What’s New with SAP NetWeaver BW 7.30 and BW Accelerator 7.20?

Applies to
SAP NetWeaver Business Warehouse 7.30 (BW7.30) and SAP NetWeaver Business Warehouse Accelerator 7.20 (BWA7.20)

Summary
This document describes the changes and new functions in BW7.30 with BWA7.20 compared to BW7.0 and BWA7.00. It provides the technical and experienced reader with a clear overview of the new functions and also gives tips and recommendations on using the functions in the user’s daily work.

Authors: Silvia Bratz, Klaus Nagel
Company: SAP AG
Created: November 2010
Latest Update: November 2011
Version: 1.3
Changes since Version 1.2

- Adjustment concerning dates and new developments
- General introduction of HANA as database for BW
- Information on where details about the BW Workspaces can be found
- Exception Aggregation for key figures with flexible currency
- Indexing of Hierarchies
- Smaller Updates e.g. on BWA only Cube

Author Biographies

**Silvia Bratz:** Development Project Manager, TIP In-Memory Platform BW (SAP AG).
Silvia Bratz joined SAP 12 years ago and began working as a technical SAP consultant for Business Intelligence. Today, she is a member of the BWA and Data Management team.

**Klaus Nagel:** Development Manager, TIP In-Memory Platform BW (SAP AG).
Klaus Nagel joined SAP 10 years ago and started as a Developer in the BW team. He now holds the position of Development Manager and leads the BWA and Data Management team.

Disclaimer

This document outlines our general product direction and should not be relied on in making a purchase decision. This document is not subject to your license agreement or any other agreement with SAP. SAP has no obligation to pursue any course of business outlined in this document or to develop or release any functionality mentioned in this document. This document and SAP's strategy and possible future developments are subject to change and may be changed by SAP at any time for any reason without notice. This document is provided 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 assumes no responsibility for errors or omissions in this document, except if such damages were caused by SAP intentionally or grossly negligent.
Table of Contents

1. Words of Welcome (Updated 1.3) ................................................................. 4
2. Introduction (Updated 1.3) ................................................................. 4
3. Upgrade .............................................................................................. 5
4. Analytic Capabilities ........................................................................... 6
   4.1. MultiProvider Cluster Access .......................................................... 6
4.2. Exception Aggregation (Updated 1.3) ............................................... 13
4.3. Top N / Bottom N ........................................................................... 36
5. Additional data in BWA ...................................................................... 38
   5.1. Master Data Provider and F4 value help with BWA (Updated 1.2) ... 38
5.2. BWA Index for VirtualProvider (Updated 1.2) ................................. 45
5.3. BWA only InfoCube (Updated 1.3) .................................................. 47
5.4. Query as InfoProvider (New 1.2) ..................................................... 52
5.5. Analytic Indexes (Updated 1.2) ......................................................... 54
   5.6. CompositeProvider (Updated 1.2) ................................................... 71
6. Flexibility – What’s Next? (Updated 1.3) ............................................ 91
   6.1. Motivation ..................................................................................... 91
6.2. BW Workspaces ............................................................................. 92
6.3. Tools ............................................................................................... 93
7. Others ............................................................................................... 94
   7.1. RSDDB - BWA Maintenance UI (Updated 1.2) ......................... 94
8. What’s New with SAP NetWeaver BW 7.30 and BW Accelerator 7.20?
   8.1. Excursion on topics ‘Partition attributes, index split and BWA landscape reorganization’ .................................................. 97
   8.1.1. Excursion on topics ‘Partition attributes, index split and BWA landscape reorganization’ .................................................. 97
   7.2. Metadata changes and BWA alignment .......................................... 99
7.3. Master data handling....................................................................... 99
7.4. Indexing Hierarchies (New 1.3) ........................................................ 100
7.5. Indexing parallelization .................................................................. 101
7.6. Master data delta Daemon ............................................................... 101
7.7. Lock & wait issues .......................................................................... 103
7.8. BWA admin parameters ................................................................... 103
7.9. Issues with table RSDDTREXNEWSID ........................................ 104
7.10. Change run .................................................................................. 104
    7.11. Deletion of master data indexes ................................................ 104
7.12. Miscellaneous (New 1.2) ............................................................... 106

BWA7.20 news ...................................................................................... 107
8. Technical information ........................................................................ 108
   8.1. Coding .......................................................................................... 108
   8.2. Important tables .......................................................................... 108
   8.3. Jobs and Logs ............................................................................... 109
9. Related Content ................................................................................ 110
Copyright............................................................................................. 111
1. Words of Welcome (Updated 1.3)

In November 2010, BW 7.30 was released as the next major release of an extremely successful product with more than 24,000 active installations (as of October 2010). SAP NetWeaver Business Warehouse Accelerator (BWA) was introduced in BW 7.0. It has made its way through the customer base and has also become the one single root for today's wider SAP strategy for in-memory computing. BWA's role has grown from both a technical and long-term strategic perspective.

Whereas BW7.0 focused on the Data Manager layer and the InfoCubes, thereby solving most of the reporting performance issues experienced by BW customers, BW7.30 with BWA7.20 branches out into the other layers of the Data Warehouse. It is now possible to move DSO data directly into BWA, either via a BWA-based InfoCube (no data persisted on the DB) or via the new HybridProvider (which automatically pushes data from a DSO into a BWA-based InfoCube).

In addition, further OLAP operations can be pushed down into the BWA engine whenever they operate on data that is exclusively located in BWA.

Now, the BWA technology has also evolved into a full blown DBMS, SAPs in-memory flagship HANA. HANA is the evolution of BWA and many of the investments going into HANA today will produce huge benefits for customers who plan to use HANA as the database for BW (RampUp of BW-HANA starts on November 7th 2011).

This paper is an important technical document that provides you with many practical insights into BWA 7.20. Discover the enormous potential of BWA. Enjoy!

Thomas Zurek
VP R&D Business Warehouse

2. Introduction (Updated 1.3)

This booklet is designed to give the reader a comprehensive overview of the new BWA-related functionality shipped in the BW7.30 release (delta to BW7.0x releases). After reading this document, the reader will know how and where to use the new functionality and will have some background information on how BWA is handled in the BW7.30 release.

This booklet is aimed at BW & BWA Administrators or consultants with technical experience. This document does not substitute the online documentation. Instead, it looks at the functionality from a different angle and provides additional guidance and insight.

This document describes the current status and will be regularly updated in SDN using the link: http://www.sdn.sap.com/irj/sdn/bwa
**BWA and HANA**

The topic of HANA as database for BW, which will go into RampUp on November 7th 2011, is not covered in this version. It is however important to understand the impact of moving from a BW + BWA landscape to a BW-HANA landscape.

Based on BW7.30 the BWA server enabled two things: Fast reporting based on a redundant storage of data in BWA’s in-memory, column store and new functionality, enabled only by the agility and flexibility in-memory technology allows.

With HANA as the database of BW the first feature is given already on the primary data set, since all data in HANA is automatically column-based and in-memory. I.e. reporting on InfoCubes is at least as fast as reporting on a BWA index before – without replicating the data! Therefore in a BW-HANA system you will no longer need to (and actually cannot) index an InfoCube as in a standard BW7.30 system.

For all new functionality delivered around BWA in BW7.30, like AnalyticIndexes, CompositeProviders, BW Workspaces, SnapshotIndexes, … from a look and feel perspective everything stays the same. I.e. all transactions, maintenance screens and process behave as before with the minor difference that the data loaded is now no longer written to a separate BWA server but directly to the HANA database.

Additional the push-down of OLAP functionality to the BWA server continues with BW-HANA. All calculation operations supported by BWA are also supported by HANA and leveraged by the BW OLAP engine.

Therefore significant portions of this document stay valid also for a BW-HANA system and while reading you can simply replace BWA with HANA.

3. **Upgrade**

Since several architectural re-designs were necessary in order to cover the broad spectrum of BWA-related functionality in BW7.30, the BWA indexes must be rebuild after the upgrade. We recommend the following approach:

Execute the BWA cleanup after the upgrade: BWA Monitor → BW Accelerator → Maintenance Function → Delete all data from BW Accelerator. This ensures that none of your BW7.0 BWA data is left over.

Rebuild the BWA indexes as necessary and appropriate via the BWA Index Maintenance (new, see below) or via process chains (as in BW7.0).

Please also note that several internal programs, functions, methods and tables have changed as well – but all externally released functionality is the same as in BW7.0! This may also be a good opportunity to make sure your system does not use any functions that have not been released.
4. **Analytic Capabilities**

4.1. **MultiProvider Cluster Access**

MultiProvider Cluster Access can be applied in every ‘OLAP situation’ if this is technically allowed by the BWA.

It can be used for any key figure in a query if more than one PartProvider is read from BWA with the same set of key figures. In case of a MultiProvider cluster access ‘higher’ operations can take place in OLAP afterwards.

**Cluster access** can be used to access the indexed data if:

- The data of **some PartProviders** in a MultiProvider is **BWA indexed** and
- All key figures of these BWA-indexed PartProviders used in a query on a MultiProvider are **mapped homogeneously** (that is, the same participating key figures from each InfoCube are identified) and
- The **PartProviders** are **identically** structured in terms of the key figures contained
Key figures that have been selected from all involved PartProviders are homogeneously mapped key figures:

![Selection of Key Figures Involved](image)

The different operations in BWA for a query can be chosen on query level in transaction RSRT via ‘Properties’ → ‘Operations in BWA’ (for more details, refer to chapter ‘Exception Aggregation’):

![Operations in BWA](image)

If key figures used in a query are supplied by the corresponding InfoCubes in a MultiProvider and option ‘3’ or ‘6’ is selected for operations in BWA, a cluster access instead of a single PartProvider access is executed.

The remaining PartProvider with key figures used in a query, which are not supplied by all InfoCubes in a MultiProvider, have to be read separately. If Option ‘2’ is selected, you can explicitly turn off the MultiProvider cluster access.

Advantages (for more details, see ‘Usage Recommendations’)

a) Reduce the load on the application server: BW splits the MultiProvider read access into single statements for the PartProvider (or splits even further for E/F fact tables, constant selection, etc.). Each statement is executed in a separate dialog process with ‘n’ process in parallel (default for ‘n’ is 6). The number of dialog processes required can now be reduced when several PartProvider are clustered and read in a single statement.

b) Reduce the amount of data transferred from BWA to the application server: When executing the UNION between the PartProvider, BWA also aggregates the data and reduces the amount of data.
Example

When a query is executed, cluster 1 and 2 are read and CUBE3 is accessed.

When a query is executed, cluster 1, 2 and 3 are accessed.
To execute analytic operations in BWA, all the necessary data needs to be in BWA.

**Example:**

A query is executed on MultiProvider MP which contains InfoCube CUBE1 and InfoCube CUBE2. The query contains key figure KYF1, which has the standard aggregation type SUM, and exception aggregation type MAX with reference characteristic CHAR1. This query contains the following sequence of steps:

1. Data is read from CUBE1 and CUBE2 with GROUP BY ‘CHAR1’,
2. UNION of both result sets is performed on the application server,
3. MAX aggregation with CHAR1 is executed on the merged data set.

Obviously step 3 (MAX aggregation) cannot be executed before the UNION of the result sets has been performed. So in order for BW to be able to push analytic operations to BWA, it was an essential ‘infrastructure development’ to also have step 2 executed in BWA (more details on the exception aggregation handling in BWA in BW7.30 can be found in the corresponding chapter).

**Standard aggregation for Queries on MultiProviders**

For queries without exception aggregation, MultiProvider mapping is performed as follows:

1. The standard PartProvider Pruning is executed based on hints and restrictions,
2. The list of PartProviders is analyzed and one cluster is generated that contains PartProviders with the following properties:
   a. PartProviders have a BWA index,
   b. The mapping between the requested fields of the PartProviders is homogenous,
   c. The mapping does not contain a ‘compounding problem’ (see SAP notes 1009987, 1086744 etc.).
3. A calculation view (CalcView) is created in BWA for this PartProvider cluster (the definition in the form of an XML document of this CalcView can be checked when executing the query in RSRT and using the debug option ‘Display SQL/BWA Query’ – see below),
4. Statements are read from the CalcView (see below for an example) and from the other PartProviders (in parallel).

This behavior is switched on by default. It can be turned off for debugging purposes in transaction RSRT using the debug option ‘Deactivate MultiProvider in BWA’.

The query runtime statistics change with this type of MultiProvider cluster access. Cluster access is indicated by an empty ‘Basic-Provider’ field and a ‘C’ in the new column ‘Cluster access’.

In this case, a new tab is displayed containing details about the cluster:
The access times for the PartProviders contained in a cluster cannot be displayed. Only the access time for the complete cluster is displayed. To analyze the time spent in BWA for a single PartProvider of a cluster, you need an in-depth analysis of the execution plan within BWA monitoring tools. In some cases, it is not possible at all to dedicate a time spent in BWA to a single PartProvider, because a lot of operations are shared among several or all parts.

**Usage Recommendations**

So what type of queries benefit from the above functionality (apart from the ones with exception aggregation of course)? As stated above, this development was an ‘infrastructure piece’ for a higher purpose and not a separate feature. Therefore the benefit for other scenarios is limited. There are 2 scenarios where it makes sense to push the MultiProvider mapping to BWA outside of the exception aggregation cases:

1. **Reduce the load on the application server:** BW splits the MultiProvider read access into single statements for the PartProviders (or splits even further for E/F fact tables, constant selection, …). Each statement is executed in a separate dialog process with ‘n’ process in parallel (default for ‘n’ is 6). The number of dialog processes required can now be reduced when several PartProviders are clustered and read in a single statement.

