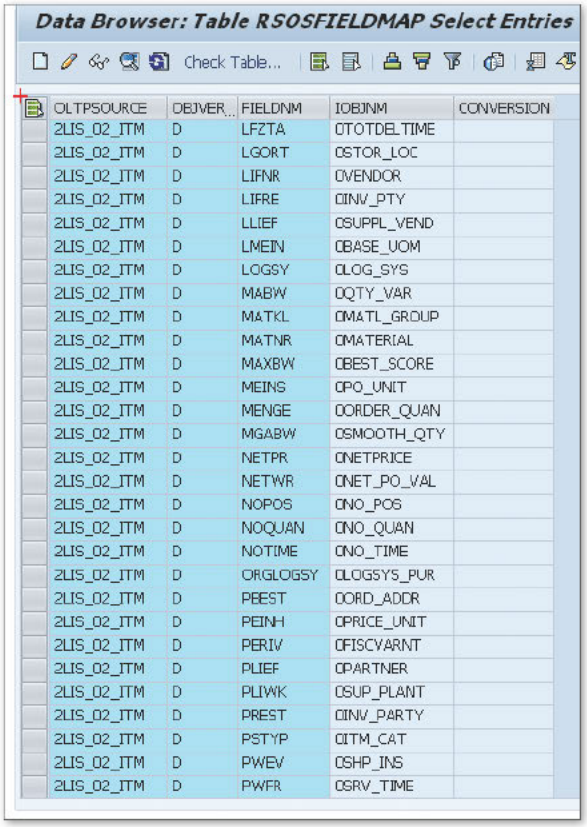
InfoSet	

Extraction by Function Module

Function Module	ZPP_RPT_PP_REPORTS
Extract.Struct.	ZBIPP_ORDERS

Figure 1 Structure of Function Module RSAX_BIW_GET_DATA_SIMPLE

2. In the source system, go to Transaction RSO2. Enter the DataSource and click the CREATE button.
3. In the screen, provide the function module name and extract structure. The extract structure (E_T_DATA in step 1) is an ABAP structure made through Transaction SE11. It needs to have all the fields you want to send to SAP NetWeaver BW via this DataSource. Make your own in the Z namespace.
4. Replicate the DataSource in SAP NetWeaver BW. Go to Transaction RSA1, click on DATASOURCE on the left side pane, click on the application component where we assign our DataSource, and replicate this into SAP NetWeaver BW (see Figure 2).


Data Browser: Table RSOSFIELDMAP Select Entries

Check Table...

OLTPSOURCE	OBJVER...	FIELDNM	IOBJNM	CONVERSION
2LIS_02_ITM	D	LFZTA	0TOTDELTIME	
2LIS_02_ITM	D	LGORT	0STOR_LOC	
2LIS_02_ITM	D	LIFNR	0VENDOR	
2LIS_02_ITM	D	LIFRE	0INV_PTY	
2LIS_02_ITM	D	LLIEF	0SUPPL_VEND	
2LIS_02_ITM	D	LMEIN	0BASE_UOM	
2LIS_02_ITM	D	LOGSY	0LOG_SYS	
2LIS_02_ITM	D	MABW	0QTY_VAR	
2LIS_02_ITM	D	MATKL	0MATL_GROUP	
2LIS_02_ITM	D	MATNR	0MATERIAL	
2LIS_02_ITM	D	MAXBW	0BEST_SCORE	
2LIS_02_ITM	D	MEINS	0PO_UNIT	
2LIS_02_ITM	D	MENGE	0ORDER_QUAN	
2LIS_02_ITM	D	MGABW	0SMOOTH_QTY	
2LIS_02_ITM	D	NETPR	0NETPRICE	
2LIS_02_ITM	D	NETWR	0NET_PO_VAL	
2LIS_02_ITM	D	NOPOS	0NO_POS	
2LIS_02_ITM	D	NOQUAN	0NO_QUAN	
2LIS_02_ITM	D	NOTIME	0NO_TIME	
2LIS_02_ITM	D	ORGLOGSY	0LOGSYS_PUR	
2LIS_02_ITM	D	PBEST	0ORD_ADDR	
2LIS_02_ITM	D	PEINH	0PRICE_UNIT	
2LIS_02_ITM	D	PERIV	0FISCVARNT	
2LIS_02_ITM	D	PLIEF	0PARTNER	
2LIS_02_ITM	D	PLIWK	0SUP_PLANT	
2LIS_02_ITM	D	PREST	0INV_PARTY	
2LIS_02_ITM	D	PSTYP	0ITM_CAT	
2LIS_02_ITM	D	PWEV	0SHP_INS	
2LIS_02_ITM	D	PWFR	0SRV_TIME	

