Thomas Schröder

SAP® BW Performance Optimization Guide
# Contents

## 1 Introduction and Overview 11

1.1 Structure of the Book ................................................................. 12
1.2 How to Use This Book ............................................................... 14
1.3 Acknowledgements ........................................................................ 15

## 2 Data Warehousing and SAP BW 17

2.1 Introduction to Data Warehousing .................................................. 17
2.2 Differences Between Operational and Decision Support Systems ........ 20
2.3 Structure of Data Warehousing Systems ........................................ 21
2.4 Overview of SAP BW ...................................................................... 24
  2.4.1 Administration and Customizing ............................................... 25
  2.4.2 Data Extraction in SAP BW .................................................... 26
  2.4.3 Data Storage in SAP BW ........................................................ 26
  2.4.4 Metadata Management ........................................................... 30
  2.4.5 Analysis and Reporting ........................................................... 30

## 3 Basic Principles of the SAP BW Architecture 35

3.1 Software Components of an SAP BW System ............................... 35
3.2 Overview of the SAP BW Architecture ......................................... 38

## 4 Basic Principles of SAP Memory Configuration 47

4.1 Terminology Definitions ............................................................... 47
4.2 Basic Principles of SAP Memory Management ............................... 48
  4.2.1 Sequence of Memory Usage .................................................. 53
  4.2.2 SAP Profile Parameters ......................................................... 56

## 5 SAP BW Sizing 61

5.1 The SAP Sizing Process ............................................................... 62
5.2 Size Categories of SAP BW Systems ........................................... 63
9.5 Database Optimizer ................................................................. 227
9.6 Database Statistics ................................................................. 228
9.7 Administration of Database Statistics ..................................... 231
  9.7.1 Administration of Database Statistics Using BRCONNECT ..... 231
  9.7.2 Statistics for InfoCubes ...................................................... 233
  9.7.3 Administration of Database Statistics Using Transaction DB20 ... 235
  9.7.4 Automating the Database Statistics in the DBA Planning Calendar ........................................ 239

10 Reporting Performance ......................................................... 243
  10.1 OLAP Processor ................................................................. 243
  10.2 OLAP Cache ................................................................. 245
  10.3 OLAP Cache Monitor ......................................................... 247
    10.3.1 Cache Structure ......................................................... 250
    10.3.2 Global Cache Parameters ............................................ 252
    10.3.3 OLAP Properties for InfoProviders .................................. 257
    10.3.4 Cache Purging and Swapping ......................................... 259
  10.4 Query Monitor ................................................................. 260
    10.4.1 Query Properties ........................................................ 261
    10.4.2 Debugging Options ....................................................... 268
    10.4.3 Performance Information in the Query Monitor ................. 272
  10.5 Query Trace ................................................................. 273
  10.6 SAP BW Reporting Agent .................................................... 277
    10.6.1 Printing Queries .......................................................... 277
    10.6.2 Precalculating Web Templates ....................................... 280
  10.7 Frontend Performance and Network ..................................... 282
    10.7.1 SAP BEx Analyzer .......................................................... 282
    10.7.2 Web Reporting .............................................................. 286
    10.7.3 Hardware and Software Recommendations for the SAP BW Frontend .............................................. 288
  10.8 Performance Aspects Relevant to SAP BEx Queries and Excel Workbooks ................................................. 290
    10.8.1 Using Cell Editors (Exception Cells) ................................. 291
    10.8.2 Using Formulas in Excel Workbooks .................................. 291
    10.8.3 Using Restricted Key Figures, Filters, and Selections in Queries ... 292
    10.8.4 Query Read Mode ............................................................ 292
    10.8.5 Calculated Key Figures in Queries ....................................... 293
    10.8.6 Queries on MultiProviders ................................................ 294

11 Aggregates ................................................................. 301
  11.1 Basic Principles ................................................................. 303
    11.1.1 Aggregates on Characteristics ......................................... 303
    11.1.2 Aggregates on Navigation Attributes .................................. 306
    11.1.3 Aggregates on Hierarchy Nodes ........................................ 311
    11.1.4 Aggregates on Fixed Values ............................................. 313
11.1.5 Exception Aggregations in InfoCubes .......................................... 315
11.1.6 Line Item Aggregates (Flat Aggregates) .................................. 317

11.2 Automatic Creation of Aggregates ............................................. 319
  11.2.1 Proposing Aggregates from BW Statistics ......................... 321
  11.2.2 Proposing Aggregates from the Query Definition .................. 323

11.3 Manual Creation of Aggregates ................................................ 325
  11.3.1 Analyzing the Query Using the Query Monitor (RSRT) ............ 326
  11.3.2 Creating the Aggregate ...................................................... 329
  11.3.3 Checking and Evaluating Aggregates ................................... 331

11.4 Maintaining Aggregates ........................................................... 332
  11.4.1 Aggregate Rollup ............................................................. 332
  11.4.2 Checking the Aggregate Tree (Rollup Hierarchy) .................... 340
  11.4.3 Summarizing Aggregates/Optimization ................................. 342
  11.4.4 Switching Off Aggregates ................................................. 343
  11.4.5 Analyzing and Monitoring the Filling and Rollup of Aggregates ... 344
  11.4.6 Hierarchy and Attribute Changes ....................................... 346
  11.4.7 Parameterizing the Hierarchy and Attribute Change Runs ........ 348
  11.4.8 Analyzing and Monitoring a Hierarchy and Attribute Change Run .................................................. 350
  11.4.9 Delta Process/Rebuild ...................................................... 351
  11.4.10 Aggregate Block Sizes ..................................................... 352
  11.4.11 Parallel and Serial Change Runs ....................................... 353

12 Compression and Partitioning ..................................................... 357
  12.1 Compressing InfoCubes ......................................................... 357
  12.2 Partitioning InfoCubes ............................................................ 360
    12.2.1 Partitioning at the Database Level .................................... 361
    12.2.2 Partitioning at Application Level ...................................... 366

