



**How-to Guide  
SAP NetWeaver 2004s**

# **How to... Create Efficient MultiProvider Queries**

**Version History:  
3.00 – July 2006**

**Applicable Releases:  
SAP NetWeaver 2004s  
(Business Information Management - Enterprise  
Reporting, Query, and Analysis)**

© Copyright 2006 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C<sup>®</sup>, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data

contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

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

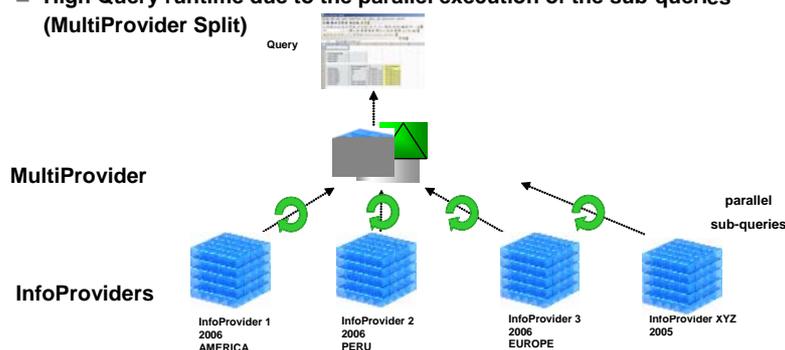
Any software coding and/or code lines /strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

# 1 Business Scenario

In a scenario where queries are executed on a MultiProvider, the system usually will perform a data selection on all InfoProviders that are part of the MultiProvider. Now if the MultiProvider is logically partitioned, for example by region as shown in the figure below, and the user selects data from one InfoProvider only, say "Europe", the system will run a sub-query on all three regional InfoProviders nevertheless. Therefore, more database accesses are required compared to a query built directly on the European InfoProvider.

## Query run time characteristics

- MultiProvider is used between a Query and the InfoProviders
- MultiProvider including many (>10) InfoProviders
- All InfoProviders of an application have the same structure (InfoObjects)
- Each InfoProvider contains homogeneous data slices divided by more than one characteristic (for example year, country or time zone etc.)
- Query runtime usually need data only from 1 or 2 InfoProviders
- High Query runtime due to the parallel execution of the sub-queries (MultiProvider Split)



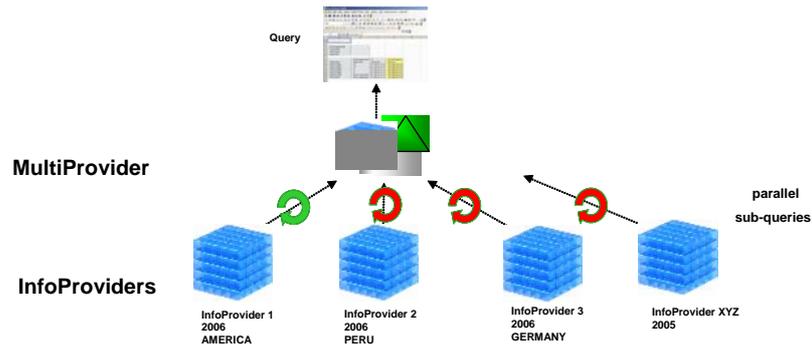
© SAP AG 2006

THE BEST-RUN BUSINESSES RUN SAP™ 

There are three solutions to gain efficiency:

### Restriction of the executed sub-queries

- Set a characteristic as a constant value of InfoProvider
- Include characteristic, InfoProvider (0INFOPROV) in query filter conditions
- Restrict the technical InfoObject 0INFOPROV automatically



© SAP AG 2006.

THE BEST-RUN BUSINESSES RUN SAP™ 

## 2 Introduction

### 2.1 Benefits

This document describes a solution that provides efficiency gains on one hand while on the other hand avoiding the disadvantages described in the first chapter. The solution will keep the required database selections on the MultiProvider to a minimum. One query will be sufficient for all selection scenarios whether the user selects a single region, several regions, or all regions. At the same time the system will determine dynamically, on which InfoProviders a sub-query needs to be run.

An additional benefit is that the logical partitioning can easily be changed. The solutions provide a flexible way of mapping what content is stored in which InfoProvider and also allows adding or removing InfoProviders to the MultiProvider without the need of maintaining queries.