« *Figure 2 DataSource Replication in SAP NetWeaver BW*

Step 3: Create Virtual Provider and Transformation

After you create the DataSource, use the following routine SAP NetWeaver BW steps to create an InfoCube and build the required transformations/Data Transfer Processes (DTPs):

1. **Create VirtualProvider**
   In Transaction RSA1, right-click on any InfoArea, and choose CREATE VIRTUALPROVIDER.
2. **Create a transformation between the DataSource and the virtual cube**
   Right-click on the cube and click TRANSFORMATION. The purpose of the transformation is to map DataSource fields with SAP NetWeaver BW InfoObjects.
3. **Create a Data Transfer Process**
   When the Transformation is complete, then create the DTP.

Tip 100

Analyzing the Performance of SAP BusinessObjects Reports using SAP NetWeaver BW Statistics

You can drastically improve system performance by analyzing SAP BusinessObjects reports using SAP NetWeaver BW statistics.

When you use SAP BusinessObjects BI tools for reporting on SAP NetWeaver BW data, it's common to see serious performance issues on the reports. However, it's very difficult to determine the root cause of the performance issues as there are different layers of connectivity and too many technologies working together.

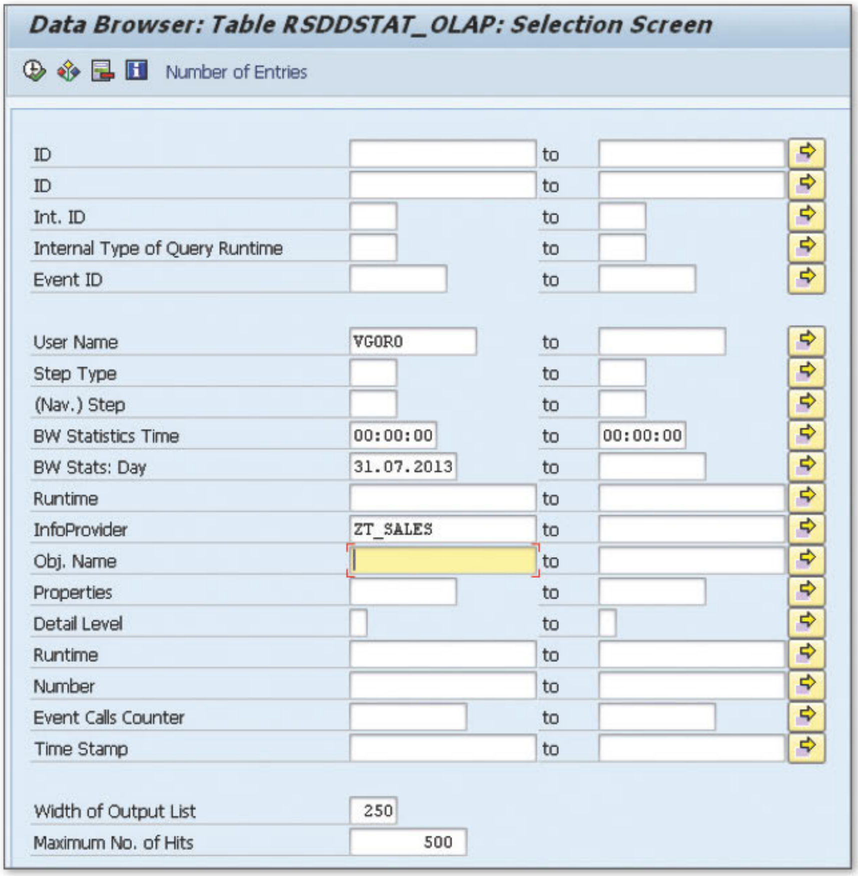
To circumvent this issue, this tip will explain how you can use the SAP NetWeaver BW statistics tables to pinpoint issues with SAP BusinessObjects query performance.

✓ And Here's How ...

To illustrate this tip, it's assumed that you have executed some reports using any of SAP's BusinessObjects BI tools. We'll demonstrate how the information in the SAP NetWeaver BW statistics tables can be used to identify the root cause of the performance issues. For the purpose of this tip, we'll use SAP NetWeaver BW statistics tables RSDDSTAT_OLAP and RSDDSTATEVENTS. RSDDSTAT_OLAP contains the data from the events from the areas for the front end and the calculation layer of the analytic engine.