13 Extraction and Load Performance .............................................. 371
  13.1 Data Sources and Their Integration ....................................... 372
  13.2 Data Flow in SAP BW ............................................................ 374
  13.3 Enabling the Communication Between SAP R/3 and SAP BW ......... 376
  13.4 Transfer Techniques ............................................................... 378
    13.4.1 Application Link Enabling (ALE) ..................................... 378
    13.4.2 Intermediate Document (IDoc) ....................................... 378
    13.4.3 Business Application Programming Interface (BAPI) ........... 378
  13.5 Transfer Methods ................................................................. 379
  13.6 Persistent Staging Area .......................................................... 380
    13.6.1 Partitioning the PSA ...................................................... 382
    13.6.2 Processing Options for the PSA ....................................... 383
  13.7 Monitoring the Load Processes .............................................. 385
    13.7.1 Header Data of the Load Request ..................................... 386
    13.7.2 Status Information for the Load Request ............................ 387
13.7.3 Detail Data of the Load Request .................................................... 389
13.8 Performance Aspects for Data Extraction .................................................... 393
  13.8.1 Maintaining the Control Parameters for the Data Transfer ........ 393
  13.8.2 Main Memory Requirements for the Data Extraction ............ 398
13.9 Performance Aspects Relevant to Data Transformation ....................... 399
  13.9.1 Simulating the Update ................................................................. 401
  13.9.2 ABAP Source Code in Transfer and Update Rules .............. 404
  13.9.3 Optimization Measures in SQL Programming .................... 405
13.10 Performance Aspects Relevant to the Update of InfoCubes .......... 406
  13.10.1 Deleting Secondary Indices ....................................................... 406
  13.10.2 Increasing the Number Range Buffer of Dimension Tables .......... 407
  13.10.3 Uploading Transaction Data ...................................................... 409
13.11 Performance Aspects Relevant to the Update of Standard ODS Objects .......................................................... 411
  13.11.1 Structure of a Standard ODS Object .................................... 411
  13.11.2 Options for Optimizing the Activation Runtime ................. 413

14 Appendix 417

14.1 Transparent Tables of SAP BW Statistics .................................................... 417
  14.1.1 Table RSDDSTAT: Statistical BW Data for Aggregate Selection and Accounting .......................................................... 417
  14.1.2 Table RSDDSTATAGGR: Statistical BW Data for Aggregate Selection and Accounting .............................................. 419
  14.1.3 Table RSDDSTATWHM: Statistical BW Data for Warehouse Management .......................................................... 420
14.2 Job Prefixes in SAP BW ................................................................. 421
14.3 Transactions in the SAP BW System ........................................................ 422
14.4 Transactions Relevant to BW in the SAP R/3 System ......................... 426
14.5 Processing Steps and Calling Programs in the RSMO Monitor ............ 426
14.6 SAP R/3 and BW System Tables ...................................................... 428
14.7 SAP Notes ................................................................. 430
  14.7.1 Database Settings ................................................................. 430
  14.7.2 System Settings ................................................................. 431
  14.7.3 Aggregates ................................................................. 431
  14.7.4 Composite SAP Notes and FAQs ........................................ 432
14.8 SAP Online Resources ................................................................. 432
14.9 Optimizing the Performance of Workbooks Containing Excel Formulas .. 433

The Author 439

Index 441
10 Reporting Performance

This chapter familiarizes you with options for analysis, administration, and optimization of reporting performance. It introduces the tools contained in SAP BW for analysis and improvement, and discusses the effects of query design on performance.

We've already addressed reporting performance to some extent in Chapter 6, which dealt with data modeling, and we'll discuss it further in Chapter 11, which focuses on aggregates. Good reporting performance presupposes the design aspects discussed in data modeling and creating appropriate aggregates. This chapter describes the various monitoring tools of SAP Business Information Warehouse (SAP BW) that help you with analysis during the running of queries. SAP BW also features tuning tools that will help you to optimize the query performance.

You can use the tools described in this chapter for a detailed analysis of running queries, the parameterization of query properties, and for tuning the query performance. As we did in Chapter 6, we'll also discuss various aspects of performance during the creation of queries.

10.1 OLAP Processor

To access the data stored in SAP BW, the end user doesn't need to know which database tables store the key figures and characteristics. Nor does the end user need to call specific aggregates in a query, or pay particular attention to the changed status of data in an InfoCube.

Data access does not occur directly on the data stored in the database; instead, data is accessed with the online analytical processing (OLAP) processor. The OLAP processor is a query management tool that translates the query defined by the end user into a language specific to the database. Then, it returns the data stored in the InfoProviders to the frontend in a multidimensional and formatted view.

Data access occurs with the analysis tools provided by the SAP Business Explorer (BEx), for example, BEx Analyzer, or with third-party query tools. You can use the following interfaces to connect the frontend tools of third parties (see Figure 10.1):
OLE DB for OLAP (ODBO)
OLAP BAPI (Business Application Programming Interface)
XML for Analysis (XML/A)

The interfaces are based on MDX (MultiDimensional Expressions) Processor, a query language developed by Microsoft for queries on multidimensional data.

Figure 10.1 Integrating the OLAP Processor in SAP BW

Table 10.1 lists the essential functions of the OLAP processor:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigation</td>
<td>➤ Dicing according to characteristics and removing dices</td>
</tr>
<tr>
<td></td>
<td>➤ Drilldown into hierarchy nodes and drillup.</td>
</tr>
<tr>
<td>Filtering</td>
<td>➤ Limit and slice selection on characteristic values (individual values, value areas, and hierarchy values)</td>
</tr>
<tr>
<td>Presentation</td>
<td>➤ Display and hide results</td>
</tr>
<tr>
<td></td>
<td>➤ Display key values and texts</td>
</tr>
<tr>
<td>Calculation and Aggregation</td>
<td>➤ Standard aggregation, such as totaling individual values</td>
</tr>
<tr>
<td></td>
<td>➤ Exception aggregation (MIN, MAX, and AVG)</td>
</tr>
</tbody>
</table>

Table 10.1 Functions of the OLAP Processor (Selection)
Analysis of the data is not only determined by the definition of the query, but also by the configuration of the OLAP processor, as described in more detail in the following sections.

### 10.2 OLAP Cache

The OLAP processor has a memory area that stores the result of every query in the main memory of the application server, or in tables or files. By storing the query results, new calls of the query with the same selection criteria, or subsets of the query, don’t require another selection in the database—instead, they can be answered from the faster cache memory of the application server. The OLAP cache is therefore an efficient performance tool to use for optimizing query runtimes.

The OLAP cache distinguishes between the **local cache** and the **global cache**. The local cache stores the results calculated by the OLAP processor for a session specifically by a user in the roll area. It is used when it is impossible to store the query result in the cross-transaction, global cache. Such a situation can arise when the global cache has been deactivated or turned off on the InfoProvider, or on the query. You cannot use the local cache in multiple sessions or by multiple users.

The global cache is a cross-transaction application buffer that stores the results and navigation status of a query in the main memory of the application server. As long as the OLAP processor needs the objects, they are stored in the roll area. All query sessions and query users can use the global cache.