### 2.2 Solution Overview

Apart from setting a characteristic to a fixed value in the InfoProvider, there are two solutions provided in this how-to guide:

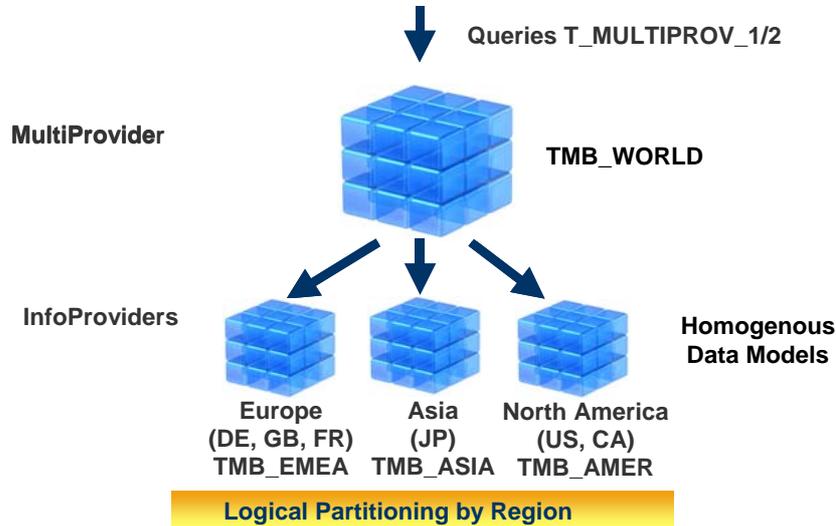
1. Restricting the InfoProvider automatically
2. Restricting the InfoProvider characteristic using a customer exit variable

The first solution can be used if the InfoProvider(s) can be automatically determined by checking the selected entries of the partition criteria set in the global filter of a query against the entries in the dimension table in the InfoProvider(s). Therefore, this solution only works on InfoCubes, but not on InfoSets or DataStore Objects. This solution should be used if the data in the InfoProviders is homogeneous. Please note that solution #1 is available as of a certain support package (see step-by-step instructions).

In the second solution, the virtual InfoProvider characteristic is used to develop efficient queries on a MultiProvider. However, instead of restricting the InfoProvider characteristic to a fixed InfoProvider, a customer exit variable is used. Based on the user input, the customer exit is used to dynamically determine, which InfoProviders actually need to be queried. If the user does not select any data from some of the InfoProviders, the system will ignore these InfoProviders, which leads to the desired efficiency gain. Solution #1 needs to be implemented if the data slices in the InfoProviders are not homogeneous and the InfoProviders can not be determined by the global restriction of the query.

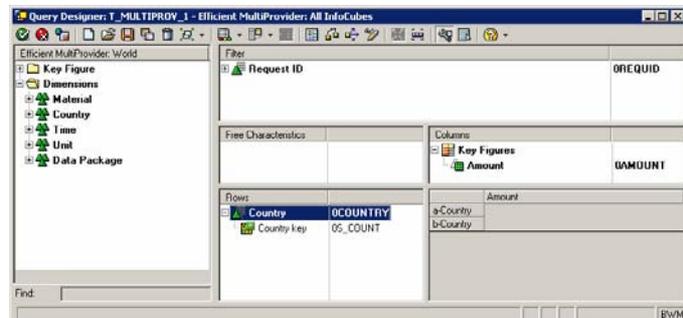
## 2.3 Example and Testing

In this paper we will use a MultiProvider that is logically partitioned by region based on the country characteristic.

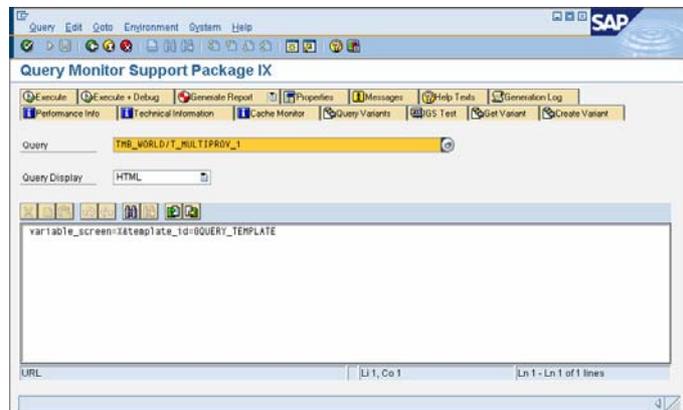


We use one query on the MultiProvider. At execution time, the user enters which country or countries should be selected. Therefore, we do not know upfront whether Europe, North America, or Asia data or a mix of them will be selected.