2. **Reduce the amount of data transferred from BWA to the application server:** When executing the UNION between the PartProviders, BWA also aggregates the data and thus reduces the amount of data. Example: The MultiProvider MP contains CUBE1 and CUBE2. Both have a BWA index and both have the same structure. CUBE1 contains the year = 2009 data, and CUBE2 contains the year = 2010 data. If the year is in the GROUP BY of the query, the result of the UNION between data from CUBE1 and CUBE2 is not smaller than the sum of both cubes.
But if year is not in the GROUP BY, the result after the UNION may be as small as $\frac{1}{2}$ of the sum of both data sets.
Example of an XML for a MultiProvider cluster:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<cubeSchema version="2" operation="createCalculationScenario" defaultLanguage="EN">
  <calculationScenario name="DBW:BIA:Q93:CS_JQ7AKUKD5PFBK3YIAAAAAAUQQXHQ"
    scenarioHints createInMemoryOnly="true" polestarEnabled="false"/>
  <dataSource>
    <olapDataSource name="DBWVC_003" source="DBW:BIA:Q93:DBWVC_003">
      <attributes>
        <allAttribute/>
      </attributes>
      <attributeGroups>
        <allCharacteristic/>
      </attributeGroups>
    </olapDataSource>
    <olapDataSource name="DBWVC_X03" source="DBW:BIA:Q93:DBWVC_X03">
      <attributes>
        <allAttribute/>
      </attributes>
      <attributeGroups>
        <allCharacteristic/>
      </attributeGroups>
    </olapDataSource>
    <olapDataSource name="DBWVC_Z03" source="DBW:BIA:Q93:DBWVC_Z03">
      <attributes>
        <allAttribute/>
      </attributes>
      <attributeGroups>
        <allCharacteristic/>
      </attributeGroups>
    </olapDataSource>
  </dataSource>
  <calculationViews>
    <bwMultiprovider name="MBWVC_003" defaultViewFlag="true" forceRequestAllParts="false">
      <filter/>
      <viewAttributes>
        <viewAttribute name="0CURRENCY_SID" isVisible="true" datatype="int" ignoreFromInputsFlag="false" forceGroupBy="false"/>
        <viewAttribute name="0UNIT_SID" isVisible="true" datatype="int" ignoreFromInputsFlag="false" forceGroupBy="false"/>
      </viewAttributes>
      <keyfigures>
        <keyfigure name="VC_COUNT" isVisible="true" datatype="int" ignoreFromInputsFlag="false" aggregationType="SUM"/>
        <keyfigure name="VC_DATE2" isVisible="true" datatype="fixed" length="17" scale="0" ignoreFromInputsFlag="false" aggregationType="MAX"/>
        <keyfigure name="VC_ORDN" isVisible="true" datatype="int" ignoreFromInputsFlag="false" aggregationType="SUM"/>
        <keyfigure name="VC_ORDW" isVisible="true" datatype="fixed" length="17" scale="3"/>
      </keyfigures>
    </bwMultiprovider>
  </calculationViews>
</cubeSchema>
```
Example of a BWA statement for a MultiProvider cluster:

```
TREX Join Index: 0bw:bia:q93:_cs_jq7akukd5fbk3yiaaaaaauqqxhq

Requested Attributes (SELECT)
0CURRENCY_SID INTO S____005
0UNIT_SID INTO S____006

Key Figures
  VC_COUNT Aggregation 01 INTO Z____683
  VC_DATE2 Aggregation 08 INTO Z____684
  VC_ORDN Aggregation 01 INTO Z____686
  VC_ORDW Aggregation 01 INTO Z____687
  VC_TURN Aggregation 01 INTO Z____663
  1ROWCOUNT Aggregation 02 INTO Z____042

Constraints
Name: $$KEYDATE$$  Value: 20100715
Join Conditions (FROM)
Query Entries (WHERE)
```

For more information, see the documentation about ‘Query Properties’ under
and about ‘Operations in BWA’ under

### 4.2. Exception Aggregation (Updated 1.3)

**What is Exception Aggregation?**

The aggregation behavior determines whether and how key figure values are aggregated in reports using the different characteristics or characteristic values.

In order to calculate the values of key figures, the data from the InfoProvider has to be aggregated to the detail level of the query and formulas may also need to be calculated. The system must aggregate using multiple characteristics.

**Exception aggregation** defines how a restricted, calculated key figure or a formula is aggregated in the Analytic Engine using one or more reference characteristics.

Calculated key figures and formulas in BW 7.30 used with exception aggregation behave in the same way, at least to a large extent. However in exceptional cases, there can be differences in the query results.
Application cases for exception aggregation include warehouse stock, which cannot be added up over time, or counters that count the number of characteristic values for a certain characteristic.

**Example:**
The key figure ‘Number of Employees’ would be added up using the characteristic ‘Cost Center’ and not a time characteristic. Here you would define a time characteristic as an exception characteristic with the aggregation type ‘Last Value’.

The sequence in which the values are calculated affects the result of the query. The aggregation types are set during the definition of the key figure. During aggregation, the Analytic Engine in BW proceeds as follows:

---

**Sequence of Aggregation Execution in the BW Analytic Engine**

1. **Standard aggregation** is executed. This is the general aggregation behavior of a basic key figure as defined in the InfoObject Maintenance (RSD1).

2. **Perform currency or unit conversion** in the Analytic Engine in case it is configured in the Query Designer. If mixed currencies / units on non-empty key figures are used, the Analytic Engine displays "" (instead of a currency/unit).

3. **Exception aggregation** determines how a restricted or calculated key figure is aggregated in the Analytic Engine with respect to one or several exception characteristics.

---

**Note:**

Even in cases where you have not included the reference characteristic in the query explicitly, an aggregation is executed implicitly using the reference characteristic.

The system also reads the values at reference characteristic level from the underlying InfoProvider for the query. Due to this increased granularity, especially when you are using reference characteristics with a large number of distinct characteristic values (instances), the system might transfer large quantities of data to the analytic engine (OLAP processor) for further processing. For example, the reference characteristic 0BPARTNER is used in the exception aggregation of a calculated key figure. 2,000,000 different characteristic values of 0BPARTNER are suitable for filtering the query. As a result, at least 2,000,000 records are transferred to the analytic engine. This means that even if the data volume that is displayed in the front end is rather small, the data volume that is to be read might be very large.
For individual descriptions and examples regarding aggregation or query processing, see the documentation in help.sap.com:

Aggregation in general:

Exception Aggregation:
http://help.sap.com/saphelp_nw70ehp2/helpdata/en/82/f2dc37f0f12313e10000009b38f8cf/frameset.htm

Query examples:
http://help.sap.com/saphelp_nw70ehp2/helpdata/en/47/6078d2dcce6834e10000000a421937/frameset.htm

In general, you can define the following types of aggregation behavior for formulas and calculated key figures in the Business Explorer Query Designer:

<table>
<thead>
<tr>
<th>Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception Aggregation</td>
</tr>
<tr>
<td>Use Standard Aggregation</td>
</tr>
<tr>
<td>Total</td>
</tr>
<tr>
<td>Maximum</td>
</tr>
<tr>
<td>Minimum</td>
</tr>
<tr>
<td>Exception If More Than One Record Occurs</td>
</tr>
<tr>
<td>Exception If More Than One Value Occurs</td>
</tr>
<tr>
<td>Exception If More Than One Value &lt; 0 Occurs</td>
</tr>
<tr>
<td>Average</td>
</tr>
<tr>
<td>Average of Detailed Values That Are Not Zero, Null, or Error</td>
</tr>
<tr>
<td>Average Weighted with Calendar Days</td>
</tr>
<tr>
<td>Average Weighted with Working Days</td>
</tr>
<tr>
<td>Counter for All Detailed Values</td>
</tr>
<tr>
<td>Counter for All Detailed Values That Are Not Zero, Null, or Error</td>
</tr>
<tr>
<td>First Value</td>
</tr>
<tr>
<td>Last Value</td>
</tr>
<tr>
<td>Standard Deviation</td>
</tr>
<tr>
<td>No Aggregation Along Hierarchy</td>
</tr>
<tr>
<td>No Aggregation of Posted Nodes Along the Hierarchy</td>
</tr>
</tbody>
</table>

Possible exception aggregation settings in the Query Designer
Which Exception Aggregation Types and Features are supported?

With SAP NetWeaver Business Warehouse 7.30, you can process specific analytical operations in SAP Business Warehouse Accelerator 7.20 directly, rather than in the Analytic Engine (OLAP processor) in SAP NetWeaver Business Warehouse, to significantly improve performance.

As a lot of calculations can be performed in BW Accelerator 7.20 with SAP NetWeaver BW 7.30, the overall query runtime will be improved. Additional scenarios can now be optimized in this release.

The following graphic shows examples of exception aggregation types. The table represents a type of fact table with booked values for characteristics and the key figure ‘Quantity sold’.

In the speech bubbles questions are addressed that can be expressed using exception aggregation. The bubbles point to the identified (MIN, MAX) or calculated results (COUNT, AVERAGE) of the aggregation.
This graphic shows the aggregation types and features which are supported in BW 7.30 / BWA 7.20 before SP05.

<table>
<thead>
<tr>
<th>Key Figure</th>
<th>SUM</th>
<th>MIN</th>
<th>MAX</th>
<th>CNT</th>
<th>CN0</th>
<th>AVG</th>
<th>AV0</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o unit or currency</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>w/ fix unit or currency</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>w/ flexible unit or currency</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This graphics show the aggregation types and features which are supported in BW 7.30 / BWA 7.20 with SP05. The corresponding BWA Revision in case BW is used with BWA can be found in the Revision Matrix under [http://www.sdn.sap.com/irj/sdn/bwa?rid=/webcontent/uuid/602253eb-bda3-2d10-a094-d906f012cad5](http://www.sdn.sap.com/irj/sdn/bwa?rid=/webcontent/uuid/602253eb-bda3-2d10-a094-d906f012cad5)

In general the BW system checks for the absolute minimum Revision number and avoids the BWA usage in case the Revision number is too low.

<table>
<thead>
<tr>
<th>Key Figure</th>
<th>SUM</th>
<th>MIN</th>
<th>MAX</th>
<th>CNT</th>
<th>CN0</th>
<th>AVG</th>
<th>AV0</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o unit</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>w/ fix unit</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>w/ flexible unit</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key Figure</th>
<th>SUM</th>
<th>MIN</th>
<th>MAX</th>
<th>CNT</th>
<th>CN0</th>
<th>AVG</th>
<th>AV0</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o currency</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>w/ fix currency</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>w/ flexible currency</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
The names of the Exception Aggregation Types can be found in domain RSAGREXC

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUM</td>
<td>Summation</td>
</tr>
<tr>
<td>MIN</td>
<td>Minimum</td>
</tr>
<tr>
<td>MAX</td>
<td>Maximum</td>
</tr>
<tr>
<td>CNT</td>
<td>Counter for all detailed values</td>
</tr>
<tr>
<td>CN0</td>
<td>Counter for all detailed values that are not null, zero or error</td>
</tr>
<tr>
<td>AVG = SUM/CNT</td>
<td>Average</td>
</tr>
<tr>
<td>AV0 = SUM/CN0</td>
<td>Average of all detailed values that are not null, zero or error</td>
</tr>
</tbody>
</table>
The most common and performance-critical case of exception aggregation is the ‘count distinct’. Other types like MIN, MAX and AVERAGE can also be pushed down to the BWA except for the case where the relevant key figure has a flexible unit.

Calculation of exception aggregation in BWA is possible for single InfoCubes and MultiProviders. To process exception aggregation in BWA, data has to be indexed permanently in BWA (or at least parts of the data). If parts of the data are not permanently BWA-indexed, these parts are read from the database and temporary indexes are created during query execution. Only the data which is really required is indexed, and not the complete PartProvider.

Calculating the exception aggregation type ‘Counter for all detailed values’ in BWA is possible for all key figure types, because it is a counter that operates independently of units/currencies.

If a key figure has a flexible unit definition and the key figure is used to define exception aggregation of type SUM, MIN, MAX or AVO, it cannot be executed in BWA (even if it contains just one single characteristic value, e.g. only unit ‘HL’). You can use key figures with a fixed unit and stage the data accordingly as a workaround, until unit conversion is pushed down to the BWA/HANA.

When the option ‘Constant Selection’ is selected, this selection becomes independent of filter settings. This means that the result remains constant during navigation. This can also be handled in the BWA.

Furthermore you can calculate a constant with exception aggregation ‘SUM’ or multidimensional exception aggregation ‘SUM’ for a constant (see examples of how a counter can be used).

Which Scenarios and Features are not supported?

The following exception aggregation functions are not supported in SAP NetWeaver BW Accelerator 7.20 (note that this is not a comprehensive ‘negative’ list). The function is executed within the ABAP-side of the Analytic Engine instead.

Calculating exception aggregation for a single key figure in BWA is not possible if the key figure uses one of the following functions:

- Non-cumulative key figures (inventory key figures)
- Key figures with elimination of internal business volume
- Virtual key figures
- Currency Conversion (possible with BW7.30 SP5)
- Unit Conversion
- Formula calculation before aggregation

If a query also contains key figures that cannot be processed with Exception Aggregation in BWA, nevertheless evaluate to execute it against BWA with option ‘6’. The figures that cannot be processed with Exception Aggregation in BWA, will then be calculated in the ABAP-side of the Analytic Engine and the key figures that can make use of the functionality are processed in BWA. Thereby you can improve the overall query runtime.
Calculating exception aggregation for a query in BWA is not possible if the query uses one of the following functions:

- Determination of time-dependent hierarchies for an event (temporal hierarchy join)
- MultiProvider with compound characteristics that are used either directly or indirectly by the query (as a reference characteristic for exception aggregation for example). At least one of these characteristics is defined in such a way that the compounding parent is supplied with homogenous data, but at least one of the contained InfoProviders does not contain the compounding child (compounding problem, see SAP note 1009987).

It is strongly recommended that you create compound-consistent scenarios (see note 1086744). In BW 7.30, it is not possible to create compound-inconsistent scenarios unless you define this explicitly on the MultiProvider level.

- Virtual characteristics

Furthermore:

- Further exception aggregation types like ‘variance’, first/last value’
  All other exception aggregation types and features available in the Query Designer and which are not mentioned above as supported types and features are not supported.

- Data Function ‘count’
  Data function ‘count’ is not supported with BWA. Use the supported options instead to create a counter see examples below on how a counter can be used in a query.

- Count scenario with Boolean Operators
  A scenario where the number of customers, for example, is calculated with a sales volume <= 1 million USD per region with a calculated key figure or formula (0VC_TURN <= 1.000.000) is not supported with BWA.
How can a counter be used in queries (examples)?

The following examples show how a counter can be built using different methods (examples a) to d)). These methods are supported by the SAP BW Accelerator 7.20. In example (e), a stacked counter (multidimensional formula aggregation counter) is created.

Example a): Global CKF Counter (Exception aggregation ‘count’ based on a calculated key figure)
Example b): Global CKF Counter (Constant ‘1’ with exception aggregation COUNT/SUM)
Example c): Formula Aggregation Counter (Constant ‘1’ with exception aggregation COUNT/SUM)
Example d): Formula Aggregation Counter (‘Key figure’ with exception aggregation COUNT)
Example e): Multidimensional Formula Aggregation Counter

Query Examples a) – e)

(Example InfoProvider: CRTREX10 with InfoCubes 0BWVC_009 and ZBWVC_009; Setting ‘6’ = Exception Aggregation in BWA)

a) Global CKF counter (Exception aggregation ‘count’ based on a calculated key figure)

Create a calculated key figure with the respective key figure (for example, OVC_TURN) only and set the exception aggregation to ‘counter for all detailed values’ on a reference characteristic (for example, OVC_PROD1).
b) **Global CKF counter (Constant ‘1’ with exception aggregation ‘COUNT’ or ‘SUM’)**

1. **Display Calculated Key Figure 'CKF_20091210191139'**
   - Description: CKF_20091210191139
   - Technical Name: CKF_20091210191139
   - Use Standard Text
   - Detail View: OVC_TURN

2. **Change Calculated Key Figure**
   - Aggregation
     - Exception Aggregation
     - Ref. Characteristic: OVC_PROD1
   - Calculation After Aggregation
c) Formula Aggregation ‘count’ (Constant ‘1’ with exception aggregation ‘COUNT’ or ‘SUM’)

Formula aggregation allows you to perform effective counting. The formula just contains a 1 with exception aggregation COUNT or SUM.
d) **Formula Aggregation ‘count’ (‘Key figure’ with exception aggregation COUNT)**

![Formula Aggregation 'count' (‘Key figure’ with exception aggregation COUNT)](image)

Query results for a) to d)

![Query results](image)
e) Multidimensional Formula Aggregation ‘count’ (based on a formula with exception aggregation ‘count’ on another component with exception aggregation)

If you need to count tuples (Product, Customer), you can define two formulas:

- First, formula 1 to represent the ‘Product Counter’: with exception aggregation SUM over Product (see example c) above
- Second, formula 2 to represent ‘Product and Customer Counter’: ‘Product Counter’ with exception aggregation SUM over Customer (Local Redefinition of exception aggregation through formula)

```
Change Formula

General | Aggregation | Display | Extended | Conversions | Planning | Calculations