Only the local cache was available up to SAP BW 2.0B. As of SAP BW 3.0B, both the global cache and the local cache are available. All users can

---

**Table 10.1** Functions of the OLAP Processor (Selection) (cont.)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Performance</td>
<td>- Transparent consideration of aggregates</td>
</tr>
<tr>
<td></td>
<td>- Reuse of query results in memory (OLAP caching)</td>
</tr>
<tr>
<td>Data Status</td>
<td>- Transparent consideration of status changes in data (the display of correct and consistent load requests)</td>
</tr>
<tr>
<td></td>
<td>- Transparent consideration of change runs and rollups</td>
</tr>
<tr>
<td>Interface</td>
<td>- Translation of MDX commands from third-party interfaces into OpenSQL</td>
</tr>
<tr>
<td></td>
<td>- Translation of queries from analysis tools of the business explorer</td>
</tr>
</tbody>
</table>

---
call objects buffered in the global cache unless using it is impossible, in which case the local cache is used.

When the OLAP processor uses data from the cache during a query run, the response time of the query is improved, because the read process of the cache is much, much faster than another selection of data in the database.

But a query can use the OLAP cache only if a previous query call used the same selection criteria, or if the new query calls a subset of a selection that has already run. The OLAP cache cannot be used with other selection criteria in a second call of the same query. In this case, the query results are stored in the cache and are available for a repetition of the identical call.

Invalidating the OLAP cache

A query may not use the OLAP cache if the data in the InfoCube has changed (inserted, updated, or deleted). For BasicCubes, non-transac-
tional operational data store (ODS) objects, and master data providers, SAP BW automatically sets a timestamp in Table RSDINFOPROVDATA when the data is changed. When a query is run, the OLAP processor compares this timestamp with the timestamp of the cache package. The cache package is ignored if the timestamp is older than the timestamp in Table RSDINFOPROVDATA.

A query cannot use the OLAP cache in the following cases:

► Activation of master data
► Use of navigation attributes or numeric variables with replacement data from attributes
► Activation of hierarchies used in the query as a selection or presentation hierarchy
► Modification of the query definition and regeneration

Up to SAP BW 3.0B, Support Package 17, a query was always regenerated as soon as something in the definition of the query or a reusable element of the query (a variable, structure, or a calculated or limited key figure) changed. At regeneration, the data used by the query in the OLAP cache was automatically invalidated, in other words, deleted. As of SAP BW 3.0B, Support Package 18 or SAP BW 3.1C, Support Package 12, a modification of the query does not automatically trigger the regeneration of the query. Instead, SAP BW compares the old definition of the query with the new definition and all its subobjects.
A regeneration of the query and deletion of the cache data occurs in the following cases:

- Modification of InfoProviders or the InfoObjects involved
- Modification of the currency translation or currency translation key
- Modification of specific variables or some of their properties

The query is regenerated for modifications of text variables and their use; however, the cache data is not invalidated in this case. The query is not regenerated for modification of texts and the exchange of structure elements.

If a query contains virtual characteristics or key figures, the standard settings of the OLAP cache cannot be used, because the OLAP cache manages only the cache objects in its own buffer area and the data targets based on the query. It does not invalidate the cache when data in other database tables is modified—data read using customer exit variables, for example. Nevertheless, if the OLAP cache is to be used, you must explicitly set the Query Properties in the corresponding dialog window, so that the data is written to the cache after it is read from the database and run through the customer exit.

Note that the OLAP cache is not invalidated after a modification of the data by the customer exit: therefore, the current data can differ from the cached data.

10.3 OLAP Cache Monitor

The OLAP cache monitor is the central monitoring tool for the OLAP cache. You can use the OLAP cache monitor to obtain a view of the global cache parameters, analyze the memory use of the query runtime objects, and analyze the underlying, current cache structure.

You can call the OLAP cache monitor using Transaction RSRCACHE. The Cache Parameter button enables you to call the settings for the global cache (see also Section 10.3.2). The Main Memory button can be used to call the current memory use of the OLAP cache (see Figure 10.2).
Figure 10.2 OLAP Cache Monitor

In the **Technical Info** submenu, you can call the current parameters of the runtime objects:

<table>
<thead>
<tr>
<th>OLAP Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum cache size</td>
<td>Maximum size of the cache in MB. Default: 200 MB</td>
</tr>
<tr>
<td>Current cache size</td>
<td>Total memory required for all cache objects in KB</td>
</tr>
<tr>
<td>Current swap size (for main memory cache mode with swapping or cluster/flat-file cache)</td>
<td>Size of the swap memory (background memory flat file or cluster table) in KB</td>
</tr>
<tr>
<td>Cache filled</td>
<td>Percentage of the size of the filled cache of the overall cache</td>
</tr>
<tr>
<td>Total current entries</td>
<td>Sum of the current cache entries and current swap entries</td>
</tr>
<tr>
<td>Current cache entries</td>
<td>Number of all cache structure elements</td>
</tr>
<tr>
<td>Current swap entries</td>
<td>Number of all entries in the background memory</td>
</tr>
</tbody>
</table>

Table 10.2 OLAP Cache Parameters
You can use the **Buffer Objects: Hierarchical Display** submenu to call a hierarchical view of all buffer objects of a query directory in the OLAP cache. You can use the **Buffer Objects: List Display** submenu to display the cache objects in chronological sequence or a physical view of how they are stored in the cache.

By double-clicking on the **Query Name and Hierarchies/Variables** level, you can call a detailed display that shows the technical name, the date of the cache entry, the creator of the cache entry (the user who called the query), and information on the use of hierarchies or variables for each query (see Figure 10.3).

![Figure 10.3 Detailed Information on the Cache Objects](image)

You can delete selected buffer objects from the cache (see Figure 10.4) by highlighting the cache element that you want to delete and using the context menu. The **Delete** button removes all the buffer objects in the active view from the OLAP cache.
10.3.1 Cache Structure

The objects stored in the OLAP cache are stored in a specific cache structure in the buffer area. They can be displayed using the cache monitor.

From a logical viewpoint, the cache objects are structured hierarchically. A query directory is created for each query; the directory contains the structure elements of the cache.

The query directory maps the memory objects contained in the cache in three levels:

<table>
<thead>
<tr>
<th>Memory Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>Technical name of the query</td>
</tr>
<tr>
<td>Hierarchies and variables</td>
<td>Structure for variables requested for the query</td>
</tr>
<tr>
<td>Selection and data</td>
<td>Structure for the complete selection for the query</td>
</tr>
</tbody>
</table>

Table 10.3 Memory Objects in the OLAP Cache
As shown in simplified form in Figure 10.5, an entry in the query directory of the OLAP cache is created for each query, depending on the selected selection criteria.