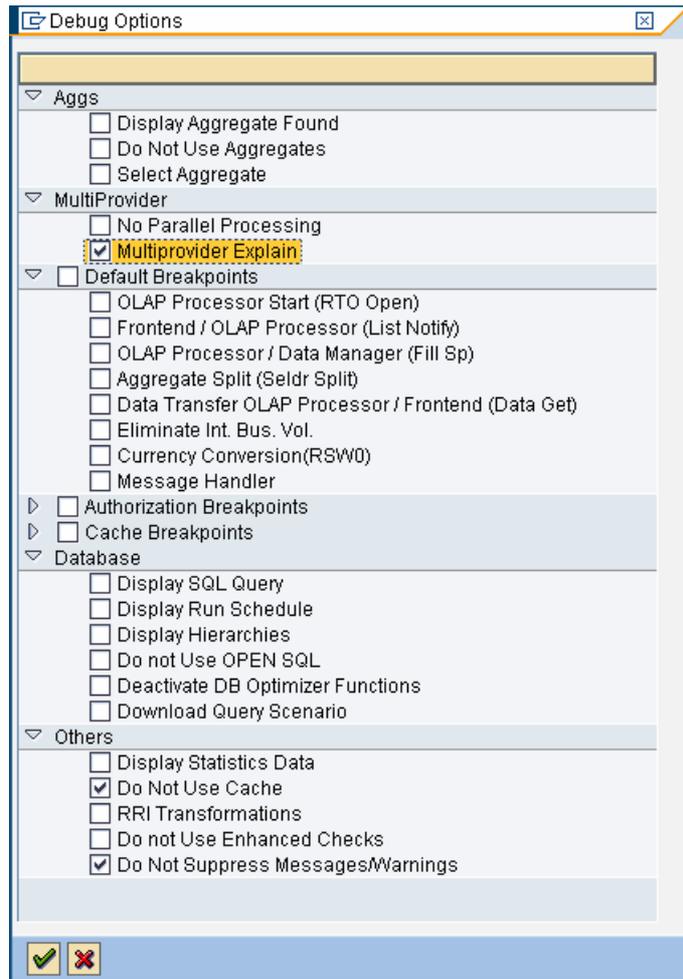
1. The figure on the right shows the query definition. Country is restricted to a user input variable.



2. Execute the query in transaction RSRT using "Execute + Debug" you can use the MultiProvider Explain to analyze which InfoProviders will be accessed.



- Select the “MultiProvider Explain” option.



- Enter various selections for country.

For example, select countries from North America and Asia.



- The messages will show that both InfoProviders are accessed no matter what countries are selected.

In the example, you can see that the Europe InfoProvider is read although no European countries were selected.



Efficient MultiProvider: All InfoCubes

Country	Amount
US United States	\$ 100.00
CA Canada	\$ 30.00
JP Japan	\$ 200.00
<b>Overall Result</b>	<b>\$ 330.00</b>

## 3 The Step by Step for Solution #1

The main facts about this solution are listed in the following slide:

Query Design

- **Restrict the technical InfoObject 0INFOPROV automatically**
  - Maintain customizing table “RRKMULTIPROVHINT” with partitioning criteria (InfoObject and Multiprovider)
  - During the query generation and query run time the not-needed PartProviders are automatically filtered
  - The InfoProvider filter is global and not per restricted key figure
    - Low maintenance effort (only InfoObject and MultiProvider)
    - QueryDesigner user does not need information about the underlying data model
    - Inhomogeneous InfoProviders are not supported (filter is global and not for each calculated key figure)
    - Available as of BW 3.0 SP30 (For BW 3.0 SP28 OSS Note 913975 is a prerequisite for note 911939). Also, Note 950258 needs to be applied.
    - DataStore Objects and InfoSets are not supported
    - No Line-Item Dimension is supported for partitioned InfoObject Criteria

© SAP AG 2006, Title of Presentation / Speaker Name / 1

THE BEST-RUN BUSINESSES RUN SAP™ 

### Technical Details:

The RRKMULTIPROVHINT can be used instead of or in conjunction with using the constants on the individual InfoCubes. The constants will direct the run schedule to the correct InfoCube, but if you want, you can maintain things centrally in this table if the constants aren't set.