Description | Technical Name

Formula = 1 (Exc Aggr. TOTAL on Product)

Change Formula

General | Aggregation | Display | Extended

Description | Technical Name

Formula 2 (Exc Aggr. TOTAL on Person)

Query Result
```
There are 58 combinations of Product and Customers for color ‘brown’.

How can I enable BWA processing for Exception Aggregation and what are the prerequisites?

a) **Enablement on query level in transaction RSRT → Properties**

Available operations
Explanation of Available Operations

Option 0 ‘No operation in BWA’

Select this option if you do not want the query to use an existing BWA index. It makes the option in SAP note 1161525 obsolete (Table: RSRREPDIR with field NOHPA = X).

If the option ‘do not use BWA due to virtual characteristics/key figures’, which is now an obsolete option, - see note 1161525 for details - has been set before the update to version 7.30 or higher, the option ‘No operation in BWA’ is set after the update.

Option 2 ‘Single access per InfoProvider (as in Release 7.0)’

With BWA access, only one InfoProvider is read (corresponds to the behavior of executing a query in BW 7.0x with BWA Release 7.00). BWA is used in the same way as described under Option ‘3’. The only difference is that in case of a MultiProvider for each involved InfoProvider a separate access to its BWA index takes place.

Option 3 ‘Standard’

This option is set by default and enables the use of BWA indexes. If the query contains virtual characteristics or key figures, the system shows an appropriate message. For queries on MultiProvider with homogeneously mapped key figures (and where the corresponding PartProvider are BWA indexed), cluster access is used (see chapter ‘MultiProvider Cluster Access’).

Option 6 ‘Exception Aggregation’

This option enables the specific exception aggregation calculations in BWA which are mentioned above. A cluster is also accessed here for queries on MultiProvider with homogeneously mapped key figures (and where the corresponding PartProviders are BWA indexed). If the system cannot calculate the exception aggregation in BWA because of the functions used, it uses the standard option (operations in BWA = ‘3’) instead.

b) Enablement on InfoProvider level in transaction RSDIPROP

Using transaction RSDIPROP, you can set the option ‘operation in BWA’ on the InfoProvider level, which is the default setting.

c) Enablement for multiple Queries in transaction RSRT → Environment → Query Mass Maintenance

In RSRT (properties) you can set the option ‘operation in BWA’ per query or take over the setting from the InfoProvider. If you choose ‘query mass maintenance’ you can change the setting for multiple queries at the same time. If a query has already been created and the option is changed on InfoProvider level, this does not take effect when the query is already there and the option in the query is not set to ‘InfoProvider specific’. In this case the ‘query mass maintenance option’ can be used.
d) Prerequisites and Processing of Exception Aggregation

Exception Aggregation is processed in SAP BW Accelerator:

- If the option for operations in BWA of the query is set to ‘6’ and
- As soon as at least one PartProvider of the MultiProvider has a permanent BW Accelerator index (created in transaction RSDDB) and
- The query, key figure and formula definitions meet the condition to be executed with BWA (see chapter ‘supported types and features’) and
- None of the PartProviders have a compounding problem for any of the requested characteristics

If parts of the data are not BWA-indexed, these parts are read from the database and temporary indexes are created during query execution. Only the data which is really required is indexed, and not the whole PartProvider. The temporarily created indexes are removed once they have been used (leave query result or change navigation state). These indexes cannot be reused.

However, you should only choose option ‘6’, if most of the data is permanently indexed in the BWA.

If the BWA cannot calculate the Exception Aggregation due to missing functionality and option ‘6’ is set, the option ‘3 = standard’ is used automatically. This means that the features are processed by using BWA in the ‘classic way’ (for selection/aggregation) and the OLAP parts are executed in the ABAP-side of the Analytic Engine.

In these cases, the user is notified via a message (see note 1489846).

We recommend that you also process OLAP functionality in the BW Accelerator. But depending on the overall data model, system resources and query layout, there are some situations where OLAP functionality should not be used or does not show a significant benefit.

As there are many dependencies (e.g. BW Accelerator status etc.) and possible influential factors (for example, objects involved in the data model), you can only check and verify whether OLAP functionality is useful in the project environment.
What are the current recommendations for the usage?

Due to the fact that there are many dependencies for processing Exception Aggregation in BWA, it is not possible to implement a logic to automatically set the correct option for query processing (3 = standard or 6 = exception aggregation in BWA).

The dependencies are the:

- Data Model
- Data Volume
- Request Status
- BWA Indexed PartProvider
- BWA Status (Available Memory)

It is beneficial for the performance of Exception Aggregation in BWA if all PartProviders are permanently indexed in BWA or if the data volume to be temporarily indexed is small or can be indexed faster with BWA than in the ABAP-side of the Analytic Engine.

The temporary result set to calculate Exception Aggregation might not fit in the memory of the corresponding blade when using the SAP BW Accelerator. Anyhow the intermediate result set for exception aggregation can become much larger before the query breaks, if exception aggregation is executed in BWA, in comparison to exception aggregation executed within ABAP. This means you can calculate data volumes that you could not calculate before or at least calculate them with much lower query runtimes than before.

Queries with counters or average calculations in particular benefit significantly from execution in BWA.

<table>
<thead>
<tr>
<th>Nr. of records after 1st aggr.</th>
<th>Runtime DB (sec)</th>
<th>Runtime BWA 7.0 (sec)</th>
<th>Runtime BWA 7.2 (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,163,222</td>
<td>154.62</td>
<td>139.64</td>
<td>7.28</td>
</tr>
<tr>
<td>10,000,008</td>
<td>320.73</td>
<td>280.09</td>
<td>27.8</td>
</tr>
</tbody>
</table>

Where can I see if a key figure has been processed with Exception Aggregation?

In transaction RSRT you can execute a query with option ‘Execute and Explain’. The following example shows, how the system informs, if Exception Aggregation has been used. It also shows the reason why it has not been used.

Execution of the query in RSRT with option ‘Execute and Explain’
Execute query in RSRT with option 'Execute and Explain'.

Choose 'Exception Aggregation in BWA'.

GoTo a key figure and choose 'Enhanced Menu' → 'Key Figure Definition'.
Information on key figure '0VC_ZHL': ‘For key figure 0VC-ZHL exception aggregation is executed in BWA.'
Patch Level of Customer System:

Log for Exception Aggregation in BWA

User wants exception aggregation in BWA

Query SD_20091111_CS_PART_CNT_3 contains exception aggregations for BWA execution

Structure element combinations (KIDs) whose exception aggregation can take place in BWA

<table>
<thead>
<tr>
<th>KID</th>
<th>WGR</th>
<th>AGGREXC</th>
<th>KYFRM</th>
<th>CONSTANT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1003</td>
<td>CNT</td>
<td>CNT</td>
<td>0VC_ZHL</td>
<td></td>
</tr>
<tr>
<td>1004</td>
<td>CNT</td>
<td>CNT</td>
<td>0VC_ZHL</td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td>CNT</td>
<td>CNT</td>
<td>0VC_ZHL</td>
<td></td>
</tr>
<tr>
<td>2005</td>
<td>CNT</td>
<td>CNT</td>
<td>0VC_ZHL</td>
<td></td>
</tr>
</tbody>
</table>

Structure element combinations (KIDs) that exception aggregation cannot take place for in BWA

<table>
<thead>
<tr>
<th>KID</th>
<th>WGR</th>
<th>AGGREXC</th>
<th>KYFRM</th>
<th>CONSTANT</th>
<th>REASON</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>1</td>
<td>OVC_MAX</td>
<td></td>
<td></td>
<td>No exception aggregation for WGR</td>
</tr>
<tr>
<td>1002</td>
<td>2</td>
<td>OVC_MAX</td>
<td></td>
<td></td>
<td>No exception aggregation for WGR</td>
</tr>
<tr>
<td>2001</td>
<td>1</td>
<td>OVC_MAX</td>
<td></td>
<td></td>
<td>No exception aggregation for WGR</td>
</tr>
<tr>
<td>2002</td>
<td>2</td>
<td>OVC_MAX</td>
<td></td>
<td></td>
<td>No exception aggregation for WGR</td>
</tr>
<tr>
<td>3001</td>
<td>1</td>
<td>OVC_MAX</td>
<td></td>
<td></td>
<td>No exception aggregation for WGR</td>
</tr>
<tr>
<td>3002</td>
<td>2</td>
<td>OVC_MAX</td>
<td></td>
<td></td>
<td>No exception aggregation for WGR</td>
</tr>
</tbody>
</table>

Log for Exception Aggregation in BWA

User does not want exception aggregation in BWA

Query /VIRTUAL-000001 contains exception aggregations for BWA execution

Structure element combinations (KIDs) whose exception aggregation can take place in BWA

<table>
<thead>
<tr>
<th>KID</th>
<th>WGR</th>
<th>AGGREXC</th>
<th>KYFRM</th>
<th>CONSTANT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>SUM</td>
<td>0VC_ZHL</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>SUM</td>
<td>0VC_ZHL</td>
<td></td>
</tr>
</tbody>
</table>

Structure element combinations (KIDs) that exception aggregation cannot take place for in BWA

<table>
<thead>
<tr>
<th>KID</th>
<th>WGR</th>
<th>AGGREXC</th>
<th>KYFRM</th>
<th>CONSTANT</th>
<th>REASON</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>0VC_ZHL</td>
<td></td>
<td></td>
<td>No exception aggregation for WGR</td>
</tr>
</tbody>
</table>
Execution in the SAP Business Explorer Analyzer

GoTo a key figure and choose: ‘Key Figure Definition’
Information on key figure ‘0VC_ZHL’: ‘For key figure 0VC_ZHL exception aggregation is executed in BWA.’
4.3. **Top N / Bottom N**

Top N (and the same applies to Bottom N) is defined in an analytic query as a ‘condition’ – a condition is a display filter and has no effect on the (sub-) totals. The conditions are evaluated as the last step of OLAP processing, before the data is handed over to the frontend layer. This has a decisive effect on how the data is handled during navigation and also the detail level in the OLAP cache.

This type of OLAP operation cannot be simply pushed down to BWA without significantly changing and endangering the behavior of many existing scenarios, because the OLAP operations cannot generally be swapped.

But with TOP N conditions in a pre-query, it is possible to push down the TOP N condition to BWA in scenarios without changing the behavior at all (a pre-query that fills a list of values used as filters for a subsequent and main query). The result of the pre-query filter has an impact on the complete query result, and also on the (sub) totals.

Example: Query TOPN_KNTEST shows the ‘Top 3’ customers with respect to key figure 0VC_ZHL. Query TOPN_KNTEST_MAIN has a global filter with a variable based on a replacement path which is filled by the TOPN_KNTEST query.

![Example diagram of Top N/Bottom N conditions](image)

*Condition defined in query TOPN_KNTEST*
Query result for TOPN_KNTEST_MAIN.

TOPN_KNTEST returns 3 customer IDs that are used as a filter in the TOPN_KNTEST_MAIN query. In this case, only 3 values are returned from BWA as the pre-query result set and not the complete list of customers and the OLAP processor filters afterwards.

<table>
<thead>
<tr>
<th>Session UID</th>
<th>Step UID</th>
<th>Step Count</th>
<th>Start Time</th>
<th>Handle Type</th>
<th>Info</th>
<th>Object Name</th>
<th>Event ID</th>
<th>Event Text</th>
<th>Duration</th>
<th>Counter</th>
<th>Event Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>0610</td>
<td>OLAP: Query Qam</td>
<td>0.01960</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3999</td>
<td>OLAP: Other Time</td>
<td>0.02081</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3999</td>
<td>OLAP: Other Time</td>
<td>0.00345</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>5400</td>
<td>Authorization Buffer</td>
<td>0.01230</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>5610</td>
<td>OLAP: Exit Variables</td>
<td>0.000082</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3600</td>
<td>OLAP: Initialization</td>
<td>0.01317</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>4200</td>
<td>Value Authorization</td>
<td>0.005972</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3600</td>
<td>OLAP: Settings</td>
<td>0.03187</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>2560</td>
<td>Cache Generation</td>
<td>0.000913</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3910</td>
<td>Total DISIBL</td>
<td>0.00000</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>9611</td>
<td>Total DISIBL</td>
<td>0.00000</td>
<td>22</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>9600</td>
<td>Data Manager</td>
<td>0.025104</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>1110</td>
<td>OLAP: Data Selection</td>
<td>0.004136</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>2520</td>
<td>Cache Committ</td>
<td>0.003231</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3100</td>
<td>OLAP: Read Data</td>
<td>0.003019</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3200</td>
<td>OLAP: Transfer</td>
<td>0.005159</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>39952</td>
<td>3 x Qvry Query Dists</td>
<td>0.004029</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3910</td>
<td>OLAP: Query Qam</td>
<td>0.11664</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3999</td>
<td>OLAP: Other Time</td>
<td>0.010024</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3999</td>
<td>OLAP: Other Time</td>
<td>0.001074</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3999</td>
<td>OLAP: Other Time</td>
<td>0.004153</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3600</td>
<td>OLAP: Initialization</td>
<td>0.031207</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3600</td>
<td>OLAP: Settings</td>
<td>0.009541</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>2560</td>
<td>Cache Generation</td>
<td>0.003279</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3911</td>
<td>Total DISIBL</td>
<td>0.00000</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>9611</td>
<td>Total DISIBL</td>
<td>0.00000</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>9600</td>
<td>Data Manager</td>
<td>0.11250</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3110</td>
<td>OLAP: Data Selection</td>
<td>0.001593</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0002TH0001</td>
<td>0002TH0001</td>
<td>11 11 2010</td>
<td>08 35</td>
<td>OLAP</td>
<td>OWVNC_CTO</td>
<td>TOPN_KNTEST_MAIN</td>
<td>3100</td>
<td>OLAP: Read Data</td>
<td>0.001508</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Query runtime statistics for query TOPN_KNTEST_MAIN with pre-query TOPN_KNTEST. Note that the DBSEL Counter is 0 for the pre-query where the ‘Top 3’ condition was evaluated directly in BWA, because it is no longer possible to determine what this number is.

Usage/limitations:

- The same key figure limitations as for the exception aggregation apply here, i.e. only key figures w/o unit or currency conversion or with fixed currency conversion can be used.
- The Top N condition is evaluated by the BWA, even if the pre-query is on a MultiProvider. Handling and limitation is exactly the same as for the exception aggregation described above.
- To push the Top N condition in the pre-query to BWA, the pre-query has to be set to ‘Operations in BWA = 6’ in RSRT – the same as for the exception aggregation.
5. Additional data in BWA

This chapter gives an overview of some of the new functions for BWA 7.20 in the BW7.30 release. It also provides some background information in addition to the usual online help and provides guidance on how and where to use the new functions.

For more information, see the documentation about ‘Indexing BW Data in SAP NetWeaver BW Accelerator’ under http://help.sap.com/saphelp_nw73/helpdata/de/4c/2c87e2477f51e6e100000000a42189b/frameset.htm.

5.1. Master Data Provider and F4 value help with BWA (Updated 1.2)

To enable the F4 value help for BWA, additional data has to be indexed in BWA: Texts (in all languages), Display Attributes, and the texts of the Navigation/Display Attributes. Since a BWA-based F4 help is not always required or necessary, this data is not indexed by default, but only on request. Call transaction RSDDDB to select the InfoObject and trigger indexing process.

BWA index (not F4-enabled) for InfoObject 0VC_PROD2 and logical index (JOIN construct) for F4 help
You can also enable the F4 value help with BWA for InfoObjects by selecting the InfoObjects of an InfoCube when creating the BWA Index for the InfoCube. You do this in the ‘InfoObject Options’ Tab of the BWA-Index Maintenance screen. The list of InfoCube characteristics is displayed and the ‘Master Data Index’ column is input-enabled, if the InfoObject can have a BWA index and does not already have one. The BWA indexes for the InfoObjects are created automatically during the indexing process for the InfoCube.

If a BWA index exists for an InfoObject, the F4 value help behaves as follows:

- Type ‘D’ (Only Values in InfoProvider): Statement is executed reading from BWA if all PartProviders are indexed.
- Type ‘M’ (Values in Master Data): Statement is executed against master data index in BWA.
- Type ‘Q’ (Only Posted Values for Navigation): Statement is executed against master data index in BWA. The IN list with booked values from navigation is used as a filter condition.
The following example shows the statements for the F4 value help against InfoObject 0VC_PROD2 in a variable popup during query execution (on a MultiProvider with all PartProviders in BWA) in mode ‘Only values in InfoProvider’ and ‘Values in Master data tables’.
Only values in InfoProvider

1. A CalcView is created for the MultiProvider UNION with only the SID of OVC_PROD2 as a field,
2. The booked values are read using the single field of the CalcView (SID of OVC_PROD2). The result is not transferred to the application server. Instead it is stored as a temporary index in BWA.