In this example, query 0001 is run with selection variable 0CUSTOMER for value area 1100 to 1200. Query name, selection variable, and hierarchy, and selection data are stored hierarchically in the cache data packages. A new cache entry is generated for another call of query 0001 with input variable 0CUSTOMER = 1300, because the selected value set is not yet present in the cache. If the next call of query 0001 selects the input value 0CUSTOMER = 1500, the query can use the cache, because the subset is already present in the cache (from the first query call).

If a second call of a query cannot read data from the cache, the variable assignment is often the reason. If variables are part of the fixed filter in the query, the OLAP cache must be set up again for each query. The reason is that the key of queries in the OLAP cache consists of "query" + "variables which cannot be changed for navigation" + "selection hierarchies". The cache always stores the value sets (or their subsets) calculated by the OLAP processor and can find the stored subsets or value sets for reuse only if the key of the subset is also a subset of the key of the previously calculated value set.

If you select the Can be changed in query navigation setting for a variable, the variable is no longer part of the Hierarchy/Variable key. It becomes part of the selection condition that is one level lower in the cache hierarchy. The same Hierarchy/Variable subtree is selected for variable selection if the entry has been calculated and stored.
You activate the **Can be changed in query navigation** setting in the SAP BW Variables Editor (see Figure 10.6).

To enable the greatest possible reusability of the cache objects, you should allow queries to be changed in query navigation. But please note the following when reusing cache objects. Only subsets of previously selected sets can be read from the cache, because the OLAP processor examines the relationships between the subsets. It does not combine different cache entries into a new subset.

### 10.3.2 Global Cache Parameters

You can set the global cache parameters via the SAP BW customizing menu using Transaction SPRO: **SAP Customizing Implementation Guide** • **SAP NetWeaver** • **SAP Business Information Warehouse** • **Reporting-relevant Settings** • **General Reporting Settings in Business Explorer** • **Global Cache Settings** (see Figure 10.7).
Storing the query results in the OLAP cache requires additional memory in the main memory of the application server. The size of the OLAP cache must be appropriate to manage the frequency of query calls and the number of users. The size of the global cache depends on the size of the local cache. Cache objects that are no longer used are deleted from the roll area when the size of the local cache is exceeded—for both types of cache objects.

You can configure the cache parameters using Transaction RSCUSTV14 (see 2 in Figure 10.7):

- **Cache Inactive**
  Activation of this configuration deactivates the cross-transaction cache. A query can no longer use the global cache—the local cache is used instead.

- **Local Size MB**
  This parameter sets the size of the local OLAP cache (in MB).
Global Size MB
This parameter sets the maximum value of memory use of all objects in the cross-transaction cache (in MB). The memory use is based on the memory requirements of the objects in the shared memory buffer. The memory usage in the shared memory buffer is generally greater, because it stores the OLAP cache runtime objects in compressed from in the application buffer, along with additional administrative data.

When setting the size of the global cache, note that the actual size of the cross-transaction cache is determined by the minimum value of the Global Size MB parameter and the actual memory available in the shared memory buffer (profile parameter rsdb/esm/buffersize_kb). You should therefore use Transaction ST02 to check whether the size of the export/import buffer is appropriate. The default setting of 4,096 KB is often too small. SAP recommends the following settings:

- rsdb/esm/buffersize_kb = 200000
- rsdb/esm/max_objects = 10000.

Persistence mode
The persistence mode sets whether and in what form cache data is to be stored and how the data is used when the maximum memory size has been reached. The following modes are available. You can set the modes in the OLAP cache parameters (Transaction RSCUSTV14).

- Inactive
  The data is deleted from the memory when the memory available for caching has been consumed.

- Flat file
  The data is swapped out into a file when the memory available for caching has been consumed. A repeated call of the cache object loads it into the cache memory.

- Database table
  The data is stored in a non-transparent cluster table or in a transparent table with BLOB\(^1\) (binary large object) in the database when the cache memory has been consumed. A repeated call of the cache object loads it into the cache memory.

---

\(^1\) BLOB (Binary Large Object) is a special type of data that can hold character strings of variable length (up to L+2 bytes, where L < 2\(^{16}\)). It is therefore suitable for storing large data quantities.
The persistence mode is closely related to the cache mode. The cache mode determines how query results and navigation statuses are stored in the cache as compressed files. The persistence mode determines how the cache objects are to be stored when the cache memory has been consumed. You can set the cache mode in Customizing for an InfoProvider as a default value for all queries on the InfoProvider in the SAP BW Customizing menu (Transaction SPRO): SAP Customizing Implementation Guide • SAP NetWeaver • SAP Business Information Warehouse • Reporting-relevant Settings • InfoProvider Properties or selectively for a query in the query monitor (Transaction RSRT). See also Section 10.4 on this topic.

Table 10.4 provides an overview of the cache mode and the persistence mode:

<table>
<thead>
<tr>
<th>Persistence Mode</th>
<th>Cache Mode</th>
<th>Persistent cache per application server or across several application servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inactive</td>
<td>Data is purged from the cache when the cache memory has been consumed (corresponds to the main memory cache without swapping mode).</td>
<td></td>
</tr>
<tr>
<td>Flat file</td>
<td>Data is stored in a flat file when the cache memory has been consumed (corresponds to the main memory with swapping mode).</td>
<td>Cache objects are stored as files in a directory on the application server or across several application servers on a network.</td>
</tr>
<tr>
<td>Cluster table</td>
<td>Data is stored in a non-transparent cluster table when the cache memory has been consumed (corresponds to the main memory with swapping mode).</td>
<td>Data is stored as a non-transparent cluster table in the database (depends on the cache mode with or without application server in the key).</td>
</tr>
<tr>
<td>BLOB table</td>
<td>Data is stored in a transparent table with BLOB when the cache memory has been consumed (corresponds to the main memory with swapping mode).</td>
<td>Data is stored as a transparent table with BLOB in the database (depends on the cache mode with or without application server in the key).</td>
</tr>
</tbody>
</table>

Table 10.4 Relationship Between Cache Mode and Persistence Mode

If you don’t select an entry for the persistence mode (initial value), the system sets the persistence mode to Inactive.

You should check which persistence mode is selected for the queries. For larger result sets, the transparent table (BLOB) can provide better per-
formance, because of its more efficient database operations. For smaller result sets, the cluster table can be advantageous, because the BLOB fields in the database require more administrative effort.

To store cache objects in files (cache modes: Main Memory with Swapping, Flat File Cache per Application Server, and Flat File Cache Across Application Servers), you must first define the logical file path, the physical file path, and the file name. You can maintain the parameters using Transaction FILE, or via the Logical File Names menu in the OLAP cache monitor (see Section 10.3 and Figure 10.8).

