MultiProviders

Technical Details and the Slow Moving Item Scenario as an Application Example

ADDON.NDBWAW

Release 3.0B

Copyright

Icons

Icon	Meaning
⚠	Caution
💬	Example
💡	Note
🧭	Recommendation
SYN	Syntax

Typographic Conventions

Type Style	Description
Example text	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths and options. Cross-references to other documentation.
Example text	Emphasized words or phrases in body text, titles of graphics and tables.
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example, SELECT and INCLUDE.
`Example text`	Screen output. This includes file and directory names and their paths, messages, source code, names of variables and parameters as well as names of installation, upgrade and database tools.
`EXAMPLE TEXT`	Keys on the keyboard, for example, function keys (such as `F2`) or the `ENTER` key.
`Example text`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.

MultiProviders

Definition

A MultiProvider is a type of InfoProvider that combines data from a number of InfoProviders and makes them available as a whole to reporting. The MultiProvider does not itself contain any data. Its data comes entirely from the InfoProviders on which it is based. These are combined using a union operation.

InfoProviders and MultiProviders are the objects or views that are relevant for reporting.

Use

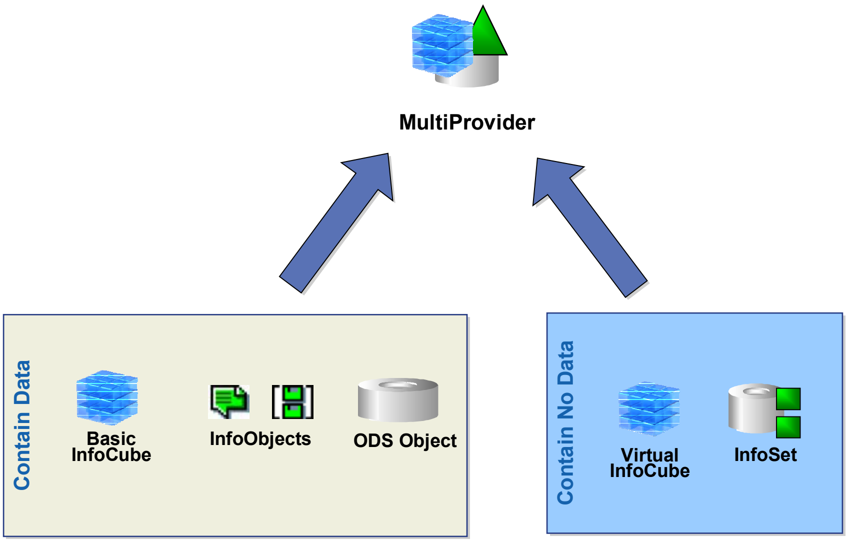
A MultiProvider allows reporting using several InfoProviders.

InfoCube and InfoCube: You have an InfoProvider with actual data for a logically closed business area and an equivalent InfoProvider with plan data. You can combine the two InfoProviders into one MultiProvider so that you can compare the actual data with the planned data of a query. In BW release 2.0B/2.1C, this combination of two InfoCubes was still called a MultiCube.

InfoCube and InfoObject: You have an InfoCube with your products and sales. Combine the InfoObject 0MATERIAL with it. In this way you can display the "slow-moving items" since products that do not result in sales are also displayed. For a detailed description of the procedure, refer to the Slow Moving Items Scenario [Page 9].

Structure

A MultiProvider can consist of different combinations of the following InfoProviders: InfoCube, ODS object, InfoObject and InfoSet.



The data from these objects is combined into a MultiProvider via a union operation. Here, the system constructs the union set of the involved data sets. In other words, all values of these data sets are combined. As a comparison: InfoSets are created using joins. These only combine values that appear in both tables. In contrast to a union, joins build the intersection of the featured tables

As a comparison, see also InfoSet [Ext.].

In a MultiProvider, every characteristic in every InfoProvider involved should correspond to one characteristic or navigation attribute (as long as these are available). If it is not otherwise clear, you have to define the InfoObject which the characteristic of the MultiProvider should be assigned to.

> The MultiProvider contains the characteristic 0COUNTRY and an InfoProvider contains the characteristic 0COUNTRY as well as the navigation attribute 0CUSTOMER__0COUNTRY. In this case select exactly one of these InfoObjects in the assignment table.

Select a key figure contained in a MultiProvider from at least one of the InfoProviders involved. In general, exactly one of the InfoProviders provides the key figure. However, there are cases where it is better for the user to select from more than one InfoProvider:

> If the key figure is 0SALES and it is stored redundantly in more than one InfoProvider (that is, contained completely in all value combinations of the characteristics), you should select from exactly one of the InfoProviders involved (otherwise the value would be incorrectly added in the MultiProvider for each occurrence).