```xml
<cubeSchema version="2" operation="createCalculationScenario"
      defaultLanguage="DE">
  <calculationScenario name="OBW:BIA:Q93:cs_jtzkwox3iusbvyanalauqqxhq">
    <scenarioHints createInMemoryOnly="true" polestarEnabled="false"/>
    <dataSources>
      <olapDataSource name="OBWVC_003" source="OBW:BIA:Q93:OBWVC_003">
        <attributes>
          <attribute name="__numoffacttablerows" isVisible="true"
                     datatype="int" ignoreFromInputsFlag="false" isKeyfigure="true">
            <keyfigureProperties aggregationType="COUNT"/>
          </attribute>
        </attributes>
      </olapDataSource>
      <olapDataSource name="OBWVC_X03" source="OBW:BIA:Q93:OBWVC_X03">
        <attributes>
          <attribute name="__numoffacttablerows" isVisible="true"
                     datatype="int" ignoreFromInputsFlag="false" isKeyfigure="true">
            <keyfigureProperties aggregationType="COUNT"/>
          </attribute>
        </attributes>
      </olapDataSource>
      <olapDataSource name="OBWVC_Z03" source="OBW:BIA:Q93:OBWVC_Z03">
        <attributes>
          <attribute name="__numoffacttablerows" isVisible="true"
                     datatype="int" ignoreFromInputsFlag="false" isKeyfigure="true">
            <keyfigureProperties aggregationType="COUNT"/>
          </attribute>
        </attributes>
      </olapDataSource>
    </dataSources>
    <calculationViews>
      <bwMultiprovider name="MBWVC_0KN" defaultViewFlag="true"
                        forceRequestAllParts="false"/>
    </calculationViews>
  </calculationScenario>
</cubeSchema>
```
3. Texts and attributes are read from the special F4 value help logical index for the InfoObject with filtered on the booked values, by using the temporary index as a JOIN condition.

Values in Master Data tables

The statement is executed directly against the BWA index for the InfoObject – same statement as above, but without the JOIN filter with the temporary index.

Usage recommendations

We do not recommend activating BWA indexes for all InfoObjects by default. You should only select InfoObjects where the users experience bad performance with the F4 value help. These are usually InfoObjects with a lot of characteristic values and attributes.
Maintenance processes for the additional master data indexes

The Navigational and Display-Attributes are updated in BWA during the standard Attribute Changerun. The text indexes are updated automatically once new texts have been loaded. If the system cannot index the texts afterwards (e.g. due to a communication error w/ BWA), the texts in the database tables are not rolled back. This means that the database and BWA have different versions of the texts. The text indexing operation must be restarted and completed to correct this problem.

Texts in queries

After the query result has been determined the texts of InfoObjects are selectively read from the database. As BWA is not faster in this case than the database (with index access), the BWA is not used here.

Fuzzy Search with BWA

It is possible to activate a fuzzy search when running the F4 value help against BWA. In other words, not only are exact matches displayed, but also similar values are found. In this case, similar means that less than n steps are required to convert value1 to value2, where n = (# of characters) * 0.6. The fuzzy search is currently only used for the text fields.

The fuzzy search is deactivated by default. To activate the search (system-wide for all BWA-based F4 value helps), you have to set the parameter RSMD_RS_TXTSEARCH = ‘F’ in table RSADMIN.

Search result with exact search
Search result with fuzzy search

Master Data Provider with BWA

If the F4-value help is activated for an InfoObject, Queries on the Master data Provider are executed against the BWA. For technical reasons, there is only one exception: The InfoObject contains time-dependent and time-independent attributes and the query reads a time-dependent key figure attribute from the Master Data Provider. This query now reads the data from the database.

Limitations

It is only possible to activate the F4 value help with BWA for InfoObjects with generic master data access. InfoObjects with separate master data read classes cannot have a BWA index.

If an InfoObject is used just as an Attribute (not as characteristic in an InfoCube dimension) it cannot be selected in RSDDB for indexing.

If an InfoObject is referencing another one e.g. CRM_PARTNER is created with reference on 0BPARTNER, CRM_PARTNER itself cannot be indexed because it shares the master data of 0BPARTNER which can be indexed.
5.2. BWA Index for VirtualProvider (Updated 1.2)

With BW7.30, it is possible to create a BWA index for a snapshot of the data in a VirtualProvider. The physical layout of the BWA index is the same as for an InfoCube BWA index. The facts are stored in an F index with JOINs to the shared master data indexes modeled in the logical index on top. Any query reading data from the VirtualProvider automatically is re-directed to BWA, if the BWA index is ‘active’.

To create a BWA index for a VirtualProvider, select the ‘VirtualProvider’ button in transaction RSDDB. The ‘BWA Index Settings’ tab now has additional properties that can be maintained. Since there is no DB fact table to count the number of records, it is essential that you enter an estimation of the number of fact records for the VirtualProvider fact index. Otherwise the system will not split the fact index into several parts in BWA and this leads to reduced read performance with large fact indexes. It is not necessary to give the exact number, but rather the correct size category.

You also have the option of entering an ABAP class name and a delta characteristic. The class can implement two delivered Interfaces (an example and a description are supplied with the coding of class CL_RSDDB_BVIP_SUPER).
- IF_RSDDB_BVIP_PARALLEL: When this interface is implemented, the initial indexing process can be split up into several processes by using a customer-defined partitioning (ranges) of the source data (using the interface method GET_RANGES). The system does not check whether or not the ranges overlap or cover the complete range of available data – this is purely the responsibility of the customer.

- IF_RSDDB_BVIP_DELTA: When this interface implemented, a delta load can be implemented to the BWA index of the VirtualProvider. This is only possible if the underlying data source provides a real delta and has a simple delta criterion that can be described in a single read pointer which is the value of one of the VirtualProvider characteristics. This characteristic has to be specified as the ‘delta characteristic’ for the BWA index of the VirtualProvider.
The following code example shows how this read pointer can be found using the current date minus one day. The initial indexing process can then load all the data that is less or equal to the load date and each daily load can load the data from the previous day and add it. The delta characteristic could then be OCALDAY, for example.

```plaintext
IF_RSDDB_BVIP_DELTA_GET_READPOINTER

METHOD if_rsddb_bvip_delta_get_readpointer.

  DATA: l_date TYPE sy-datum.
  l_date = sy-datum.

  l_date = l_date - 1.
  r_readpointer = l_date.