Figure 10.8  Maintaining the Logical and Physical File Names

First, you must define the logical file path of the flat file (Step 1: Logical File Path Definition). The platform-independent, logical file path is analyzed at runtime to determine the platform-dependent, physical path. The physical file path is defined in the next step (Step 2: Assignment of Physical Paths to Logical Paths). Table 10.5 provides an overview of the parameterization of the logical file path:
The physical path is platform-independent and must be set according to the system configuration. The file should be located as close to the application server as possible so that it can be found and read quickly. The physical path is set up according to the following structure: `/usr/<SYSID>/global/<FILENAME>`. Except for the `<FILENAME>` parameter, you can select the path per your requirements, as long as it conforms to the configuration of the system. The `<FILENAME>` parameter must be replaced with the physical file name. The complete, platform-specific file name is actually created automatically only at runtime: it consists of the physical path and the physical file name. A specific schema is stipulated for the definition of the physical file name.

To store the file on the application server, you must use the syntax `CACHE_<HOST>_<SYSID>_<PARAM_1>`

For cross-application server storage, you must use the syntax `CACHE_<SYSID>_<PARAM_1>`

The `<HOST>` parameter specifies the name of the computer or server. The variable is unnecessary if the data is stored across application servers. The `<SYSID>` parameter differentiates the systems when two SAP BW systems run on one application server. The `<PARAM_1>` parameter is a sequential number in HEX.

### 10.3.3 OLAP Properties for InfoProviders

You can use the SAP BW Customizing menu to make global settings for the OLAP cache and to parameterize the default settings of an InfoProvider regarding its read and cache modes. Use Transaction SPRO and

<table>
<thead>
<tr>
<th>Logical File Path</th>
<th>Cache Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BW_OLAP_CACHE</td>
<td>Main Memory Cache with Swapping</td>
<td>Logical file path to store the data to be swapped</td>
</tr>
<tr>
<td></td>
<td>Persistent cache for each</td>
<td>Logical file path to store all data if the cache data is to be stored in one file. The application server is an element of the file name.</td>
</tr>
<tr>
<td></td>
<td>application server</td>
<td></td>
</tr>
<tr>
<td>BW_OLAP_CACHESPAN</td>
<td>Persistent cache across</td>
<td>The flat file name indicates the logical file path for storing all data. The application server is not an element of the file name.</td>
</tr>
<tr>
<td></td>
<td>application servers</td>
<td></td>
</tr>
</tbody>
</table>

Table 10.5 Parameterization of the Logical File Path
menu path SAP Customizing Implementation Guide · SAP NetWeaver · SAP Business Information Warehouse · Reporting-relevant Settings · General Reporting Settings in Business Explorer · InfoProvider Properties to call the maintenance dialog (see 1 in Figure 10.9).

After you select the InfoProvider 2, you can set the OLAP properties, Read Mode and Cache Mode for the InfoProvider 3.

The settings apply to all queries that will be created for the InfoProvider. You can make OLAP default settings for existing queries using the query monitor (see Section 10.4).

The read mode of the InfoProvider determines how the OLAP processor retrieves the data during navigation. New queries are then created with the default read mode of the InfoProvider. If no default read mode is set for the InfoProvider, the read mode is set to Hierarchical Reread.

You can use the cache mode of the InfoProvider to set the type of storage for query results calculated by the OLAP processor. You can select from the options listed in Table 10.6:
The cache modes, *Persistent Cache per Application Server* and *Persistent Cache Across Application Servers* are available as of SAP BW 3.0B, Support Package 13, or SAP BW 3.1C, Support Package 07.

For specific InfoProviders, for which the SAP BW system does not control data changes (RemoteCubes and transactional ODS objects, for example), the query results calculated by the OLAP processor cannot be stored, by default, in the cross-transaction application buffer (cache validity = 0 seconds). You can maintain the cache validity, that is, the retention period of the cache objects, for queries of such InfoProviders in the customizing settings of the InfoProvider. The cache validity of queries of other InfoProviders is automatically determined with the timestamp of the last change of their metadata, master data, and transaction data.

### 10.3.4 Cache Purging and Swapping

When the memory capacity of the OLAP cache is exhausted (when the maximum cache size has been reached), cache objects must be purged (deleted) or stored elsewhere so that additional data can be written to the cache. Depending on the cache mode selected, you have two options here:

- **Main Memory Cache Without Swapping**
  
  Data is purged from the cache (deleted).

- **Main Memory Cache With Swapping**
  
  Data is swapped from the cache to a background memory.

<table>
<thead>
<tr>
<th>Cache Mode in InfoProvider</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache is inactive</td>
<td>The cross-transaction cache is switched off for the selected InfoProvider.</td>
</tr>
<tr>
<td>Main memory cache with/without swapping</td>
<td>The cross-transaction cache is switched on for the selected InfoProvider (default value). Cache data is stored in a background cache once the memory available for caching has been consumed (with swapping).</td>
</tr>
<tr>
<td>Persistent cache per application server or across application servers</td>
<td>The cache data is stored persistently in a cluster table or in flat files for each application server or across application servers. Unlike the case with the main memory cache mode, no swapping takes place.</td>
</tr>
</tbody>
</table>

Table 10.6 Cache Mode of the InfoProvider

---

OLAP Cache Monitor 259
You can view the status of the cache objects using the status flag in the OLAP cache monitor (Main Memory - Buffer Objects: Hierarchical Display or Buffer Objects: List Display, see Figure 10.10).

Figure 10.10 Status of the Buffer Objects in the Main Memory of the OLAP Cache

The Write flag is set when a cache element is written to the cache for the first time and the cache mode is persistent. No Write flag is set for the new cache elements if the cache mode is not persistent. The Read flag is set when the cache object is called again. Cache objects with a Read flag have been read from the OLAP cache. Cache objects that have been swapped from the cache to background memory are marked with the Swapped flag. The Dirty flag is set when the data is written to background memory before being purged and the cache mode is persistent. The Dirty flag corresponds to the Write flag with purging. The Directory flag marks the highest node of the content directory of the queries.

10.4 Query Monitor

The query monitor is the administration, testing, and monitoring tool for SAP BEx queries. You can use the query monitor to generate, test, and run SAP BEx queries and to configure general properties of queries. You can call the query monitor using Transaction RSRT (see Figure 10.11).
From the viewpoint of performance, the settings for query properties, the performance information, and the test functions of a query (debugging) are especially important. The following sections describe these points in more detail.

### 10.4.1 Query Properties

In the **Query Properties** dialog window of the query monitor (see Figure 10.12), you can configure settings for the Read Mode, Cache Mode, and Optimization Mode of the query.