However, if 0SALES is in a InfoProvider, for example as an actual value saved in another plan value, so that there is no overlapping in the data records (in other words: Sales is divided disjunctively between several InfoProviders), then it makes sense to make a selection from several InfoProviders.

You can divide MultiProviders into the following categories:

1. homogenous MultiProviders:

   These consist of technically identical InfoProviders, such as InfoCubes with exactly the same characteristics and key figures, where an InfoCube contains the data for 2001, for example, and a second InfoCube contains data for 2002. Homogenous MultiProviders can be used to partition on the modeling level of the InfoProvider.

2. Heterogeneous MultiProviders:

   These are made up of InfoProviders that only have a certain number of characteristics and key figures in common. Heterogeneous MultiProvider can be used to simplify the modeling of scenarios by dividing these into sub-scenarios. Each sub-scenario is represented with its own InfoProvider. An example here is a sales scenario which is made up of the sub-processes order, delivery and payment. Each of these sub-processes has its own (private) InfoObjects (delivery location, invoice number, for example), and its own cross-process objects (such as customer or order number). It makes sense here to model each sub-process in its own InfoProvider, which is then combined into a MultiProvider with the other sub-processes. It is now possible to:

   – model all sub-scenarios in an InfoProvider, or

   – create an InfoProvider for each sub-scenario, and then combine these in a MultiProvider.

   The latter usually leads to a simplification of the modeling process and can improve load and read performance (see below: Executing MultiProvider Queries in Parallel).

Integration

MultiProviders only exist as a logical definition. The data is still saved in the InfoProviders it is based on.

A query based on a MultiProvider is divided into sub-queries internally. There is a sub-query for each InfoProvider in the MultiProvider. These sub-queries are usually processed in parallel.

You can find detailed information from:

Dividing a MultiProvider Query into Sub-Queries [Page 7]

Executing Sub-Queries for a MultiProvider in Parallel [Page 8]

MultiProviders with Non-Cumulative Values

Several non-cumulative cubes can also be grouped in a MultiProvider. These non-cumulative cubes can contain different time characteristics.

See also Non-cumulatives [Ext.].

Dividing a MultiProvider Query into Sub-Queries

Use

A query based on a MultiProvider is internally divided into sub-queries. There is a sub-query for each InfoProvider involved in the MultiProvider.

Functions

The division of a MultiProvider query into sub-queries can be very complex. If you have defined a query for a MultiProvider and want to display how the query has been sub-divided, you need to call up transaction RSRT. This can be a useful step if your query does not behave as expected.

To observe how the query is divided, proceed as follows:

Execute the query via RSRT, using the *Execute + Debug* option. Choose the *MultiProvider Explain* option. In the upper part of the screen, which also displays the query result, you can see messages describing how the query was divided. The following messages can appear:

- DBMAN 133: A mapping rule exists that maps a characteristic (or navigation attribute, for example) in the MultiProvider to a characteristic (or navigation attribute) of the same type (but not the same name) in the specified InfoProvider.
- DBMAN 134: The query contains a general restriction on the specified characteristic (or navigation attribute). This does not exist for the specified InfoProvider. This probably is the reason why the sub-query is left out for this InfoProvider.
- DBMAN 135: The specified key figure either does not exist in the specified InfoProvider, or was not selected for the MultiProvider. As a result, the sub-query does not read any values for this key figure.
- DBMAN 136: The sub-query for the selected InfoProvider was excluded. The reasons for this can be found in the preceding messages.
- DBMAN 137: A characteristic (or navigation attribute) does not exist in the specified InfoProvider. For this reason, all conditions in the same query column are irrelevant, and are not considered in the sub-query.
- DBMAN 138: All conditions were deleted in all query columns (see DBMAN 137). This is because they could not be filled from the specified InfoProvider. Thus, access to these was excluded.

- DBMAN 139: The query only contains key figures that do not appear in the specified InfoProvider. Thus, access to these key figures was excluded.
- DBMAN 140: A characteristic is set constantly for a particular value. This condition is not consistent with a condition contained in the MultiProvider query. As a result, access to the specified InfoProvider was excluded.
- DBMAN 141: This message describes a query restriction, to which a reference was made in the preceding message. It contains information about
  - the involved InfoCube/InfoProvider,
  - the query column (FEMS),
  - whether the condition is inclusive (I) or exclusive (E),
  - the involved characteristic (or navigation attribute),
  - the relational operator,
  - and, if applicable, the operands of the condition.
- DBMAN 144: This message describes a situation in which a restriction on characteristic A in the MultiProvider can apply to a characteristic B in the specified InfoProvider, since a restriction (of the same level) already exists for B. The specified InfoProvider reads this data without this restriction. This restriction is processed by the BW OLAP processor afterwards.
- DBMAN 145: The specified InfoObject is interpreted as a real key figure by the specified InfoProvider. This can be relevant for a MultiProvider query when all other key figures in the query are not contained in this InfoProvider, where the sub-query would need to be left out (see DBMAN 139). In this case, this option is not available.