ENDMETHOD.
```

Recommendations

A VirtualProvider BWA index should only be used where data is not changed frequently in the source or where the real-time aspect of the VirtualProvider is not important.

Please note that the snapshot contains not only a snapshot of the data but may also contain user-specific data - in this example, the user who executed the indexing process. If authorization-relevant filters are applied directly in the data source of the VirtualProvider, users running queries reading from the VirtualProvider using the BWA-Index might see data for which they have no authorization!

For more information, see the documentation about the ‘VirtualProvider’ under http://help.sap.com/saphelp_nw73/helpdata/en/4C/2C87E2477F51E6E10000000A42189B/frameset.htm

5.3. BWA only InfoCube (Updated 1.3)

What is a BWA only InfoCube?

A SAP Business Warehouse Accelerator only InfoCube (InfoCube with data persistency only in the SAP BW Accelerator) is an InfoCube with transaction data that is exclusively persisted in the BWA. The metadata of the InfoCube (InfoCube definition) is stored on the database. Therefore, it is not a new InfoProvider type but a new flavor of the standard BW InfoCube.

The package dimension index is the only table filled in BW when transactional data is loaded, as this index is needed to manage the BW requests. In addition, one record per data package (with the aggregated sums of the key figures and information on the corresponding request) is written to the Fact table of the InfoCube. This enables consistency checks to be performed on sum-level (Transaction RSRV ➔ ‘Check key figure totals in database against index’).

The fact table index is flat, i.e. the master data SIDs (Surrogate IDs) are directly written to the fact index without dimension indexes, except for the package dimension.
What is the purpose of the BWA only InfoCube?
The SAP Business Warehouse Accelerator only InfoCube offers increased value to the Enterprise Data Warehouse (EDW) implementation and can simplify the Layered Scalable Architecture (LSA) of the EDW.

It offers the following main advantages:

- Saves disc space on the database
- Prevention of data redundancy
- Reduction of load on the database
- Moves the reporting layer (not the warehouse layer!) to the BWA server
- This means the restriction not to report on the EDW layer data (i.e. Propagator layer DSO data) no longer applies.
- Flat F table index can speed up data staging significantly as no DIMIDs (Dimension IDs) have to be determined (in some scenarios, the creation of DIMIDS is the most time-consuming part).
- No need to create dimensions based on performance criteria. In InfoCube maintenance, the dimensions can still be created for semantic structuring. You simply define all characteristics in a single dimension of the BWA only InfoCube, whereas the multi-dimensional model only exists on the virtual/semantic layer of a MultiProvider.
- Standard InfoCubes that were previously created due to performance reasons will become obsolete.
- BWA only InfoCubes can be used with a HybridProvider. This means you can achieve a tight coupling between the DataStore object and the BWA.

From a reporting perspective, the BWA only InfoCube is the same as a standard InfoCube indexed in BWA. However, please consider the restrictions of this object carefully.
How to create and fill a SAP Business Warehouse Accelerator only InfoCube

In general, there is no difference to the process of creating an InfoCube in the Data Warehousing Workbench.

There are two options to choose from: ‘InfoCube stores its data in the database’ and ‘InfoCube only stores its data in the BWA’
When creating a DTP, you can use any source that allows a DTP to load data into the BWA only InfoCube. The DTP uses a generated report to write the data directly into the fact index on the BWA. The program is generated during the design time of the BWA only InfoCube. The SIDs (Surrogate IDs) for the master data are created in the same way as for regular InfoCubes during the runtime of the DTP. This means that the data on the BWA is SID-based (the same as with a standard InfoCube). Hierarchies and navigational attributes can be used without any limitation.

**What are the current restrictions, prerequisites and recommendations for deployment?**

If you intend to use BWA only InfoCubes, an SAP BW Accelerator is a prerequisite. Nevertheless it is not required to have a BWA installed on the Development and Quality Assurance system as well. If you activate a BWA-only cube in a system without maintained RFC-connection to a BWA, the activation will succeed without problems because the indexing part is simply skipped due to the missing RFC-connection. Data loads to these systems (DEV/QA) will certainly fail.

The data is exclusively saved on the BWA and there is no automatic recovery mechanism available.

If you model a DataStore object as a data layer and use the BWA only InfoCube as a reporting layer, you can recover data (e.g. due to a corrupt index) by running a DTP on the underlying DataStore Object (DSO).

Real-time InfoCubes - as used in integrated planning - cannot be defined as BWA only InfoCubes due to the open request handling.

The data is staged via standard DTP to the BWA only InfoCube. The load can only be parallel processed after serial extraction. SAP BW 3.x data staging mechanisms cannot be used.

It is not possible to delete single records from a fact index of the BWA. This means that **selective deletion and archiving, which internally uses selective deletion, cannot be executed** for an InfoCube in BWA. For an InfoCube with database persistency and BWA index on top, the index can be dropped and rebuilt from the fact table.

As there is no fact table on the database for the BWA only InfoCube also no drop and rebuild in case of selective deletion can be done. However, you have the option to drop the InfoCube content, selectively delete it from the underlying DataStore object and reload it to the BWA only InfoCube. Furthermore, it is not advisable to choose a BWA only InfoCube when **request-wise deletion** is performed frequently. In this case, it is recommended that you recreate the BWA index from time to time, unless you can reload from an underlying DSO.

There is **no possibility to compress** a BWA only InfoCube. Therefore it is not recommended that you use this object when a lot of changes, for example reverse bookings, are loaded. If the changes are significant and an underlying DSO is available, reload the data on a regular basis from the source DataStore object. Here the data is compressed in the active table.

Furthermore as you cannot use compression with a BWA only InfoCube, it is also **not possible to compress with zero elimination**. Compression in general is mostly not needed in the BWA use case from a performance perspective. However it is still used for ‘semantic reasons’. When using a standard InfoCube with BWA index, you can rebuild the BWA index after having executed ‘compression with zero elimination’ and use this feature - originally designed as a database performance feature - also due to ‘semantic reasons’.
Finally, we do not recommend the use of a BWA only InfoCube when **non-cumulative** key figures for inventory handling are included, because compression is needed for the recalculation of the reference point.

5.4. Query as InfoProvider (New 1.2)

What is a Query as InfoProvider (aka QueryProvider)?

You can use a Query and release it as QueryProvider in transaction RSDDB. When creating a QueryProvider, an InfoProvider in the '@Q'-namespace is derived. From a technical point of view, the query is executed and the result is written to a flat table when the BWA-Index for the QueryProvider is created.

The option to release a query as QueryProvider in RSDDB is similar to the flag 'Query is used as InfoProvider' in RSRT.

What is a Query Snapshot Index?

When you index such a QueryProvider on the BWA, the so-called Query Snapshot Index (QSI) is created. The QSI is a snapshot with index type ‘ICQ’.
What is the purpose of the QueryProvider/Query Snapshot Index?

It can make sense to process data in a query as an InfoProvider. The advantage in doing this is that you can calculate the query’s functions in the Analytic Server before reading the data. This improves performance, especially in the case of simple queries that contain a very large amount of data. Furthermore with this, data of all possible InfoProviders (InfoSet, DSO etc.) can be indexed on the BWA.

The following are scenarios for using a query as an InfoProvider:

- You can use a QueryProvider as the data source for an analysis process and use SAP NetWeaver BW Analysis Process Designer to create analytical indexes on it
- You can use a QueryProvider - especially if it contains mass data – over the Analytic Server’s relational interface as a data source for the SAP BusinessObjects Data Federator and perform analyses with this, for example in SAP BusinessObjects Web Intelligence.
- You can use the Query Snapshot Index with Explorer, accelerated
- Also the CRM segmentation capabilities are based on a QueryProvider which must be indexed on BWA.

The following functions are supported:

- Standard aggregations (SUM, MIN, MAX)
- Formulas that are totaled before aggregation (+, −, constant factor)
- Units and currency translations
- Constant selection
- Formula with quotients, like prices
- Formula with exception aggregation

How to create and fill a Query Snapshot Index?

In transaction RSDDB you can choose ‘QueryProvider’ and press button ‘release query as InfoProvider’. Afterwards you can index this QueryProvider on the BWA as a snapshot by entering the name of the Query and push button ‘Create’.

What are the limitations of a QueryProvider/Query Snapshot Index?

Not all query functions are supported which means that not all queries can be used as QueryProvider.

The index on the QueryProvider is a snapshot index. The up-to-dateness of data in the underlying Provider of the QueryProvider could already have changed and the data of the QSI is older.

The up-to-dateness of the data can be verified in transaction RSDDDB.

In the CRM segmentation use case you can only report on a QSI, not on the QueryProvider. Nevertheless you can schedule the rebuild of the QSI to receive a QSI with current data. For more information, see the documentation about the ‘QueryProvider’ under

Flexibility

5.5. Analytic Indexes (Updated 1.2)

In this chapter the Analytic Index created in SAP BW Accelerator with Analysis Process Designer is described.

What is an Analytic Index?

An Analytic Index is a data container with data exclusively saved in the SAP BW Accelerator as primary persistence. The Analytic Index stores its data in a simple star schema. It contains facts and characteristics (called dimensions) with attributes. When creating an Analytic Index, an InfoProvider in the ‘@3’-namespace is derived, which can directly be used to report on every other InfoProvider. The metadata of an Analytic Index is also saved in the database of the BW system to preserve its definition.

In the Analysis Process Designer (APD; transaction RSANWB) the Analytic Index is used as a new target object to persist results calculated from an Analysis process.

What is it the purpose of an Analytic Index?

The Analytic Index empowers the IT department to perform rapid prototyping and easily create ad-hoc scenarios with the SAP BW Accelerator (BWA) inside the SAP NetWeaver Business Warehouse. The creation process is simple as there is no need to create InfoObjects for characteristics and key figures in advance, for example, when creating an Analytic Index. Therefore you can create objects which do not affect other BW scenarios and do not depend on BW content objects.

The Analysis Process Designer is a tool for creating and filling an Analytic Index. It is considered the most suitable due to its ad-hoc capabilities. For example, additional data can be quickly uploaded to the SAP BW Accelerator, by using a flat file as the source in APD.

An Analytic Index can be used within a CompositeProvider, which models unions and joins of different data providers. This enables you to perform ad-hoc reporting on combined data (for example, joins of BW data and flat file data). Furthermore, the query result can be saved in an Analytic Index (=cache pre-calculated queries).

How to create and fill an Analytic Index

The following example shows how to create and load an Analytic Index via an analysis process. Transactional data and attributes for the characteristic ‘Product’ in this example are available in different Excel sheets. When defining an Analytic Index for dimensions and facts and attributes for a characteristic are to be filled, this has to be taken into account in advance. This is because it is currently not possible to change the Analytic Index definition later on. Attributes can also be loaded from a file if required.
a) Creation of an Analytic Index for Facts with source data from an Excel file via APD

Example of a simple star schema

Example Excel File with transactional data

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProdID</td>
<td>PG_ID</td>
<td>CURRENCY</td>
<td>PRICE</td>
<td>Description</td>
</tr>
<tr>
<td>PDS01</td>
<td>DS10</td>
<td>USD</td>
<td>549.99</td>
<td>Notebook Speedy I</td>
</tr>
<tr>
<td>PDS02</td>
<td>DS10</td>
<td>USD</td>
<td>650.99</td>
<td>Notebook Speedy II</td>
</tr>
<tr>
<td>PDS03</td>
<td>DS10</td>
<td>USD</td>
<td>600.99</td>
<td>PC Thinkbox I</td>
</tr>
<tr>
<td>PDS04</td>
<td>DS20</td>
<td>USD</td>
<td>8.88</td>
<td>USB Adaptor</td>
</tr>
<tr>
<td>PDS05</td>
<td>DS20</td>
<td>USD</td>
<td>234.99</td>
<td>iPhones PX2</td>
</tr>
<tr>
<td>PDS06</td>
<td>DS20</td>
<td>USD</td>
<td>39.98</td>
<td>Stereo Kit</td>
</tr>
<tr>
<td>PDS07</td>
<td>DS20</td>
<td>USD</td>
<td>15.99</td>
<td>Camera Connector</td>
</tr>
<tr>
<td>PDS08</td>
<td>DS30</td>
<td>USD</td>
<td>22.99</td>
<td>USB Storage</td>
</tr>
</tbody>
</table>

Analysis Process components
What's New with SAP NetWeaver BW 7.30 and BW Accelerator 7.20?

1. Source

2. Mapping

3. Target
For the mapping step an additional support has been provided. In case the connection between source and target has not yet been drawn, the system makes a proposal.

When you click on ‘create Analytic Index’ a pop-up appears which proposes the sources of which the structure could be taken over to create the Analytic Index. In case a source has been chosen here, a connection is drawn automatically from the node to the data target (Analytic Index) and the fields are proposed in the dimension and key figure tab strip.

**How can the Analytic Index be related to an existing InfoObject?**

The system references BW characteristics and uses the properties of these characteristics (‘Referenced InfoObject’ under ‘dimensions’) for analysis with the ABAP Analytic Engine (OLAP processor). This means that display attributes, texts, hierarchies and analysis authorizations of the InfoObjects can be re-used.

Important note: If a dimension of an Analytic Index references to an existing InfoObject and the Analytic Index contains values that do not exist for the InfoObject, new SIDs are generated at Query runtime and written to the master data tables. If the Analytic Index contains values that do not match the InfoObject definition, the Queries abort at the time of the SID generation.

BW key figures can also be referenced, for example, if the number of decimal places of the referenced key figure should be used. If the referenced key figure is of type ‘amount’, a currency dimension has to be assigned.
How can local authorizations be loaded for an Analytic Index?

If analysis authorizations are needed for characteristics without InfoObject reference, it is possible to load authorization data into an Analytic Index. These local analysis authorizations are evaluated during query execution like the standard authorizations.

For authorization-relevant dimensions that do not reference characteristics, the authorizations can be loaded by filling a structure in the same way as for analysis authorizations with the following fields: Attribute Name (ATTR_NAME), User (USERNM), Authorization (Technical Name) (AUTH), InfoObject (IOBJNM), Sign (SIGN), Selection Operator (OPTION), Selection Value (Lower Limit) (LOW) and Selection Value (Upper Limit) (HIGH). Authorization data can be extracted from an InfoProvider. Furthermore users and values can be loaded from an Excel file using an analysis process.
What do you need to remember when loading Excel Files?

**Example for Excel Conversions**

<table>
<thead>
<tr>
<th>Date</th>
<th>Product</th>
<th>Sales Doc</th>
<th>Channel</th>
<th>Quantity</th>
<th>Unit</th>
<th>Revenue</th>
<th>Currency</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.04.04</td>
<td>7389</td>
<td>3</td>
<td></td>
<td>99</td>
<td>EA</td>
<td>1519.00</td>
<td>USD</td>
</tr>
<tr>
<td>21.04.04</td>
<td>7389</td>
<td>3</td>
<td></td>
<td>113</td>
<td>EA</td>
<td>2643.85</td>
<td>USD</td>
</tr>
<tr>
<td>21.04.04</td>
<td>7473</td>
<td>2</td>
<td></td>
<td>55</td>
<td>EA</td>
<td>3580.45</td>
<td>USD</td>
</tr>
<tr>
<td>21.04.04</td>
<td>7472</td>
<td>1</td>
<td></td>
<td>110</td>
<td>EA</td>
<td>6048.90</td>
<td>USD</td>
</tr>
<tr>
<td>26.04.04</td>
<td>7472</td>
<td>1</td>
<td></td>
<td>43</td>
<td>EA</td>
<td>2793.57</td>
<td>USD</td>
</tr>
<tr>
<td>26.04.04</td>
<td>7472</td>
<td>1</td>
<td></td>
<td>75</td>
<td>EA</td>
<td>4307.20</td>
<td>USD</td>
</tr>
<tr>
<td>26.04.04</td>
<td>7472</td>
<td>3</td>
<td></td>
<td>282</td>
<td>EA</td>
<td>5010.32</td>
<td>USD</td>
</tr>
<tr>
<td>26.04.04</td>
<td>7472</td>
<td>3</td>
<td></td>
<td>120</td>
<td>EA</td>
<td>4797.6</td>
<td>USD</td>
</tr>
<tr>
<td>26.04.04</td>
<td>7473</td>
<td>4</td>
<td></td>
<td>55</td>
<td>EA</td>
<td>3024.95</td>
<td>USD</td>
</tr>
<tr>
<td>26.04.04</td>
<td>6943</td>
<td>2</td>
<td></td>
<td>30</td>
<td>EA</td>
<td>19529.7</td>
<td>USD</td>
</tr>
<tr>
<td>26.04.04</td>
<td>6877</td>
<td>1</td>
<td></td>
<td>800</td>
<td>EA</td>
<td>13792</td>
<td>USD</td>
</tr>
<tr>
<td>02.04.04</td>
<td>6877</td>
<td>1</td>
<td></td>
<td>250</td>
<td>EA</td>
<td>5747.3</td>
<td>USD</td>
</tr>
<tr>
<td>02.02.04</td>
<td>6912</td>
<td>1</td>
<td></td>
<td>250</td>
<td>EA</td>
<td>150247.5</td>
<td>USD</td>
</tr>
<tr>
<td>02.02.04</td>
<td>6900</td>
<td>2</td>
<td></td>
<td>70</td>
<td>EA</td>
<td>621.0</td>
<td>USD</td>
</tr>
<tr>
<td>02.02.04</td>
<td>6899</td>
<td>1</td>
<td></td>
<td>12</td>
<td>EA</td>
<td>6595.89</td>
<td>USD</td>
</tr>
<tr>
<td>02.02.04</td>
<td>6899</td>
<td>4</td>
<td></td>
<td>55</td>
<td>EA</td>
<td>488.4</td>
<td>USD</td>
</tr>
<tr>
<td>02.02.04</td>
<td>7479</td>
<td>1</td>
<td></td>
<td>80</td>
<td>EA</td>
<td>710.4</td>
<td>USD</td>
</tr>
<tr>
<td>02.02.04</td>
<td>7479</td>
<td>1</td>
<td></td>
<td>100</td>
<td>EA</td>
<td>7049.7</td>
<td>USD</td>
</tr>
<tr>
<td>02.02.04</td>
<td>6809</td>
<td>3</td>
<td></td>
<td>1000</td>
<td>EA</td>
<td>23499.0</td>
<td>USD</td>
</tr>
<tr>
<td>21.03.04</td>
<td>6809</td>
<td>3</td>
<td></td>
<td>180</td>
<td>EA</td>
<td>5137.4</td>
<td>USD</td>
</tr>
<tr>
<td>21.03.04</td>
<td>6731</td>
<td>1</td>
<td></td>
<td>13</td>
<td>EA</td>
<td>9016.89</td>
<td>USD</td>
</tr>
<tr>
<td>21.03.04</td>
<td>6631</td>
<td>1</td>
<td></td>
<td>500</td>
<td>EA</td>
<td>4440</td>
<td>USD</td>
</tr>
<tr>
<td>21.03.04</td>
<td>6630</td>
<td>1</td>
<td></td>
<td>125</td>
<td>EA</td>
<td>7513.75</td>
<td>USD</td>
</tr>
<tr>
<td>21.05.04</td>
<td>6630</td>
<td>1</td>
<td></td>
<td>400</td>
<td>EA</td>
<td>10592</td>
<td>USD</td>
</tr>
<tr>
<td>20.05.04</td>
<td>6841</td>
<td>1</td>
<td></td>
<td>53</td>
<td>EA</td>
<td>4790.56</td>
<td>USD</td>
</tr>
<tr>
<td>20.05.04</td>
<td>6841</td>
<td>1</td>
<td></td>
<td>350</td>
<td>EA</td>
<td>82246.5</td>
<td>USD</td>
</tr>
<tr>
<td>20.03.04</td>
<td>6854</td>
<td>1</td>
<td></td>
<td>675</td>
<td>EA</td>
<td>10723.25</td>
<td>USD</td>
</tr>
<tr>
<td>20.03.04</td>
<td>6854</td>
<td>1</td>
<td></td>
<td>1090</td>
<td>EA</td>
<td>24134.9</td>
<td>USD</td>
</tr>
<tr>
<td>10.04.04</td>
<td>5431</td>
<td>3</td>
<td></td>
<td>120</td>
<td>EA</td>
<td>78118.0</td>
<td>USD</td>
</tr>
<tr>
<td>10.04.04</td>
<td>5431</td>
<td>3</td>
<td></td>
<td>700</td>
<td>EA</td>
<td>8216.9</td>
<td>USD</td>
</tr>
<tr>
<td>10.04.04</td>
<td>5431</td>
<td>3</td>
<td></td>
<td>425</td>
<td>EA</td>
<td>6795.75</td>
<td>USD</td>
</tr>
<tr>
<td>10.04.04</td>
<td>5073</td>
<td>1</td>
<td></td>
<td>300</td>
<td>EA</td>
<td>11984</td>
<td>USD</td>
</tr>
</tbody>
</table>

The columns for ‘Product’, ‘Sales Doc’ and ‘Channel’ have the default format ‘general’ in the Excel spreadsheet. The system does not recognize these columns as characteristics (as they are not explicitly defined as type ‘text’ e.g.). Therefore, when the file is mapped to a new ‘Analytic Index’ target in APD, the following assignment of fields to dimensions and key figures is suggested:
You can remove ‘Product’, ‘Sales Doc’ and ‘Channel’ from the key figures and create these fields as dimensions, but this might lead to conversion issues. For example, ‘Product 10’ is loaded with value '10.0000000000000000000000' in BWA, which will be displayed in the queries. If the reference for dimension ‘Product’ is set to an InfoObject of type NUMC with length 5 (after loading), then a query on this field aborts during query execution (InfoObject ‘Product’ in the SAP NetWeaver Business Warehouse has data type ‘NUMC’ with length ‘5’, for example).

Therefore, it is strongly recommended that you specify the correct cell format definition in Excel or use an InfoObject that matches as reference. For example, if the column ‘Product’ is formatted as ‘text’ in Excel, it is correctly suggested as a dimension with data type ‘string’ in the definition of the Analytic Index, and it will be loaded with the values as they are displayed in Excel.
Administration

The user interface for the administration of Analytic Indexes RSDD_LTIP shows an overview of all Analytic Indexes and the respective InfoProvider in the ‘@3’ namespace with statistical information. The new ‘@3’ namespace prevents naming conflicts and shows the difference between ad-hoc and standard scenarios. A where-used list shows in which Analysis Processes, CompositeProviders and/or queries the selected Analytic Index is used.
What are the current restrictions, prerequisites and recommendations for the deployment?

Analytic Indexes are not integrated in the BW repository and therefore cannot be transported. As ad-hoc scenarios are mainly created in the productive system and are considered to have a medium life span.

The data saved in an Analytic Index resides exclusively in the BWA as a primary persistency. There is currently no option to store data on the database and no restore mechanism is available. Therefore, the original source data has to be stored somewhere else.

After an Analytic Index has been created, its structure cannot be changed anymore. It is also not possible to add any new dimensions or attributes. However, a copy function is available, which can be used as workaround (copy the index metadata to a second version and enhance/change it before activation, delete the old index and copy back to the old name again).

The following objects can be flexibly modified:

<table>
<thead>
<tr>
<th>Change Analytical Index</th>
<th>TREX Administration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytic Index</td>
<td>Saved</td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Key At</th>
<th>Test</th>
<th>InfoObject</th>
<th>Referenced InfoObject</th>
<th>Authorization</th>
<th>Derived InfoObject</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Date</td>
<td></td>
<td>Date</td>
<td></td>
<td></td>
<td></td>
<td>@03B_A_TUTORIAL@F1</td>
</tr>
<tr>
<td>b</td>
<td>Character</td>
<td></td>
<td></td>
<td>Product</td>
<td></td>
<td></td>
<td>@03B_A_TUTORIAL@F2</td>
</tr>
<tr>
<td>c</td>
<td>Character</td>
<td></td>
<td></td>
<td>Sales Document</td>
<td></td>
<td></td>
<td>@03B_A_TUTORIAL@F3</td>
</tr>
<tr>
<td>d</td>
<td>Character</td>
<td></td>
<td></td>
<td>Channel</td>
<td></td>
<td></td>
<td>@03B_A_TUTORIAL@F4</td>
</tr>
<tr>
<td>e</td>
<td>Character</td>
<td></td>
<td></td>
<td>Unit</td>
<td></td>
<td></td>
<td>@03B_A_TUTORIAL@F5</td>
</tr>
<tr>
<td>f</td>
<td>Character</td>
<td></td>
<td></td>
<td>Currency</td>
<td></td>
<td></td>
<td>@03B_A_TUTORIAL@F6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Change Analytical Index</th>
<th>TREX Administration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytic Index</td>
<td>Saved</td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Ty, Test</th>
<th>InfoObject</th>
<th>Referenced InfoObject</th>
<th>Unit Dimension</th>
<th>Derived InfoObject</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Floating Poi</td>
<td>Quantity</td>
<td></td>
<td></td>
<td>@03B_A_TUTORIAL@F7</td>
</tr>
<tr>
<td>b</td>
<td>Floating Poi</td>
<td>Revenue</td>
<td></td>
<td></td>
<td>@03B_A_TUTORIAL@F8</td>
</tr>
</tbody>
</table>

It is not possible to create an Analytic Index if the SAP BW Accelerator is not available. If an Analytic Index is corrupt, the query will fail. To recover from this situation, the index has to be activated/saved and loaded again.

An Analytic Index cannot be used as a PartProvider in a MultiProvider.

If consolidated data is needed, it should be staged with data warehousing mechanisms including error handling and the APD/AI scenario should not be used. The APD is not a warehouse management staging tool (providing proper delta, consolidated data, quality), but uses the resulting data from the staging processes to execute Analysis operations.
To load higher data volumes, use a query with the property flag ‘Query is used as InfoProvider’ as a source of type ‘InfoProvider’ in APD for automatic package-by-package reading (and if OLAP features are needed that are already contained in the query).

To use an Analytic Index, you need an SAP BW Accelerator and the person who creates it in the APD must have access to the BW Backend. The APD itself is more a tool for the technical business analyst working in IT than in a business department.

**How is the Analytic Index displayed in the SAP Business Explorer Query Designer or in transaction RSRT?**

**a)** Flat File Example **without** InfoObject Reference in the Business Explorer Query Designer

The InfoProvider @3SB_AI_TUTORIAL can be found via the search functionality in the Query Designer and can be selected here to create a query:

![Query Definition - @3SB_AI_TUTORIAL](image)

**Query Definition - @3SB_AI_TUTORIAL:**
As ‘product group’ has been flagged as a ‘navigation attribute’ when defining the Analytic Index, this characteristic can be used for navigation.

Select attributes (text and price) of characteristic ‘product’ in the query using the context menu: ‘properties → attributes’:

Query result:
In addition, the default query `@3SB_AI_TUTORIAL` is available for reporting:

![Query result](image)| Analytic Index from Flat File
---|---
| Author | Status of Data | 95.06.2010 02:08:15
| Chart | Filter | Information

<table>
<thead>
<tr>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity</td>
</tr>
<tr>
<td>13 008,000,000.00</td>
</tr>
</tbody>
</table>
a) Flat File Example with InfoObject Reference executed in transaction RSRT (default query)

Query result:

Choosing the display attributes ‘Product Group’ and ‘Price’

Also the hierarchies on InfoObject ‘0VC_PROD1’ are available for reporting.
How is the data displayed in a query when local analysis authorizations are used?

a) Flat File Example with InfoObject Reference executed in transaction RSRT (default query)
In the definition of the Analytic Index, the flag for ‘Authorization’ has to be set for field b = ‘Product’.

b) Executing the default query in RSRT2 – Comparison of User with Full Rights and User SB_APD_USER’
For more information, see the documentation about the ‘Analytic Index’ under
5.6. CompositeProvider (Updated 1.2)

What is a CompositeProvider?
A CompositeProvider is a data model where ‘1:n’ Analytic Indexes created with the Analysis Process Designer are combined via UNION and inner or left outer JOINS. It is a semantic definition without data persistency. Joins are calculated on an ad-hoc basis at query runtime. It is the first object in the Business Warehouse where data can really be joined in the SAP BW Accelerator (an InfoSet is a DB-JOIN and a MultiProvider is a UNION).

The transient Metadata framework is the interface between the standard BW and the CompositeProvider Provider Framework. Via the transient metadata it is possible for the CompositeProvider to provide own metadata, which can but need not reference original BW-Metadata, e.g. InfoObjects. Prerequisite for using a CompositeProvider is the SAP Business Warehouse Accelerator (BWA). The runtime object for a CompositeProvider inside the BWA is a CalculationScenario. It is utilized during querying the data via the DataManager.

The metadata model of the CompositeProvider for the SAP BW Accelerator (BWA) calculation engine is described in the CalculationScenario. It contains the definition of the UNION and JOIN operations between the Analytic Indexes as well as the list of columns (view attributes/key figures), which are available in the CompositeProvider.

This CalculationScenario is used to generate the runtime InfoProvider in the ‘@3’ namespace, which is exposed to all BI clients in the same way as any other standard InfoProvider via the Business Explorer Query Designer.

Transient Metadata Framework
What is for the purpose of a CompositeProvider?

The CompositeProvider Modeling UI is a modeling environment which enables the IT department to perform rapid prototyping and easy modeling of ad-hoc scenarios with SAP BW Accelerator (BWA) in the data warehouse.

Data can be supplied by more than one source and a combination based on field names (no keys, no temporal joins) is needed. Examples: merging two query results, combining BW data and flat files or joining master data outside the InfoCube model inside the SAP BW Accelerator.

The creation of a CompositeProvider offers a real new modeling perspective as you can combine stable enterprise scenarios with temporary requirements in the context of BW workspaces (see corresponding chapter). Here the CompositeProvider shows its full potential and is suited to the business analyst in the business department.

How to create a CompositeProvider

From APD process to query
The CompositeProvider Modeling UI as Maintenance User Interface

The user interface for the graphical modeling of a CompositeProvider is the SAP GUI CompositeProvider Modeling UI (Transaction RSLIMO).

The first step when creating a CompositeProvider is to assign a technical name. The maintenance screen opens with an empty CompositeProvider object in the middle of the screen. From the InfoArea tree, ‘n’ Analytic Indexes can be selected and combined with different join operations. When you select and apply an Analytic Index, the join type has to be specified in advance. This type can be changed later on. At least one Analytic Index is required as a UNION operation.

Choose the relevant fields for the CompositeProvider by drag and drop. A connection from the source provider to the CompositeProvider is drawn for each field. Joins on same field level are also created by drawing a line between the fields to be connected (line showing ‘Join’ as binding type; clicking on a field e.g. in the CompositeProvider and choosing right mouse-click ‘show connection’ displays all connections to the providers in case they are connected via this field). Here you can choose between inner or left outer JOIN. You can switch between technical names and field descriptions by choosing ‘show technical name’. You can also change the technical name and description by choosing ‘properties’. With push button ‘property’ a kind of ‘mass’ maintenance for technical names and descriptions is possible.

You can use one Provider several times as a reference in a CompositeProvider when binding type ‘Join’ is used. Each Provider receives an Alias name which has to be unique but can also be changed later.
In a CompositeProvider, referenced InfoObjects can be used to display attributes or use hierarchies, as is the case with Analytic Indexes. Assigning different referenced InfoObjects to the same column in the CompositeProvider means that none of the display attributes can be selected in a query.

When the CompositeProvider is saved, its definition is saved in the Business Warehouse. When it is activated, it is converted into a CalculationScenario and created in the SAP BW Accelerator. There is an option to view the XML which describes the data model.

**Definition of a CompositeProvider**