In the RRKMULTIPROVHINT table, you can specify the MultiProvider and a characteristic for partitioning and if a query has restrictions on this characteristic, the OLAP processor will check which InfoCubes can return data for the query. The data manager can then completely ignore the remaining InfoCubes. This is handled by the OLAP processor.

The system is doing a quick read of the dimension table of the InfoCube to check if the selected value (or multiple values!) are included and it's "worth" reading the full InfoCube.

### Pre-requisites:

- MultiProvider must contain only InfoCubes.
- For SAP NetWeaver 2004s, ensure you are at SPS 9 or above or apply SAP note 954889.

1. Maintain table "RRKMULTIPROVHINT" with the partitioning criteria Country (InfoObject and MultiProvider name).

See Appendix for details.

The screenshot shows the SAP Data Browser interface for table RPKMULTIPROVHINT. The window title is "Data Browser: Table RPKMULTIPROVHINT Select Entries 1". The table structure is as follows:

MULTIPROV	CHANN	POSIT
TMB_WORLD	08_COUNT	0001

Additional information from the screenshot: Table: RPKMULTIPROVHINT, Displayed fields: 3 of 3, Fixed columns: 2, List width 0250.

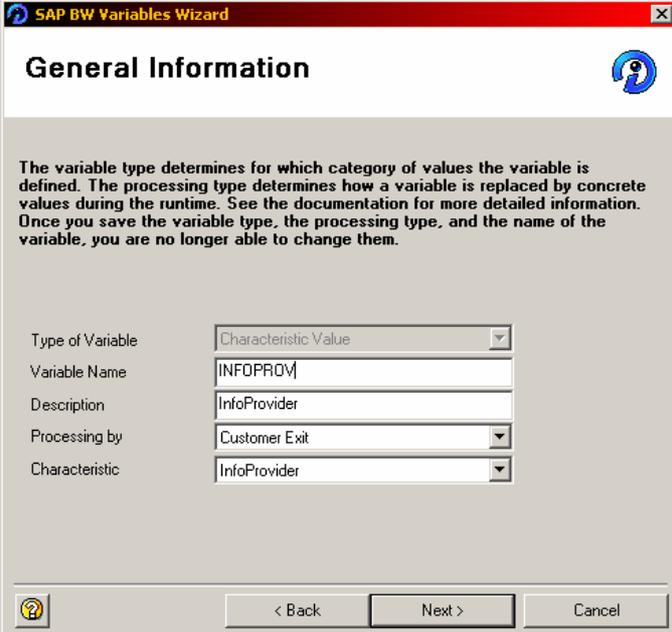
2. Execute the Query

Please check note 954889.

## 4 The Step by Step for Solution #2

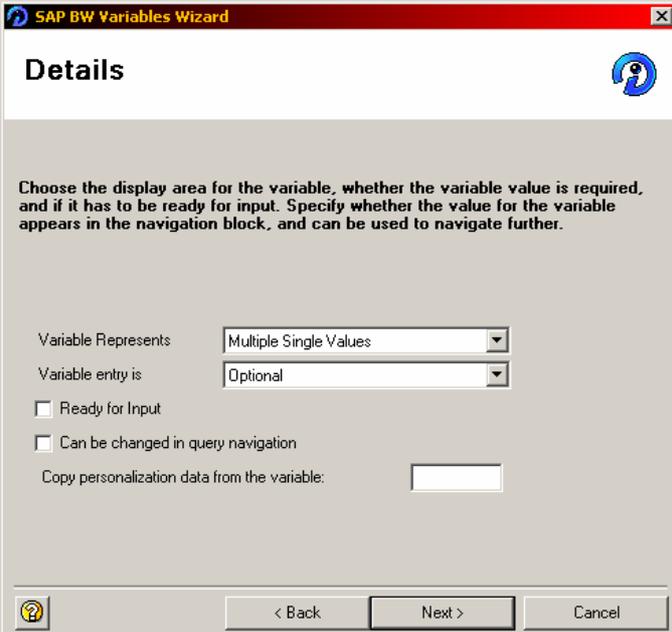
1. Create a new query (or start with a copy of the original query). Right click on the “InfoProvider” characteristic and create a new variable.

Select “Customer Exit” for the processing type.