1. Log in to the SAP NetWeaver BW system and execute Transaction SE11.

2. Enter table name "RSDDSTAT_OLAP" and execute to display the table contents. The screen shown in Figure 1 displays all the selection parameters available for filtering the data in the table.



ID		to	
ID		to	
Int. ID		to	
Internal Type of Query Runtime		to	
Event ID		to	
User Name	VGORO	to	
Step Type		to	
(Nav.) Step		to	
BW Statistics Time	00:00:00	to	00:00:00
BW Stats: Day	31.07.2013	to	
Runtime		to	
InfoProvider	ZT_SALES	to	
Obj. Name		to	
Properties		to	
Detail Level		to	
Runtime		to	
Number		to	
Event Calls Counter		to	
Time Stamp		to	
Width of Output List	250		
Maximum No. of Hits	500		

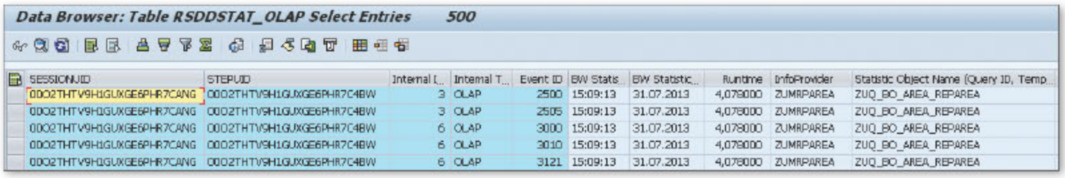
Figure 1 SE11 selection screen for RSDDSTAT_OLAP

You have several options available to filter and display the data. Some of the key data points in this table are:

- Internal Type of Query Runtime
  Typical values are "OLAP" for BEx query executions and "F4" for help while executing the query.
- Event ID
  This field contains the IDs for events that are associated with the query execution such as "Read Data" or "Generate Query".

- User Name
  Identify the user who executed the report
- Step Type
  This field would identify which tool was used to execute the report. The step type can be used to filter for statistics on specific SAP BusinessObjects tools, and you have the following options:
  - "BEX" means that the query was executed using the BEx Analyzer.
  - For all SAP BusinessObjects server tools such WebI, SAP BusinessObjects Dashboards, analysis OLAP, and SAP Crystal Reports, the value for this field would be "JAVA".
  - For the Analysis Office Excel tool, the value would be "AOE".
  - For Analysis Office PowerPoint, the value would be "AOP".
- InfoProvider
  You can use this field to display statistics information for specific InfoProviders in SAP NetWeaver BW.

3. In the selection screen in Figure 1, enter the appropriate selection values as required.
   - Enter the user name of the person who executed the SAP BusinessObjects reports.
   - If you want to analyze performance of WEbI, or SAP BusinessObjects Dashboards, or Analysis for OLAP or Crystal Reports, enter "JAVA" for Step Type.
   - Enter the current date in the BW Statistics: Day field.
4. Click on Execute. The next screen shown in Figure 2 will display all the entries for the selection criteria entered in Figure 1. The EVTIME field provides information on the run time of the events and the EVENTID provides the ID of the event that contributed to the run time. The value in the EVENTID field is a number and we will not be able to make any valuable interpretation based on that number. We therefore need to get the description of the value in EVENTID from Table RSDDSTATEVENTS.



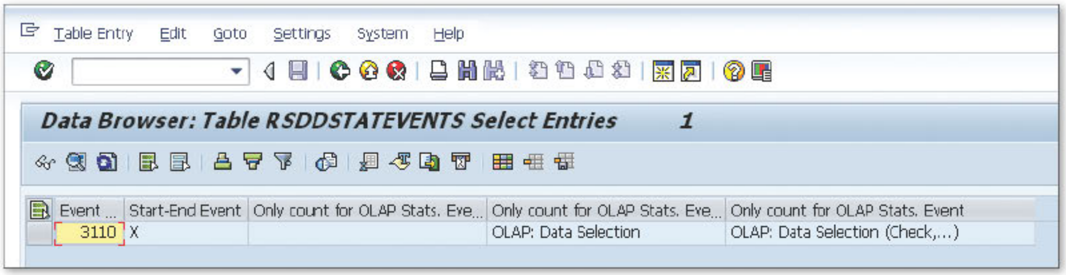
Data Browser: Table RSDDSTAT_OLAP Select Entries 500