```xml
<complexType name="SB_FLEX">
    <complexContent>
        <extension base="t4:CompositeColumn">
            <sequence>
                <element name="Col year / month" type="t4:AnalyticIndex"/>
                <element name="Calendar year" type="t4:AnalyticIndex"/>
                <element name="Distribution Channel" type="t4:AnalyticIndex"/>
                <element name="Customer" type="t4:AnalyticIndex"/>
                <element name="Product group" type="t4:AnalyticIndex"/>
                <element name="Product" type="t4:AnalyticIndex"/>
                <element name="MYCONTACT" type="t4:AnalyticIndex"/>
                <element name="NC_CITY" type="t4:AnalyticIndex"/>
                <element name="Bill Quantity" type="t4:AnalyticIndex"/>
                <element name="REORDER" type="t4:AnalyticIndex"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

**XML of a CompositeProvider**
The data of a CompositeProvider can also be displayed in the SAP GUI Modeler:

**Display CompositeProvider SB_FLEX**

- Description
- Display Data (Ctrl+Shift+F1)
- infoProvider @3DB_FLEX

### Data of a CompositeProvider

A where-used list of the Analytic Indexes used in CompositeProviders is available in transaction RSDD_LTIP.

### What is a union, an inner or left outer join?

An **inner join** is the most common join operation used in applications and can be regarded as the default join-type. Inner join creates a new result table by combining column values of two tables (A and B) based upon the join-predicate. The query compares each row of A with each row of B to find all pairs of rows which satisfy the join-predicate. When the join-predicate is satisfied, column values for each matched pair of rows of A and B are combined into a result row.

The result of a **left outer join** (or simply left join) for table A and B always contains all records of the ‘left’ table (A), even if the join-condition does not find any matching record in the ‘right’ table (B). This means that if the ON clause matches 0 (zero) records in B, the join will still return a row in the result—but with NULL in each column from B. This means that a left outer join returns all the values from the left table, plus matched values from the right table (or NULL in case of no matching join predicate).

If the left table returns one row and the right table returns more than one matching row for it, the values in the left table will be repeated for each distinct row on the right table.

The result of a left outer join (or simply left join) for table A and B always contains all records of the ‘left’ table (A), even if the join-condition does not find any matching record in the ‘right’ table (B). This means that if the ON clause matches 0 (zero) records in B, the join will still return a row in the result—but with NULL in each column from B. This means that a left outer join returns all the values from the left table, plus matched values from the right table (or NULL in case of no matching join predicate).

If the left table returns one row and the right table returns more than one matching row for it, the values in the left table will be repeated for each distinct row on the right table.

In the CompositeProvider use case, the left table is the CompositeProvider (first UNION operation that has been executed as a basis for the CompositeProvider). See the following example:

http://en.wikipedia.org/wiki/Join_(SQL); timestamp 20101115090238
Example

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORDER</td>
<td>CUSTOMER</td>
<td>WEIGHT</td>
</tr>
<tr>
<td>99</td>
<td>1000</td>
<td>10</td>
</tr>
<tr>
<td>100</td>
<td>501</td>
<td>20</td>
</tr>
<tr>
<td>101</td>
<td>502</td>
<td>30</td>
</tr>
<tr>
<td>102</td>
<td>503</td>
<td>40</td>
</tr>
<tr>
<td>103</td>
<td>504</td>
<td>50</td>
</tr>
</tbody>
</table>

Order table (left table)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORDER</td>
<td>INVOICE</td>
<td>AMOUNT</td>
</tr>
<tr>
<td>100</td>
<td>101</td>
<td>200</td>
</tr>
<tr>
<td>101</td>
<td>502</td>
<td>300</td>
</tr>
<tr>
<td>102</td>
<td>503</td>
<td>350</td>
</tr>
<tr>
<td>104</td>
<td>505</td>
<td>400</td>
</tr>
</tbody>
</table>

Invoice table (right table)

Definition of a CompositeProvider with Inner Join

<table>
<thead>
<tr>
<th>ORDER</th>
<th>CUSTOMER</th>
<th>INVOICE</th>
<th>WEIGHT</th>
<th>AMOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1000</td>
<td>501</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td>101</td>
<td>1001</td>
<td>502</td>
<td>30</td>
<td>200</td>
</tr>
<tr>
<td>102</td>
<td>1001</td>
<td>503</td>
<td>40</td>
<td>300</td>
</tr>
<tr>
<td>102</td>
<td>1001</td>
<td>504</td>
<td>40</td>
<td>350</td>
</tr>
</tbody>
</table>

Result Inner Join

Definition of a CompositeProvider with Left Outer Join
Result Left Outer Join

A union operation is used to combine data in a MultiProvider. The system constructs the union set of the relevant data sets; all the values of these data sets are combined. As a comparison: InfoSets (and joins in CompositeProviders) are created using joins. These joins only combine values that appear in both tables. In contrast to a union, joins form the intersection of the tables.

What do you have to remember when defining joins in a CompositeProvider?

When creating a CompositeProvider with join type JOIN, you can choose the option ‘Unique JOIN Columns’. This option is only available in transaction RSLIMO with BW 7.30 SP01 and should be used carefully. In other words, it can be used because the expected drill-downs will not result in incorrect data due to filters, for example, or due to a really unique JOIN condition. It can be set with the Analytic Index used in the CompositeProvider; right mouse click and select ‘unique JOIN condition’.

If this option is not set, the JOIN column is added to the GROUP BY automatically by the BW OLAP engine, in order to calculate a correct sum for the key figures from different sources. It ensures consistent data at the cost of the performance, because the result set read from the BWA is increased and a post-aggregation step is performed on the BW application server and not on the BWA.

If the option is set, aggregation is carried out before the JOIN is executed, which could lead to incorrect results if the JOIN is not really unique.

Example without option ‘unique JOIN condition’

Definition of a CompositeProvider without option ‘Unique JOIN Columns’
Execution of the default query with debug option ‘display SQL/BWA statement’

TREX Join Index: OBW_RST:BIY:SB_TEST01 OBW_RST:BIY:SB_TEST01

Requested Attributes (SELECT)

a INTO k__2720  \rightarrow a = order

Key Figures
__muaofeaktasloruw Aggregation B2 INTO Z__873
c Aggregation B1 INTO Z__2731
f002 Aggregation B1 INTO Z__2732

Constraints
Name: SSKEYDATEIS Value: 20100302

Join Conditions (FROM)

Query Entries (WHERE)

Join column ‘order’ in GROUP BY (requested attributes)
Result of default query

```

Requested Attributes (SELECT)

a INTO K___2728
f001 INTO K___2730
a = order
f001 = invoice

Key Figures
__numoffacttablervos Aggregation 62 INTO Z__073
__numoffacttablervos Aggregation 61 INTO Z__2731
f002 Aggregation 61 INTO Z__2732

Constraints
Name: $$KEYDATE$$ Value: 20100902

Join Conditions (FROM)

Query Entries (WHERE)
```

**BWA/SQL statement for drill-down on invoice**
Drill-down on invoice

**Query**

- **Initial List**
- **Bookmarks**
- **Process**

**Bookmark (Data)**

```
PAS END=1&REQUEST_NO=1&DATA_PROVIDER=DP&CMD=EXPAND&OBJ_NM=%93SB_TEST93%0INV
```

Create bookmark to refresh query

- **TREX Join Index:** OBB:RST:BIY:SB_TEST01.OBB:RST:BIY:SB_TEST01

Requested Attributes (SELECT)

- `a` INTO K____2728
- `f001` INTO K____2730

Key Figures
- `__numoffacttablerows` Aggregation 02 INTO Z____073
- `c` Aggregation 01 INTO Z____2731
- `f092` Aggregation 01 INTO Z____2732

Constraints
- Name: `$$KEYDATE$$` Value: 20100902

Join Conditions (FROM)

Query Entries (WHERE)

**BWA/SQL statement for bookmark**

```
BEx Ad Hoc Analysis
```

**Bookmark (without cache) with correct result**
Example with option 'unique JOIN condition'

Definition of a CompositeProvider with option 'unique JOIN condition'

Execution of the default query with debug option 'display SQL/BWA statement'
What's New with SAP NetWeaver BW 7.30 and BW Accelerator 7.20?