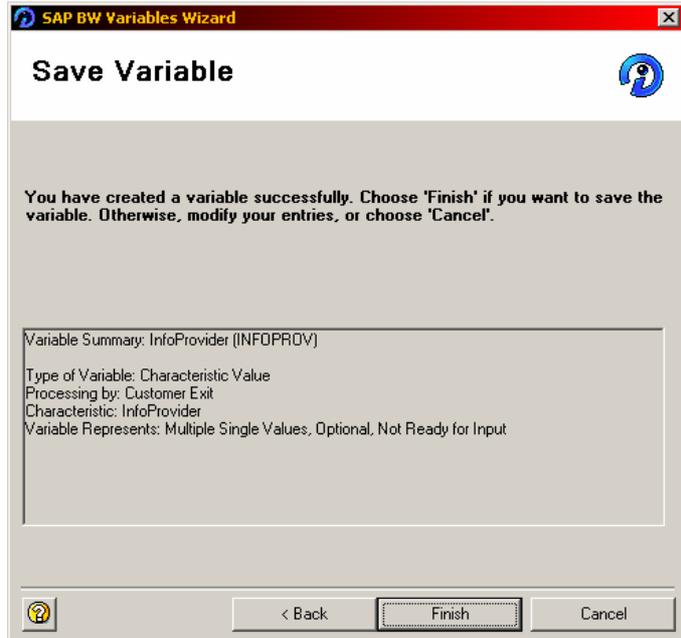
The screenshot shows the 'SAP BW Variables Wizard' dialog box, 'General Information' tab. The title bar reads 'SAP BW Variables Wizard'. The main heading is 'General Information'. Below the heading is a paragraph: 'The variable type determines for which category of values the variable is defined. The processing type determines how a variable is replaced by concrete values during the runtime. See the documentation for more detailed information. Once you save the variable type, the processing type, and the name of the variable, you are no longer able to change them.' Below this text are five fields: 'Type of Variable' (dropdown menu with 'Characteristic Value' selected), 'Variable Name' (text box with 'INFOPROV'), 'Description' (text box with 'InfoProvider'), 'Processing by' (dropdown menu with 'Customer Exit' selected), and 'Characteristic' (dropdown menu with 'InfoProvider' selected). At the bottom are three buttons: '< Back', 'Next >', and 'Cancel'.

2. In the detailed settings, select “Multiple Single Values” and “Optional”. The variable should not be ready for input or changeable in the query navigation.



The screenshot shows the 'SAP BW Variables Wizard' dialog box, 'Details' tab. The title bar reads 'SAP BW Variables Wizard'. The main heading is 'Details'. Below the heading is a paragraph: 'Choose the display area for the variable, whether the variable value is required, and if it has to be ready for input. Specify whether the value for the variable appears in the navigation block, and can be used to navigate further.' Below this text are four fields: 'Variable Represents' (dropdown menu with 'Multiple Single Values' selected), 'Variable entry is' (dropdown menu with 'Optional' selected), a checkbox for 'Ready for Input' (unchecked), and a checkbox for 'Can be changed in query navigation' (unchecked). Below these is a text box for 'Copy personalization data from the variable:'. At the bottom are three buttons: '< Back', 'Next >', and 'Cancel'.

3. Finish and save the variable.



4. Finally, add the variable to the filter of the query.



- Implement the Customer Exit for the InfoProvider Variable. Enter the coding (see also Appendix). Save and activate the coding.

- Execute the new query using RSRT as described above.

For example, if only data about Germany is selected, only one InfoProvider will be accessed and the other ones will be excluded.

The same query can be used to select US data or a mix of other countries.

**Variables for Ad Hoc Report**

Country key: DE Germany Include Insert Row

Execute Check

**Messages:**

- \*\*\* MultiProvider TMB\_WORLD: InfoProvider TMB\_AMER \*\*\*
- OSS Note 489135 describes messages in more detail
- Access to TMB\_AMER was excluded
- \*\*\* MultiProvider TMB\_WORLD: InfoProvider TMB\_ASIA \*\*\*
- Access to TMB\_ASIA was excluded
- \*\*\* MultiProvider TMB\_WORLD: InfoProvider TMB\_EMEA \*\*\*
- OINFOPROV in MultiProvider TMB\_WORLD was modeled in OINFOPROV in InfoPr
- OREQUID in MultiProvider TMB\_WORLD was modeled in OREQUID in InfoProvid
- OCOUNTRY in MultiProvider TMB\_WORLD was modeled in OCOUNTRY in InfoProv
- OCURRENCY in MultiProvider TMB\_WORLD was modeled in OCURRENCY in InfoPr
- The MultiProvider query is executed sequentially (reason: ONE)