Executing Sub-Queries for a MultiProvider in Parallel

Use

A query based on a MultiProvider is divided internally into sub-queries. There is a sub-query for each InfoProvider in the MultiProvider. Refer to:Dividing a MultiProvider Query into Sub-Queries [Page 7]

These sub-queries are usually processed in parallel.

Functions

Nevertheless, in certain special cases, described below, the system chooses sequential (non-parallel) processing:

- An entry <technical name of MultiProvider> appears in the RSADMIN table with the value (column VALUE) NOPARALLEL.
- The MultiProvider query runs in a batch process.
- The MultiProvider-Query was initiated via transaction RSRT using various debug options (SQL queries display, execution plans display, for example). Refer to:Dividing a MultiProvider Query into Sub-Queries [Page 7]
- The query requests non-cumulative key figures.
- Insufficient free dialog processes are available at the time of execution. These are required for parallel processing.
- The result of the MultiProvider query is to be saved in a file or a table.

In cases where the interim result of a MultiProvider query started in parallel is too large (more than 30000 records), the system terminates the parallel processing. This is to prevent memory overflow. In such a case, the query is automatically processed sequentially. In such cases, you should set the MultiProvider to NOPARALLEL via the entry in the RSADMIN table.

Slow Moving Item Scenario

You want to define a query that displays all products that have been purchased only infrequently or not at all. In other words, the query is also display characteristic values for which no transaction data or only low values exist for the selected period.

Procedure

In the Administrator Workbench;

1. Create a MultiProvider consisting of a revenue InfoCube, containing the InfoObject Material (0MATERIAL), and the InfoObject 0MATERIAL. The InfoObject must be set as an InfoProvider in InfoObject maintenance. In other words, you need to have assigned the InfoObject to an InfoArea. (also refer to Tab Page: Master Data/texts [Ext.]).

In the BEx Analyzer:

2. Select your MultiProvider in the Query Designer.
3. Define a query that contains the InfoObject 1ROWCOUNT in the columns.

   The InfoObject 1ROWCOUNT is contained in all “flat” InfoProviders, that is, in all InfoObjects and ODS objects. It counts the number of records in the InfoProvider. In this scenario, you can see from the row number display whether or nor values from the InfoProvider InfoObject are really displayed.

4. Save the query and execute it. All values are now displayed, including those for materials that were not purchased.

   If you filter by time (0CALYEAR, for example), values from the InfoProvider InfoObjects are not displayed since 0CALYEAR is not an attribute of 0MATERIAL. You can see this from the absence of values in the 1ROWCOUNT column in the query. If you want to restrict by time, you need to proceed as follows:

Constant Selection for the InfoObject

You need to set the constant selection for the 1ROWCOUNT key figure in order to be able to set a filter by time in this query.

1. In the Query Designer, via the context menu for 1ROWCOUNT, choose *Edit.*
2. On the left hand half of the screen, under the *data package* dimension, select the characteristic *InfoProvider* (0INFOPROV) and drag it into the right-hand screen area.
3. From the context menu for the *InfoProvider,* choose *Restrict,* and restrict across the InfoProvider InfoObject.
4. Also from the context menu for the *InfoProvider*, choose the function *Constant Selection.*
5. Save the query and execute it. You can now also set a filter for a time characteristic, the materials display remains as it was.

Displaying Slow Moving Items

If you want to display a list of slow moving items, excluding products that are selling well, you need to proceed as follows:

1. In the Query Designer, via the context menu for 1ROWCOUNT, choose *Edit.*
2. Via the context menu for *InfoProvider,* choose the function *Display Empty Values.* Also select *Constant Selection.*
3. Save the query and execute it. The result is that the system displays the materials for which there was no revenue.

Displaying Products with Small Revenues

If you want to display a list of products that have not been sold or have only been selling badly, you need to proceed as follows:

1. Set constant selection as described above, but do not select the *display empty values* function.
2. In the Query Designer, define a condition for the 0MATERIAL InfoObject. Specify a value that is to be the upper limit for a bad sale.
3. Save the query and execute it. The result is that the system displays all materials that have not been sold or have been selling badly.