No column in GROUP BY (requested attributes)

Correct result as ‘aggregation’ has been executed before the JOIN

BWA/SQL statement for drill-down on invoice
Drill-down on invoice – result line is correct **only because** it is **cached** in the OLAP processor

```
BWA/SQL statement for bookmark

TREX Join Index: OBV:RST:BIY:SB_TEST03

Requested Attribute (SELECT)  
  f001 INTO K__2735

Key Figures
  __numoffacttablerows Aggregate 02 INTO Z__073
  __numofrow Aggregate 01 INTO Z__2738
  f002 Aggregate 01 INTO Z__2737

Constraints
  Name: $$KEYDATE$$ Value: 20100902

Join Conditions (FROM)

Query Entries (WHERE)

```
Bookmark (without cache) with incorrect result (weight ‘40’ assigned to invoice 503 and 504 both result from order 102 and therefore must not be aggregated)

If you want to use navigational attributes in a CompositeProvider you have to include the navigational attributes as fields. If you just add the characteristic, its attributes will be ‘display-only’.

If you have an AnalyticIndex and reference one of its fields to an InfoObject with attributes, those attributes will be display-only as well. You need to create a CompositeProvider and map the MasterdataProvider of the InfoObject to the Analytic Index to be able to navigate on the attributes.

**How are authorizations handled in a CompositeProvider?**

In general, users reporting on a CompositeProvider can only view data for which they are authorized, thanks to the SAP NetWeaver Business Warehouse Analysis Authorization concept. Analysis authorizations loaded via the Analysis Process Designer to an Analytic Index, which is combined in a CompositeProvider, are also taken into consideration. Hierarchy node authorizations are not taken into account for Analytic Indexes. This also applies to a CompositeProvider where these Analytic Indexes have been combined.

The user might see less data when exclusively reporting on a PartProvider than when using the same PartProvider in a CompositeProvider. This is because an intersection of the authorizations is built for the CompositeProvider.

If the user is not authorized for all PartProviders contained in the CompositeProvider, the user will have no authorizations to see data.

If the user is authorized for all PartProviders but not all authorization-relevant InfoObjects are mapped, the user will only see the data he is authorized for, if the minimum aggregation authorization ‘:’ is defined for the non-mapped InfoObjects.
Finally, if the user is authorized for all PartProviders and all authorization-relevant InfoObjects are mapped or the minimum aggregation authorization is defined for the non-mapped InfoObjects, but the user has different authorizations for each PartProvider, the intersection of these authorizations is built. See the following example:

![Diagram showing intersection of analysis authorizations]

**Intersection of analysis authorizations**

The corresponding message on missing authorizations (in case providers are authorized but no InfoObjects) is displayed immediately in the CompositeProvider use case whereas the original providers would show the drill-down status but no data with the corresponding message.

**What are the current restrictions, prerequisites and recommendations for the deployment?**

To be able to create a CompositeProvider, you need an SAP BW Accelerator and the person who creates the CompositeProvider in transaction RSLIMO must have access to the BW Backend. The CompositeProvider Modeling Maintenance UI is a tool aimed more at the technical business analyst working in the IT department than in the business department. Currently a CompositeProvider cannot be transported as in the RSLIMO use case the main focus is on rapid prototyping whereas in the BW Workspaces context the focus is on flexibility for the business user.

For more information on using a CompositeProvider in the context of BW Workspaces where the business user creates models, see chapter ‘BW Workspaces’. Here the CompositeProvider shows its full potential and is suited to the business analyst in the business department.

It is not possible to create temporal joins with a CompositeProvider like with InfoSets (InfoSets are currently not available in the SAP BW Accelerator).
How is the CompositeProvider displayed in the Business Explorer Query Designer?

Scenario example

Source 1: A query on a MultiProvider contains information on ‘lost deals’ per customer.

Query as a source in the Analysis Process Designer

Source 2: An individual flat file contains information on the customer ABC ranking as well as contact names and address data (e.g. city).

Excel File as a source in the Analysis Process Designer
**Materialized Join**

In the Analysis Process Designer, a join could be materialized by using the connection type ‘join’ to join the data from the query with the data from the Excel file:

**Definition of the Analysis Process with connection type ‘join’**

**Definition of a query on InfoProvider @3SB_LOST_MAT**
### Query Result

<table>
<thead>
<tr>
<th>Table</th>
<th>Column1</th>
<th>Column2</th>
<th>Column3</th>
<th>Column4</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column5</th>
<th>Column6</th>
<th>Column7</th>
<th>Column8</th>
<th>Column9</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Flexible Join
A new approach could be to flexibly join the Analytic Indexes by using a CompositeProvider:

**Definition of the Analysis Process for the query data**

**Definition of the Analysis Process for the Excel data**
What's New with SAP NetWeaver BW 7.30 and BW Accelerator 7.20?

Definition of the CompositeProvider

Definition of the query
What's New with SAP NetWeaver BW 7.30 and BW Accelerator 7.20?

For more information, see the documentation about ‘Creating CompositeProviders’ under
http://help.sap.com/saphelp_nw73/helpdata/en/c8/0a4212d4824010adb3c727023d7c77/frameset.htm

6. Flexibility – What’s Next? (Updated 1.3)

With the functionality of Analytic Indexes and CompositeProviders, BW7.30 already has the tools in place to easily create ad-hoc models and data sets. The challenge is now to enable a power user to use this functionality in an end-user-friendly but also controlled manner. This is exactly the goal of the BW Workspaces, a feature released for BW 7.30 SP05. In the following chapters we give an overview on the BW Workspaces.

6.1. Motivation

BW Workspaces address a specific gap often encountered in data warehousing implementation projects; the conflict between the central IT department and the business department requirements.

IT has to maintain a centrally governed space with data models (central data with centrally defined semantics on that data), is bound by service level agreements as well as by compliance rules and aims to keep the data in the data containers consistent. This means that IT cannot react spontaneously to requests to change data or to add new data.

The business department in contrast acts in a more agile way in its day-to-day business. For example, a marketing department meets today and wants to initiate a marketing campaign in a few weeks time. In order to monitor the data for the campaign, the marketing department wants to set up an InfoCube to track customer reactions, for example. The business department approaches the IT department with this request and is told that a new InfoCube could be deployed in two months time at the earliest. This is the conflict that we mean.

---

Architected versus departmental data marts
6.2. BW Workspaces

This conflict can be solved by setting up a BW Workspace for the department (e.g. marketing, human resources, controlling etc.) in the SAP NetWeaver Business Warehouse that can react quickly to new and changing requirements.

A BW Workspace is a kind of ‘small sandbox’ which IT can define. It sets the boundaries, it sets the amount of resources that a BW Workspace can consume and it exposes some of the central data models to the BW Workspace (data of the models and their related semantics). It exposes them in a logical fashion only. The data is not copied over into the BW Workspace, which means that no data replication is needed.

The goal is to enable the Key Business Users to use this functionality in a dedicated and separated environment, which is deeply embedded and integrated in the existing BW landscape, and bridges the gap between the architected and the departmental data marts. Therefore, workspaces are integrated, independent and, as the usage of the SAP Business Warehouse Accelerator is mandatory, they are completely in-memory.

What happens afterwards, inside the BW Workspace, is in the hands of the Business department. However, BW Workspaces is important functionality for the Key Business User who knows which data he/she wants to combine and what to achieve with the joined data. Therefore, the Business department also needs a basic understanding on the common join-types like UNION, INNER or LEFT OUTER JOIN.
6.3. Tools

Workspaces Maintenance
In the workspace maintenance, the Business Warehouse Administrator can create workspaces in a folder structure, define the limits for each workspace and assign data (central BWA indexes or Analytic Indexes in the Business Warehouse) to the workspace. It is the central maintenance tool for workspaces for the IT department.

Workspace Designer
The BW Workspace Designer is a tool that allows the business department to work with the BW Workspaces. In this design environment, the Key Business User can upload private data, e.g. from a file (CSV or Excel), a query result or data from a BW DataSource. These locally uploaded models can be easily combined with centrally exposed models in a CompositeProvider via UNION and JOIN combinations.

The BW Workspace Designer runs in a browser and can also be embedded into the SAP Portal. This means that there is no need to install extra software on the office PC. You simply need to start the URL in the browser.

Detailed articles on the functionality of the BW Workspace with comprehensive examples and screenshots can be found on SDN see:

7. Others

7.1. RSDDB - BWA Maintenance UI (Updated 1.2)

The appearance of the UI maintenance screen (for creating and deleting BWA indexes) has changed in BW7.30. In the old wizard in BW7.0, only a few simple steps were needed to handle objects in BWA. However, in some cases, it makes sense to get a better overview of the available options for tuning and changing the behavior to suit special cases. We hope that we have achieved this without sacrificing the simplicity of the standard case in the new UI. You access the new maintenance screen using the old options (via the admin workbench or the transaction RSDDV) or by executing transaction RSDDB.

First select the object type and then enter the technical name of the object (or select it in the F4 help). The list below shows all BWA indexes for the selected object type and their status. A double-click on a line opens the display mode for the BWA index. A screen with several tabs appears. Here you can make object specific settings and view information. In most of all cases, you should be able to just ignore the tabs, press the ‘activate and fill’ button and schedule the batch job. Note that activation (creation of indexes) and filling (reading and indexing data) is now performed in one single step - the batch job.

A list of objects which have a BWA index is displayed. It additionally shows a where-use for Explorer, accelerated or the BW Workspaces.

We will now have a brief look at the tabs for BWA indexes for InfoCubes, to cover the remaining cases - or to help users who like to experiment with different options. Each object looks slightly different because they all contain certain specific settings, but once you have understood one, you should be able to understand the others.

This tab contains the standard information for a BWA object and allows you to set a BWA index to ‘inactive’. In addition, you can set the parameter NOBIANOQUERY for each individual object BW7.30 (this is not possible in BW7.0, as the parameter is global). All options for the query behavior when the BWA index does not exist are explained in detail in SAP note 1569658.
For more information, see also the documentation about the ‘Query behavior dependent on the status of the BWA index’ under


The indexing parameters are now also object-specific (in BW7.0 only global settings are possible). If the ‘System Settings’ are used, the global or default values are applied.
Here you can also toggle between the system settings and object-specific values. For example, the expected size is defined using the size of the DB tables by default. However, if you know that this size will increase significantly in a short time, you can set the expected size to a higher value and force a splitting and distribution of the data in BWA according to this size. This means you avoid the need to rebuild or reorganize the data later on.
When BW7.30 is updated to SP5 it is possible to use HANA as a database. In the HANA use case some minor things are different on the maintenance and administration side. It is e.g. not necessary for most of the InfoProviders to be indexed as they are already available in the columnar store of the database. In RSDDB it is still necessary to index snapshots for the VirtualProvider and the QueryProvider.

This is reflected in the online documentation see:
http://help.sap.com/saphelp_nw73/helpdata/de/AF/81F38183CF46B9B919F82D7BB7EC47/frameset.htm

8.1.1 Excursion on topics ‘Partition attributes, index split and BWA landscape reorganization’

In general the BWA function ‘reorganize landscape’ ensures an optimized distribution of split indexes across the BWA landscape. The reorg function can be triggered in transaction RSDDBIAMON with option ‘reorganize BWA landscape’.

In exceptional cases it might nevertheless be reasonable to specify a partition attribute manually in database table ‘rsddtrexadmin’. Please see SAP note 1591892 for details.
In the ‘InfoObjects Options’ Tab, you specify whether F4 value help BWA indexes should be created automatically for certain InfoObjects that are characteristics of the InfoProvider.

The ‘Index Info’ tab shows the relevant tables for this InfoProvider that have to be indexed in BWA, their current status, size on the DB (based on the statistics) and size in BWA.
Please note that all object-specific settings are kept, even if the BWA index is deleted. Therefore, you do not have to maintain them again, after the index has been dropped (for example, during the execution of a process chain). A description of most of the parameters and input fields is available in the F1 help.

The tabs of other objects can vary slightly, depending on the capabilities of the InfoProvider BWA index.

For more information, see the documentation about ‘Indexing BW Data in SAP NetWeaver BW Accelerator’ under

http://help.sap.com/saphelp_nw73/helpdata/de/4c/2c87e2477f51e6e10000000a42189b/frameset.htm

**Delta Index**

In transaction RSDDBIAMON you can set a delta index flag per each physical index. After deleting a BWA index also the settings concerning the delta-index used to be discarded. Since Support Package 03 this is not the case anymore. The settings are kept as any other index setting.

**7.2. Metadata changes and BWA alignment**

Previously, changes to the metadata of objects with a BWA index often meant that the indexes had to be deleted and completely rebuilt, if the structure of the corresponding original DB table had changed. In BW7.30, most common metadata changes no longer require an index rebuild. The changes are made on the index directly, which is generally a much faster process. At the moment, there are the following cases where we cannot directly adjust the index and we have to rebuild it from scratch:

- The P/Q/T – Index must be rebuild if the InfoObject changes from compounded to not-compounded and vice-versa.
- T-Index must be rebuild if time-dependency and/or language-dependency changes
- S-Index must be rebuild if the InfoObject changes from having no display attributes to having any display attribute and vice-versa.
- when the InfoObject ‘AA’, for example, is added as a Display Attribute to InfoObject ‘BB’ and one of the compound parents of InfoObject ‘AA’ is already Display Attribute of InfoObject ‘BB’ (in this case, the field in the P/Q-index must be filled with the ‘compounded’ initial value).

**7.3. Master data handling**

To enable Master Data Provider and F4 value help with BW Accelerator, the BWA master data handling had to be re-designed in BW7.30. In BW7.0, the master data in BWA always depended on an InfoCube with a BWA index that required the master data either as a characteristic or as navigation attribute. In BW7.30, the master data is always part of the BWA index of an InfoObject. There are 3 different types of BWA indexes for InfoObjects:

- IDXT (BWA Index Type) = ‘MDR’ – this is the default type. The BWA index of this InfoObject is only used for reporting – not F4-value help. In BWA, only the SID and X/Y indexes exist (if applicable). They are automatically created for all InfoObjects required by an InfoCube (for example) if it is indexed.
- IDXT (BWA Index Type) = ‘MDF’ – this is the F4 value help type. The BWA index has physical indexes for SID, X/Y, P/Q, and text tables (if applicable) and a logical index for the metadata. If the F4 value help BWA index is created and an ‘MDR’ index already exists, only the missing tables have to be indexed.
- IDXT (BWA Index Type) = ‘MDD’ – this type is created for all attributes of InfoObjects with F4 value help BWA index. These BWA indexes have the same physical indexes as the ‘MDF’, but do not have the logical index on top. This third type was introduced to avoid the situation where too many other
InfoObjects have to be indexed when an F4 value help BWA index for an InfoObject is created. In other words, it is not necessary to index the data for attributes of attributes.