The read mode of a query determines how often the OLAP processor retrieves data from the database during query navigation. The query definition tells the OLAP processor which data it must select. The query definition is determined by the InfoObjects of the query, with a distinction between the InfoObjects in rows (key figures + characteristics), the InfoObjects in columns (key figures + characteristics), the free characteristics, and the filter characteristics. As early as the first navigation step of the query, the InfoObject data in the rows, columns, and filter is read from the database. The data of the free characteristics, however, doesn't need to be read until the dicing of the free characteristics in the query occurs.
Figure 10.12 Configuring the Query Properties in the Query Monitor

Figure 10.13 Query Definition
The read mode distinguishes between three types of read processes:

- Read all data (setting A: Query to Read All Data At Once)
- Reread the data (setting X: Query to Read Data During Navigation)
- Reread the data when expanding a hierarchy (setting H: Query to Read When You Navigate or Expand Hierarchies)

The default setting for a new query is Read Mode H: Query to Read When You Navigate or Expand Hierarchies or the setting made using Transaction RDMD in Customizing for the underlying InfoProvider. In the query monitor, you can change the default settings of the read mode for existing queries. The read mode settings made in the query monitor overrule the settings of the InfoProvider underlying the query. You can define the cache validity of the cache objects only in Customizing for InfoProviders.

The Read all Data read mode includes only one read procedure. All the data needed by the query is read from the database in the first step and loaded into the main memory of the OLAP processor. For all additional navigation steps, including the navigation through the free characteristics, the data is aggregated and calculated from the main memory. Another read access to the database is not required.

![Diagram of Read Mode: Read all Data](image)

The example shown in Figure 10.14 selects hierarchy level 2 for the Country object in query call 1 and the fixed value of 9000 for the Material object. Additional navigation steps do not limit Country any further and select the fixed value of 8000 for Material. Despite the changed query selection, query call 2 does not read from the database, because all the selection data of the query is present in the OLAP cache with the first call as the result of navigation.
Once all the data of the query has been loaded into the main memory of the OLAP processor, all additional navigation steps of the query are quite fast, because they don’t need to access the database again. However, the first call of the query is very slow, because all the data, including the data of the free characteristics, must be retrieved from the database in a read step. The read process can last quite a long time for queries with many free characteristics.

The **Read all Data** read mode should be used for only very small Info-Cubes. Because this read mode also reads all the free characteristics of the query in the first step, it provides the least support for the concept of aggregates to store preaggregated subsets of data (see also Chapter 11). Queries that contain many free characteristics also require a great deal of cross-transactional memory for the OLAP cache.

In the **Reread the Data** read mode, the OLAP processor requires only the necessary data for each navigation step. The data of the free characteristics is read only when it is needed for a dice. Data is read from the OLAP cache when the navigation results have already been selected once. Unlike the **Reread the Data when Expanding a Hierarchy** read mode, this read mode always reads external hierarchies completely at the leaf level, even if a query selects a higher level.

![Figure 10.15 Read Mode: Reread the Data](image)

The example shown in Figure 10.15 handles the second query call with `Country` * completely from the OLAP cache. The first query call selected the fixed value of 9000 for `Material` and the entire external hierarchy for `Country` was read into the OLAP cache. The third query call must once
again access the database, because the OLAP cache does not yet contain the navigation result with the fixed value of 8000 for Material.

The **Reread the Data** read mode navigates more slowly than the **Read all Data** read mode, because it must access the database for every navigation step if the navigation result is not yet present in the OLAP cache. However, this read mode is best suited for the use of aggregates, even when the query uses a large number of free characteristics.

The **Reread the Data When Expanding a Hierarchy** read mode selects the smallest amount of data. This is why this mode requires the most reads on the database. Where the **Reread the Data** read mode reads the expanded hierarchy completely in the first step, the **Reread the Data When Expanding a Hierarchy** read mode reads only the data up to the selected hierarchy level in the first step. If a deeper level of the hierarchy is required in another navigation step, the database must be accessed again.

![Diagram of Reread the Data When Expanding a Hierarchy](image)

Figure 10.16 Read Mode: Reread the Data When Expanding a Hierarchy

The second query call with hierarchy level 2 in the example shown in Figure 10.16 can use the OLAP cache from Country, because the first call already selected the deeper query hierarchy level, Level 3, of Country. The third query call must access the database, because hierarchy level 4 of Country is not yet present in the OLAP cache as a navigation result.

When you use hierarchy aggregates, you must set the read mode of the query to **Reread the Data When Expanding a Hierarchy**; otherwise, you can’t use the hierarchy attributes. You should use this read mode for large hierarchies (from about 500 hierarchy nodes). If you don’t, the first call in the **Reread the Data** read mode can result in long waits.
Table 10.7 provides an overview of all three read modes.

In most cases, the **Reread the Data When Expanding a Hierarchy** read mode provides the best response times, because each navigation step has to read only the required data.

<table>
<thead>
<tr>
<th>Read Mode</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Recommendation</th>
</tr>
</thead>
</table>
| Read all data                                  | ▶ Very fast query navigation after the first call, because all the data is present in the OLAP cache. | ▶ First call is slow  
▶ Significant limitations on the use of characteristic aggregates  
▶ Requires more memory in the OLAP cache | ▶ Use this read mode only with small InfoCubes  
▶ Use this read mode only in queries containing few free characteristics. |
| Reread the data                                | ▶ The first call is very fast, because only the required data is selected.  
▶ Good hit rate for characteristic aggregates  
▶ Rapid response time for small hierarchies | ▶ Requires waiting for additional calls if the selection is not identical to the first call. | ▶ Use this read mode for small hierarchies  
▶ Use this read mode with large quantities of results. |
| Reread the data when expanding a hierarchy    | ▶ The first call is very fast, because only the required data is selected. | ▶ Selects the smallest amount of data in the first call, so that changes to navigation require read accesses to the database. | ▶ The use of this read mode is required for hierarchy aggregates. |

The read mode of an InfoCube defined via Transaction SPRO in the Customizing Guide is stored in Table RSDCUBE. The read modes of a query defined in the query properties via Transaction RSRT are stored in Table RSRREPDIR. Both tables provide a quick overview of the read modes for an InfoCube and for a query.

You can select the tables using the table browser (Transaction SE16). Figure 10.17 shows Table RSDCUBE. To view only active and usable InfoCubes, you should limit the selection to OBJVERS = "A" and OBJSTAT = "ACT".
The `READMODE` column indicates the read mode of the InfoCube:

- **A** = query reads all data at once
- **X** = query to read data during navigation
- **H** = query to read data when you navigate or expand hierarchies

The `CACHEMODE` column indicates the cache mode of the InfoCube:

- **0/Blank** = cache is inactive
- **1** = main memory cache without swapping
- **2** = main memory cache with swapping
- **3** = persistent cache for each application server
- **4** = persistent cache across application servers

Buffer objects are chronologically stored flat in the OLAP cache as data packets. After the first run of a query and every 31 days after the last optimization, the OLAP processor determines the optimal initial size of the cache.
Storage Package (SP) cache packages. You can set the time of the optimization of the memory structure of cache packages in the query properties.