SESSIONUID	STEPUID	Internal I...	Internal T...	Event ID	BW Statis...	BW Statistic...	Runtime	InfoProvider	Statistic Object Name (Query ID, Temp...
0DO2THTV9H1GUXGE6PHR7CANG	0002THTV9H1GUXGE6PHR7C4BW	3	OLAP	2500	15:09:13	31.07.2013	4,078000	ZUMRPAREA	ZUQ_BO_AREA_REPAREA
0DO2THTV9H1GUXGE6PHR7CANG	0002THTV9H1GUXGE6PHR7C4BW	3	OLAP	2505	15:09:13	31.07.2013	4,078000	ZUMRPAREA	ZUQ_BO_AREA_REPAREA
0DO2THTV9H1GUXGE6PHR7CANG	0002THTV9H1GUXGE6PHR7C4BW	6	OLAP	3000	15:09:13	31.07.2013	4,078000	ZUMRPAREA	ZUQ_BO_AREA_REPAREA
0DO2THTV9H1GUXGE6PHR7CANG	0002THTV9H1GUXGE6PHR7C4BW	6	OLAP	3010	15:09:13	31.07.2013	4,078000	ZUMRPAREA	ZUQ_BO_AREA_REPAREA
0DO2THTV9H1GUXGE6PHR7CANG	0002THTV9H1GUXGE6PHR7C4BW	6	OLAP	3121	15:09:13	31.07.2013	4,078000	ZUMRPAREA	ZUQ_BO_AREA_REPAREA

Figure 2 Table Contents of RSDDSTAT_OLAP Based on Selection Criteria Entered

5. Execute Transaction SE11. Enter table name RSDDSTATEVENTS and display table contents.

   In the selection screen enter the EVENTID "3110" that was displayed in Figure 2 and click on EXECUTE.



Data Browser: Table RSDDSTATEVENTS Select Entries 1

Event ...	Start-End Event	Only count for OLAP Stats. Eve...	Only count for OLAP Stats. Eve...	Only count for OLAP Stats. Event
3110	X		OLAP: Data Selection	OLAP: Data Selection (Check,...)

Figure 3 Contents of Table RSDDSTATEVENTS

The table contents in Figure 3 shows that the EVENTID = "3110" is for OLAP: Data Selection. This will help us conclude that issues came from the runtime for that report involving data selection.

Using this approach, we can analyze specific events related to SAP BusinessObjects reports execution and pinpoint the root cause of any performance issues that need troubleshooting.

The Authors

Buntic Georgian is a senior director at Archius (an SAP consulting company that is focused on business analytics) leading the SAP Business Analytics practice. He started working with SAP NetWeaver BW in 1998 with version 1.2B and has worked on all versions of SAP NetWeaver BW, including upgrades of the system to 3.0B, 7.0, and 7.3. His expertise includes developing BI competency centers with methodologies, best practices, how-to guides, and standards for development and deployment. He has been involved with several large scale BI implementations for SAP systems in leadership and architectural roles.

His current areas of expertise include technical architecture, building business cases for analytics, setting up BI COEs, and project management. Currently he is leading an SAP HANA implementation for one of the largest oil and gas companies in the world, and also working with several companies developing SAP HANA strategy, business case development for SAP HANA, and developing SAP HANA implementation plans.

Buntic has a graduate degree from C.T. Bauer College of Business (University of Houston), where he earned the Dean's award for academic excellence for the Master of Business Administration program.

Andrew Joo is the enterprise performance management (EPM) leader within the SAP Business Analytics Center of Excellence at IBM Global Business Services. He has more than 17 years of deep strategy, financial, cost, management, and technology consulting experience, encompassing leading industry firms, a multitude of technologies, industries, processes and methodologies, and roles with a strong track record of delivering and exceeding internal project and client expectations. In addition to serving as a thought leader and subject matter expert in the areas of enterprise performance management, business intelligence, and information management, he has been integral in project delivery, practice development, business development, industry, community outreach, and university lecturing efforts. He has an MBA in Strategy & Marketing from Rice University, a master's degree in Management Information Systems, and a bachelor's degree in Finance and Accounting.

Andrew has pioneered unique methodologies and go-to-market solutions leveraging an Integrated Enterprise Performance Management (EPM-BPC) and business intelligence paradigm that have been featured at SAP SAPPHIRE, SAP Financials, and SAP Reporting and Analytics conferences. Andrew currently serves as an Executive Advisor for both *BusinessObjects Experts* and *SAP BI Experts* for *SAPInsider*.

Index