**Efficient MultiProvider: With Variable**

Country Key Figures

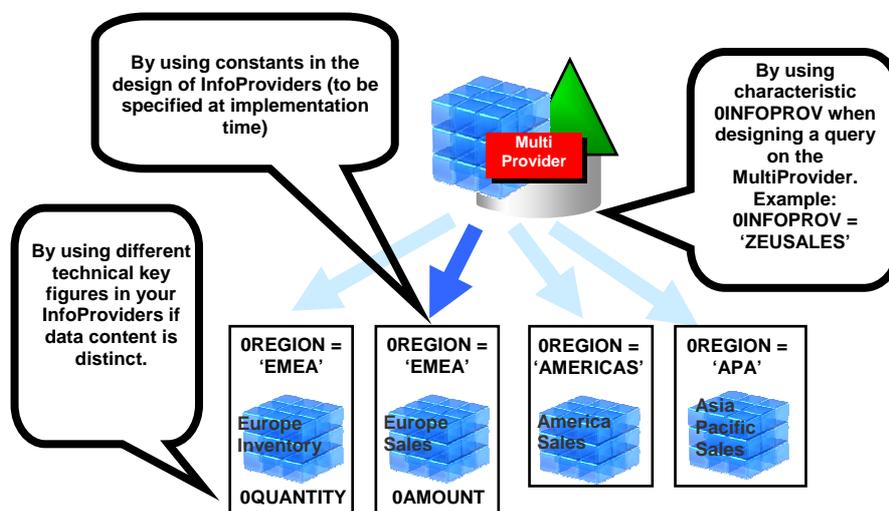
Country	Amount
DE Germany	\$ 15.00
<b>Overall Result</b>	<b>\$ 15.00</b>

## 5 Comments for Solution #2

- The solution can be implemented for SAP BW release 3.0 or greater.
- The content mapping of countries i.e. regions to InfoProviders is based on a custom table. The table and ABAP coding is built to handle single values as well as ranges.
- In general, the mapping also allows for great flexibility in managing the content. For example, data can be moved between InfoProviders or archived, new InfoProviders can be added over time or removed if necessary. All you need to do is maintain the mapping.
- Although the logical partitioning in the example is based on one characteristic (country), the solution will work also if the InfoProviders are partitioned using several characteristics.
- The example uses homogenous InfoCubes. However, the solution will work the same way when heterogeneous data models and InfoProviders (InfoCubes, DataStore Objects, InfoSets, etc.) are combined into a MultiProvider.
- If you need help in implementing the customer exit, please check the online help or see the “How to... Verify Variable Input” document for SAP BW release 2.x.

Note: For completeness, it should be mentioned that there are other ways of making efficient usage of MultiProviders. The three options are shown in the following picture. However, none of them is as flexible as the solution described in this how-to paper.

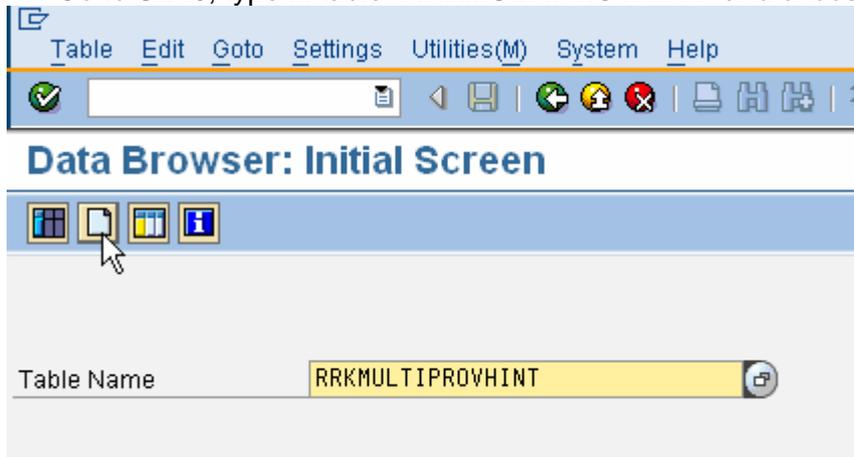
Other options to make sure that a MultiProvider only retrieves data out of relevant InfoProviders at runtime:



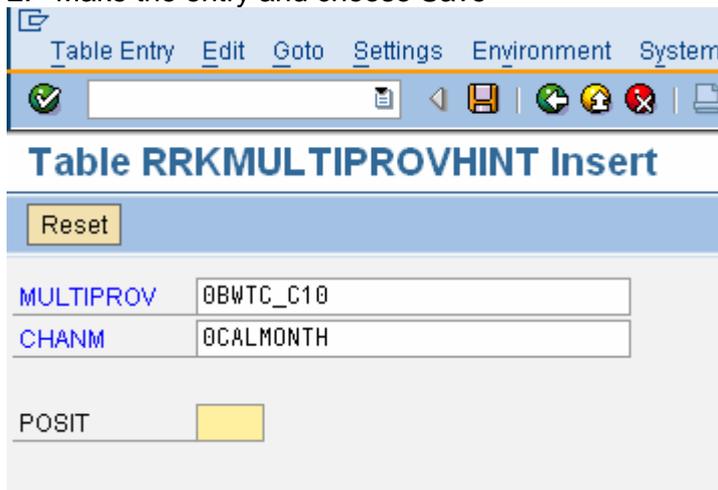
## 6 Appendix for Solution #1

### 6.1 Maintaining a table

1. Go to SE16, type in table "RRKMULTIPROVHINT" and choose New.



2. Make the entry and choose Save



## 7 Appendix for Solution #2

### 7.1 Custom Mapping Table

Transp. Table: YBW\_MAPPING Active  
 Short Text: BW: Efficient MultiProviders

Attributes | Delivery and Maintenance | **Fields** | Entry help/check | Currency/Quantity Fields

1 / 4

Field	Key	Initi...	Data element	Data T...	Length	Deci...	Short Text
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Client
LOW	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	/B10/OICOUNTRY	CHAR	3	0	Country key
HIGH	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	/B10/OICOUNTRY	CHAR	3	0	Country key
INFOPROV	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	RSINFOPROV	CHAR	30	0	InfoProvider

### 7.2 Sample Data

MANDT	LOW	HIGH	INFOPROV
800	CA		TMB_AMER
800	DE		TMB_EMEA
800	FR		TMB_EMEA
800	GB		TMB_EMEA
800	JP		TMB_ASIA
800	US		TMB_AMER

### 7.3 Customer Exit

```

*-----*
*  INCLUDE ZXRSRU01                                *
*-----*

*** Enhancement: MultiProvider using InfoProvider Variable
include YBW_INFOPROVIDER_VARIABLE.

*&-----*
*&  Include          YBW_INFOPROVIDER_VARIABLE    *
*&-----*

DATA:
  ls_var      TYPE rrs0_s_var_range,
  ls_range    TYPE rsr_s_rangesid,
  l_contained TYPE c,
  ls_mapping  TYPE ybw_mapping,
  lt_mapping  TYPE TABLE OF ybw_mapping.

* Called after variable popup
IF i_step = 2 AND i_vnam = 'INFOPROV'.

```

```

* Read mapping table
REFRESH lt_mapping.
SELECT * FROM ybw_mapping INTO TABLE lt_mapping.

* Process all selection for country variable
REFRESH e_t_range.
LOOP AT i_t_var_range INTO ls_var WHERE vnam = 'S_COUNT'.

* Process all mapping rules
LOOP AT lt_mapping INTO ls_mapping.

* Always fill LOW and HIGH, Otherwise logic below will not work
IF ls_mapping-high IS INITIAL.
  ls_mapping-high = ls_mapping-low.
ENDIF.

* Check if selection is contained in the defined InfoProvider
CLEAR l_contained.
CASE ls_var-opt.
  WHEN 'EQ'.
    IF ls_var-low BETWEEN ls_mapping-low AND ls_mapping-high.
      l_contained = 'X'.
    ENDIF.
  WHEN 'BT'.
    IF ls_var-low <= ls_mapping-high AND
      ls_var-high => ls_mapping-low.
      l_contained = 'X'.
    ENDIF.
ENDCASE.

* Add InfoProvider to return table
* Note: Use COLLECT to avoid duplicates
IF l_contained = 'X'.
  CLEAR ls_range.
  ls_range-sign = 'I'.
  ls_range-opt = 'EQ'.
  ls_range-low = ls_mapping-infoprov.
  COLLECT ls_range INTO e_t_range.
ENDIF.

  ENDLLOOP. " lt_mapping

ENDLOOP. " i_t_var_range

EXIT.

ENDIF.

```

<http://www.sdn.sap.com/irj/sdn/howtoguides>