When re-designing the handling, we also addressed the most important issues and challenges in this area: Indexing parallelization for large master data tables, lock & wait issues, size of table rsddtrexnewsid, and the change run attribute.

### 7.4. Indexing Hierarchies (New 1.3)

BW hierarchies are available for indexing in RSDDDB with BW7.30 SP05. Enter the InfoObject name and press ‘create’. The suffix _$HIE is automatically attached.

When this option is chosen, all hierarchies of an InfoObject are indexed.

![BWA Index Maintenance: Hierarchy](image)

The hierarchies defined on an InfoObject must meet the following conditions:

- There must be at least one hierarchy in active version
- Only time-independent hierarchies are allowed
- Time-characteristics are not allowed (0DATE e.g. is available as it is saved internally as CHAR (8))

In general, when a query is executed, the creation of temporary BWA hierarchy indexes does not take much time and also these indexes are reused. Therefore it is recommended to have a look first at the data manager statistics if the execution times of a query are long due to the hierarchy processing (and hierarchies for which InfoObjects are affected) before creating ‘permanent’ hierarchy indexes. If ‘permanent’ BWA hierarchy indexes are existent you have to remember that this has an impact on adjustment processes (e.g. activation of a hierarchy/ hierarchy changerun lead to index adjustment processes).

So when is this option a benefit?

It can be helpful to use this functionality in case many hierarchies have to be created anew.

When the processing time is lost for the roundtrip which is necessary to read the node information table on the database and to join it with the temporary BWA index, it can be a performance benefit when all information is available in the BWA which is the case when the hierarchy is indexed in RSDDDB.

For more information, see the documentation about ‘Indexing BW Data in SAP NetWeaver BW Accelerator’ under

7.5. **Indexing parallelization**

When indexing large master data tables (S (SID), X, Y, P, Q), the indexing process splits the data into disjoint blocks. Each block is read from the DB and indexed in BWA, in a separate background process (the same as with fact tables already in BW7.0). The split threshold and the number of blocks (and also batch processes) depend on the default/system/object settings (see above).

7.6. **Master data delta Daemon**

In BW7.30, new master data is rolled up to BWA (indexing) using a unique process - the master data delta daemon (job name in the job log (SM37) ‘BI_BIA_NEWMD_INDEX’). This process runs in the background and it is scheduled automatically after the first object in BWA is created. The daemon runs every 3 minutes (interval can be changed – see global parameters) and checks the master data queue (table rsddtrexnewsid) for new records and indexes them. Each InfoObject with new master data values for BWA is processed in a separate background process – the number of background processes that can run in parallel is defined by the parameter BATCHPARA (same as for standard BWA indexing).

You can check the application log messages of the master data delta Daemon from RSDDB->Menu: Goto->Application Log->“Load deltas in master data indexes”:

![BWA Index Maintenance: InfoCube](image)
## What's New with SAP NetWeaver BW 7.30 and BW Accelerator 7.20?

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>User</th>
<th>Number</th>
<th>External ID</th>
<th>Object text</th>
<th>Sub-object text</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.08.2010 11:08:11</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19752</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:01:48</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19752</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:04:10</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19764</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:07:16</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19755</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:10:16</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19766</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:13:16</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19757</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:16:10</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19768</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:19:13</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19759</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:22:15</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19770</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:25:17</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19771</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:28:13</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19772</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:30:40</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19773</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:33:56</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19774</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:38:03</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19775</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:41:17</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19777</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:42:05</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19779</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:49:06</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19780</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 13:00:00</td>
<td>SEUBERTA</td>
<td>50</td>
<td>NEWMD_INDEX19785</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
</tbody>
</table>

### Application Log

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>User</th>
<th>Number</th>
<th>External ID</th>
<th>Object text</th>
<th>Sub-object text</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.08.2010 12:04:15</td>
<td>SEUBERTA</td>
<td>66</td>
<td>NEWMD_INDEX19764</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
<tr>
<td>18.08.2010 12:04:15</td>
<td>SEUBERTA</td>
<td>66</td>
<td>NEWMD_INDEX19764</td>
<td>BIA Index</td>
<td>Fill BIA Index</td>
</tr>
</tbody>
</table>

- **BWA master data delta daemon job number 19764**
- **Parallel index filling process in background job TMD_002TXH1807AKWQBBWR0NUT7G**
- **Parallel index filling process in background job TMD_002TXH1807AKWQBBWR0NUT7G**
- **Parallel index filling process in background job TMD_002TXH1807AKWQBBWR0NUT7G**
- **Parallel Prozess Nummer**

> * Batch process with ID TMD_002TXH1807AKWQBBWR0NUT7G number 1 and proce
> * Process running in work process 11 with WP-RID 18014 on server locbr1_BR1_69_1
> * Start function module RSDDTREX_NEWM_INDEX_PARALLEL at (time stamp) 2010
> * Write new characteristic values for ZKNNOVAL' to BWA
> * Delta for BWA index of object ZKNNOVAL' (type 'MDP')
> * Delta for index for table /BIC/ZKNNOVAL'

1. Delta mode for index /BIC/ZKNNOVAL' set to 'Aus' {duration: '0.010016'}
2. '1' data records loaded to index /BIC/BIA/BR1/BIC/ZKNNOVAL'
3. Prepare Optimize for index /BIC/BIA/BR1/BIC/ZKNNOVAL'
4. Commit Optimize for index /BIC/BIA/BR1/BIC/ZKNNOVAL'

- **Delta created for BWA index of object ZKNNOVAL'**

1. **End of processing of function module RSDDTREX_NEWM_INDEX_PARALLEL at (time stamp) 2010**
2. **Parallel Prozess Nummer**
3. **Parallel Prozess Nummer**
7.7. **Lock & wait issues**

In BW7.0, the indexing of new master data records is part of the InfoCube Rollup. This can lead to lock and wait times if two or more InfoCubes with shared master data were rolled up at the same time.

To ensure that transactional data in an InfoCube (that is rolled up to the BWA index) has the corresponding master data records in BWA as well, the InfoCube Rollup needs to make sure that the rolled up InfoCube request is only released for reporting, once the master data daemon has indexed the records of all InfoObjects that are characteristics of the InfoCube. A wait time is still possible, but in general it will be much shorter than in BW7.0. SAP note 1640103 describes the dependencies of changerun, master data daemon and rollup processes and how to handle locking issues.

7.8. **BWA admin parameters**

Over time we have introduced global BWA admin parameters (table rsddtreqadmin) that change the default behavior of your system when reading from or writing to the BW Accelerator. In BW7.30 we have consolidated these parameters, removed the obsolete ones, made some of them object-specific (see above) and built a maintenance UI for setting parameters that have remained global. You access this UI either by executing the program rsddtreq_admin_maintain or via the BWA Monitor (menu: BW Accelerator -> Index settings -> Change global parameters).

The parameters are grouped according to topics in tabs. You can jump directly to an SAP note describing the parameter in more detail (if available). Any new parameters in BW7.30 are all located here and have an up-to-date SAP note for documentation.

For more information, see the documentation about ‘Maintaining Global BWA Index Settings’ under http://help.sap.com/saphelp_nw73/helpdata/en/4b/e2f7e720f91323e10000000a15822b/frameset.htm
7.9. Issues with table RSDDTREXNEWSID

The table RSDDTREXNEWSID is the BWA queue for new characteristic values. If new characteristic values are created in BW (e.g. by master data or transactional data load) and the InfoObject has data in BWA, the SIDs of the records are written to the queue table. In BW7.0, the size of the queue table can increase significantly if one of the following (rare and specific) situations occurs:

1. Master data records were loaded but the Rollup for the InfoCube using this InfoObject was not executed or not executed in time. The SIDs stayed in the queue for a long time.
2. New characteristic values are created in BW without reference to transactional data or with master data (indicated by empty *FL fields in the SID table). These SIDs stayed in the queue until either the master data or transactional data was loaded using this characteristic value.

Issue 1 is solved by the master data daemon; the queue is emptied on a regular basis and should not grow beyond what can be loaded in 3 minutes in BW.

Issue 2 is solved by changing the time when the SID is written to the queue table. It is now no longer written automatically after a new SID is created. However, if the *FL fields (CHECKFL, DATAFL, INCFL) of the SID table are not filled (i.e. no master data or transactional data exists), the SID is not written to the queue. Requirement: the master data load process must be aware of the fact, that it has to write the SID to the queue later on if the master data records are loaded.

Please note that this also means that SIDs might be in the SID table on the DB, but not in the corresponding SID index in BWA. However, provided that they do not reference transactional or master data, they are not used anyway. The data consistency checks for BWA are aware of this and do not report errors or warnings in these cases.

7.10. Change run

In BW7.0, in order to show consistent query results between OLAP Cache, DB (InfoCube and aggregates) and BWA at any time, the BWA indexes have to be set to ‘inactive’ if master data is updated, which is used in an InfoCube with a BWA index. Although we were able to significantly reduce this ‘inactive time’ in BW7.0, it was still a problem for some customers. Thanks to the redesign BWA indexes are no longer set to ‘inactive’ during the master data change run in BW7.30.

7.11. Deletion of master data indexes

Since master data indexes (type MDR, MDD, MDF) are now separate objects, the deletion process has also changed in BW7.30. In general, if a BWA index is deleted, only the individual physical indexes and the metadata (logical) index are deleted. If the BWA index is deleted in the maintenance UI, a popup is displayed asking if a ‘garbage collector’ job should run to delete all master data indexes that are not referenced anymore by other BWA indexes. Only MDR-type and MDD-type indexes are deleted (and not the MDF), which are created specifically for BWA-based F4 help or for accelerated master data provider access.
The application log of this process can be found via the menu Goto → application logs → ‘Deletion of master data indexes’. The job name in SM37 is BI_BIA_MD_INDEX_DELETION.

**Recommendation**

We recommend running this garbage collector job for unused master data indexes if the BWA index is permanently deleted. Do not select this option if it is only deleted and rebuilt later on, to get a new VirtualProvider snapshot or to update the InfoCube fact index after request deletions or compression, for example.
7.12. Miscellaneous (New 1.2)

Selection in Process Chain Type

In the F4-Help of the variant maintenance in process chain type 'Initial Activation and Fill of BIA Indexes' the following BW objects can now be selected:

- InfoCube
- Semantically Partitioned Object
- HybridProvider
- VirtualProvider
- InfoObject
- Query as InfoProvider

![Process Maintenance: Initial Activation and Filling of BWA Indexes](image-url)
BWA7.20 news

The BWA7.20 release has been generally available since Q1 2010 and it is possible to upgrade to BWA7.20 even if the BW system is based on a 7.00 or 7.01 releases. In general, this is done when using the SAP BusinessObjects Explorer, accelerated version, since the Explorer requires the latest BWA release. For BW this Upgrade is transparent, i.e. no processes are changed. But BW does not benefit from the new capabilities of BWA unless it is upgraded to BW7.30 as well. BW7.30 cannot run with a BWA7.00. Therefore the BWA must be upgraded as well.

The central information about the upgrade can be found in SAP note 1392524 (http://service.sap.com/sap/support/notes/1392524) and in the following documents:

http://service.sap.com/~form/sapnet?_SHORTKEY=01100035870000727989&_SCENARIO=011000358700000202&

- Installation guide: http://service.sap.com/~sapidb/011000358700001256762009E

The central innovation in BWA7.20 for BW is the CalcEngine. It enables the creation and execution of complex models on top of the existing BWA layers: Logical Layer (= Logical (OLAP) Indexes) and Physical Layer (Indexes). Using this CalcEngine, BW defines the models for MultiProviders (only at query runtime) and CompositeProviders (at design time), and the data layers for the step-by-step execution for the Exception Aggregation (see the corresponding chapters in this document for more details).
What's New with SAP NetWeaver BW 7.30 and BW Accelerator 7.20?

8. Technical information

8.1. Coding
- RSDDTREX – general services and functionality for BWA
- RSDDTREX_CORE – services used (e.g. by the query runtime, for example)
- RSDDDB – core indexing infrastructure
- RSDDDB_UI – maintenance UI
- RSDDTZA – BWA monitor

8.2. Important tables
- RSDDBOBJDIR – status and versions of BWA indexes for all objects
- RSDDBIDXSTAT – status of indexed tables
- RSDDTREXNEWSID – BWA queue for new SIDs
- RSDDJOBSTAT – status of last 10,000 master data daemon jobs
- RSDDJOBPROT – log of changes executed by master data daemon
- RSDDSTATTREX – BW statistics for indexing
- RSDDSTATTREXSERV – BWA runtime statistics for function calls from BW to BWA (indexing and query executions with special settings)
8.3. Jobs and Logs

- **Index-Creation** in RSDDB
  - Job name: BW_TR_RSDDDB_BIAINDEX_FILL – Indexing in RSDDB
  - Application Log: Object: RSDDTREX / Subobject: TAGGRFILL / Ext. Identifier: BIA_FILL_<InfoProvider> – Fill BIA Index

- **Index-Creation VirtualProvider** in RSDDB
  - Job name: BW_VIP_INIT_RSDDDB_BIAVIP_MAIN (create)
  - Application Log: Object: RSDDTREX / Subobject: TAGGRFILL / Ext. Identifier: BIA_FILL_<InfoProvider> – Fill BIA Index
  - Job name: BW_VIP_DELTA_RSDDDB_BIAVIP_MAIN (Load BWA Index delta)
  - Application Log: Object: RSDDTREX / Subobject: TAGGRFILL / Ext. Identifier: BIA_FILL_<InfoProvider> – Fill BIA Index

- **Index-Deletion** in RSDDB
  - Application Log: Object: RSDDTREX / Subobject: TAGGRFILL / Ext. Identifier: BIA_DELETE_<InfoProvider> – Delete BIA Index

- **Master Data Index-Creation**
  - Job name: BW_TR_RSDDDB_BIAINDEX_FILL – Indexing in RSDDB

- **Master Data Daemon**
  - Job name: BI_BIA_NEWMD_INDEX – Master Data Daemon

- **Deletion of Master Data** Indexes
  - Job name: BI_BIA_BI_BIA_MD_INDEX_DELETION – Delete Master Data (Garbage Collector)
  - Application Log: Object: RSDDTREX / Subobject: BWA_MD_DELETION – Delete Master Data

- **Index Checks** in RSDDBIAMON
  - Job name: BW_TR_BIAMONITORCHECKS – Standard BWA-Monitor-Checkset
  - Application Log: Object: RSDDTREX / Subobject: TAGGRCHECK – Check BIA Index

- **Rebuild all BW Accelerator Indexes**
  - Job name: BW_TR_RSDDTREX_ALL_INDEX_REBUILD – Rebuild BW Accelerator indexes
  - Application Log: Object: RSDDTREX / Subobject: TAGGRFILL / Ext. Identifier: BIA_FILL_ALL
9. Related Content

http://www.sdn.sap.com/irj/sdn/bwa
What's New with SAP NetWeaver BW 7.30 and BW Accelerator 7.20?