![Query Optimization Mode](image)

You can set the optimization mode of the query in the dialog shown in Figure 10.18:

- **Option 0**
  Query Will Be Optimized after Generation. The query is optimized after generation in this optimization mode.

- **Option 1**
  Query Optimization with Individual Periods in Days. This mode is identical to option 0, but you can also select the period of optimization in days.

- **Option 9**
  Query Optimization Inactive. In this mode, the memory structure of the table is not optimized after generation of the query.

### 10.4.2 Debugging Options

To enable specific examination of individual queries, SAP BW provides a query monitor that you can call using Transaction RSRT (see Figure 10.19). The query monitor enables the execution of individual queries with various debugging options.

After you select a query, it can be executed via the **Execute + Debug** button with a selection of various debugging options. The debugging options provide various ways for you to display or examine specific elements of a query. For example, you can select the **Display SQL/MDX Query** option to display the SQL statement of the query.
For complex queries, the **Display Aggregate Found** option enables you to display the aggregate in use. Queries on MultiProviders display the aggregates with their technical names for all the InfoProviders involved, one after the other. If an aggregate cannot be used, the InfoCubes involved are displayed along with all the InfoObjects and filter settings used in the query. When displaying an aggregate with its six-digit ID, you can use Table RSDCUBE to determine the InfoCube assigned to the aggregate.
In Figure 10.20, the query executes a database access on Aggregate 100450. The first column lists all the database accesses, one after the other. The \textit{Aggregate/InfoCube} column lists the aggregate by its six-digit, internal SAP BW ID, or the InfoCube. The \textit{InfoObject} columns list all InfoObjects required in the query access with their technical names or semantic descriptions. Some entries are listed in two separate, marked boxes, because in this way, help you to better understand their function. The first box whose column names begin with “S” contains suggested values for the aggregation type, the hierarchy used (if the aggregation type is H; otherwise, it’s 0), the hierarchy level, and fixed values (if the aggregation type is F; otherwise, it’s 0). The names of the columns in the second box begin with “A”. This box contains the corresponding entries for the aggregate found.

The aggregation type can have the following properties:

- \textbf{*}: all values
- \textbf{F}: fixed value
- \textbf{H}: hierarchy level
- \textbf{G}: navigation attribute
- Blank
As shown in Figure 10.20, the aggregate found contains many InfoObjects, including 0CALDAY, 0CALMONTH, 0CALWEEK, 0FISCPER, and 0FISCVARNT, where no fixed values or hierarchy level is set for any of these objects.

Note that the results column of the fixed values in the query monitor RSRT shows only the SID values, not the characteristic values themselves. You can identify the corresponding fixed value with the SID key in SID table /B10/S<InfoObject> or /BIC/S<InfoObject>.

If aggregates are already present but are not used, you can use the Select Aggregate debugging option to see why the available aggregates have not been used (see Figure 10.21). The InfoObjects that lead to the non-use of an aggregate are listed for all aggregates of the InfoCube. This lack of use always occurs when the corresponding InfoObject is not contained in the aggregate, or an inappropriate fixed value is defined for the InfoObject in the aggregate, or the query requires a lower degree of detail than exists in the aggregate.

Figure 10.21 “Select Aggregate” Debugging Option
When you display the results of the query in the query monitor, you can choose from the following options: List, BEx Analyzer, and HTML (see Figure 10.22). If you want to check various query navigation steps for the use of aggregates, you should select the HTML display option, because it provides the full scope of query navigation. The selected debugging option is executed at each query navigation step.

Figure 10.22 Query Results in HTML in the Query Monitor with Complete Navigation Options

10.4.3 Performance Information in the Query Monitor

You can use the Performance Info button shown in Figure 10.11 to call performance-relevant information on the query that does *not* correspond to the system recommendations. The information refers to the following areas:
The query trace is another option for logging individual query steps during the execution of a query. The trace must be switched on separately for each user for whom you want to record a trace. You can activate a trace using Transaction RSRTRACE or SAP BEx Analyzer by selecting the Trace option (see Figure 10.23).

Table 10.8 Performance Information in the Query Monitor

<table>
<thead>
<tr>
<th>Performance Info Area</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Definition</td>
<td>Query cannot use the aggregates (corresponds to the information provided in Technical Information under OLAP-relevant Data)</td>
</tr>
<tr>
<td></td>
<td>Read mode X or A (see read mode, Section 10.4.1)</td>
</tr>
<tr>
<td></td>
<td>Query cannot use the cache (corresponds to the information provided in Technical Information under Cache-relevant Data)</td>
</tr>
<tr>
<td></td>
<td>Query uses customer exits (see also Section 10.2)</td>
</tr>
<tr>
<td></td>
<td>There are non-cumulative values with AVi (corresponds to the information provided in Technical Information under Non-cumulative flags)</td>
</tr>
<tr>
<td>InfoProvider</td>
<td>The InfoProvider is a MultiProvider (see also Section 10.8.6)</td>
</tr>
<tr>
<td></td>
<td>DB statistics require checking (see also Section 9.6)</td>
</tr>
<tr>
<td></td>
<td>DB index requires checking (see also Section 9.4)</td>
</tr>
</tbody>
</table>

Figure 10.23 Switching on the Query Trace via SAP BEx Analyzer

Query Trace

273
You can configure the trace for a specific topic when you activate the query trace using Transaction RSRTRACE (see Figure 10.24). You must add individual users with the Plus (+) button and can remove them from the trace by clicking on the Minus (–) button. You can use the Configure user button to configure the trace for specific users.

Whether you activate the trace via the Trace option in SAP BEx Analyzer or by using the trace tool RSRTRACE, the trace only goes into effect when the user logs on again, after the trace has been activated. Note that you should activate the trace only for specific analyses, because the logs that result from the trace can affect the performance negatively.

After you execute the query steps that you want to analyze, you should turn off the trace, because constant trace recording demands a great deal of system performance. The log recordings are stored in a trace log and can be called at any time. You can use the DB Debugging button to go through individual steps of the trace at a later time.

You can list the recorded trace logs via the All logs button (see Figure 10.25). Double-click on one of the trace logs in the list to display the recorded trace (see Figure 10.26).
You can use the **Analyze Trace** button to call Transaction RSRCATTTRACE (see Figure 10.27).

Figure 10.26  Log Steps of Trace Recording
Figure 10.27 Test Program RSRCATTRACE

Figure 10.28 Playing Back a Recorded Log Using RSRCATTRACE
You must enter the log number of the trace log and the system in the initial view of test program RSRCATTTTRACE. You can then use the **Execute + Debug** function to rerun the recorded log, just as it was recorded at the time of logging. The system displays the dialog box familiar from Transaction RSRT once again. Here you can select functions to display the SQL statement, or the aggregates found (see Figure 10.28).

Transactions RSRTRACE and RSRCATTTTRACE are particularly well suited to generating suggested aggregates for the first navigation step and all additional navigation steps of a query.

### 10.6 SAP BW Reporting Agent

The reporting agent in SAP BW enables the execution of various reporting functions in the background. For example, you can automatically print queries in the background or precalculate web templates. In addition to other functions of the reporting agent, these two procedures are especially appropriate for warming up (populating) the OLAP cache with the results of the report precalculation, in order to enable faster live access to data from the OLAP cache with identical query navigation.

#### 10.6.1 Printing Queries

You can make the settings required for printing queries in the Reporting Agent menu of the Administrator Workbench. In the **Print** submenu, first select the InfoCube and the query for which you want to generate automatic batch printing.

If no reporting agent settings exist for a query, you must first create them (see ![Creating reporting agent settings](#) in Figure 10.29). In the subsequent screen ![Creating reporting agent settings](#), you must enter a technical name for the query-specific reporting agent settings in the **General** tab. You can use the **Print settings** tab to set the printing process. The **Print layout** tab enables you to configure the design of the InfoObjects of the query in print. If you want to use the batch printing function of the reporting agent only to populate the OLAP cache in batch, you do not need to specify the print settings and print layout in any more detail.

If you want to schedule a query with input variables for batch printing, you must first create a variant for the input of the variable (see ![Creating reporting agent settings](#) in Figure 10.30).
Figure 10.29  Creating New Reporting Agent Settings for Batch Printing

Figure 10.30  Creating Variants for Input Variables in the Reporting Agent
With the creation of a variant 2, the values for the input of the variables at runtime are already preselected to use for printing in the background 3.

After you create the reporting agent settings, you must create a scheduling package for background processing (see Figure 10.31). The scheduling package is used to set the time of background processing for the query. You can combine several reporting agent settings in one scheduling package. The settings are then precalculated in a single job (see also 1 in Figure 10.32).

![Figure 10.31 Creating Scheduling Packages in the Reporting Agent](image)

In the next step, you use drag and drop to assign a reporting agent setting to a scheduling package (1 in Figure 10.32). You can then schedule the scheduling package as a job (2 and 3) or insert it into a process chain.
10.6.2 Precalculating Web Templates

The calculation of web templates is another option for the automatic pre-calculation of query results. Web templates are HTML documents with placeholders specific to SAP BW. The templates display query results in a web browser. You can use precalculated web templates to write query results to the OLAP cache automatically, where they remain available for identical navigation steps, or where they can be called as static web reports, without having to execute an OLAP query.

You must first create a web template to be able to precalculate query results with web templates (see Figure 10.33). Then, you must create the reporting agent settings 1 for the web template.
You can configure the reporting agent setting in the subsequent screen 2. Use the **General** tab to define the name of the reporting agent setting and to enter information on the last change (after the first save) and information on the scheduling packages that will use the setting.

You can use the **Parameter** tab to define what is to be precalculated. For the calculation of large result sets, you should select the **Data** option in the **Calculate** menu because the OLAP processor will not have to request the data during later calls of the web template—only the HTML page will have to be generated. When you use workbooks, note that the reporting agent cannot precalculate workbooks. If you want to place the results of a workbook into the OLAP cache, you must schedule each query individually.

If the query uses input variables, you can create a variant that stores the selection values—just as you can use precalculation with background printing.
After you enter the reporting agent settings, you must generate a scheduling package that can be executed via a job or a process chain. The procedure is similar to the configuration of batch printing (see Section 10.6.1).

Both procedures are appropriate for automatic precalculation of a query and making the resulting quantities available as cache elements in the OLAP cache. The procedures differ in the medium used for the query results: static web report, Excel, or batch printing.

10.7 Frontend Performance and Network

In addition to the previous comments on the performance of SAP BW reports, the runtime of a query is also affected by the query frontend and the capacity of the network to transmit data. The following section first describes the basics of communication between the frontend PC and the application server. It highlights the differences between frontend performance, depending on the query tool selected (SAP BEx Analyzer and Web Analyzer), the network capacity available, and the optimization options for frontend performance.

SAP BW provides two tools for the execution of reports: SAP BEx Analyzer for reporting in Microsoft Excel, and SAP Web Applications for web reports that you can design in a web browser using SAP Web Application Designer (WAD).

10.7.1 SAP BEx Analyzer

SAP BEx Analyzer is an Excel-based client query tool of SAP BW and is based on an add-in for Microsoft Excel: sapbex.xla. Excel add-ins (*.xla files) are supplemental programs that provide user-defined commands and functions (for data analysis, for example) in Excel.

The SAP BW frontend is a frontend component specific to SAP BW that is installed locally with the SAP GUI on the frontend PC in the ...\SAP\Frontend\BW directory or on a Windows Terminal Server (WTS).

The SAP frontend for Windows (SAP GUI for Windows) has been delivered in compilations since July 1999. A new compilation is assembled when the components contained within it (such as the add-on for SAP BW or SAP Supply Chain Management, SAP SCM) have changed. The new compilation contains all the components of the previous version, the patches that have appeared in the meantime, and the new components.
Index

A
ABAP buffer → program buffer
ABAP Dictionary buffer 52
ABAP routine 400, 404, 405
abap/heap_area_dia 50, 54, 57
abap/heap_area_nondia 50, 56, 57
abap/heap_area_total 55, 56, 57, 144
abap/heaplimit 50, 58
Activation queue 67, 412, 413
size of activation queue 413
Administrator mode → user mode
Administrator Workbench 25
Aggregate 23, 303
aggregate tree 340
aggregation levels 303
automatic creation 319
block size 352
compression 336
creating 329
delta process 351
dimension table 319
E fact table 304, 336
evaluation 331
F fact table 304, 336
fact table 319
hierarchy/attribute change run 307
key figures with exception aggregation 306, 315
line item aggregate 317
line item dimension 305
manual creation 325
of the hierarchy and attribute change run 346
on characteristics 303
on fixed values 313
on hierarchy nodes 311
on navigation attributes 306
on time-constant navigation attributes 307
on time-dependent navigation attributes 308
optimizing 322, 325, 342
proposing from SAP BW statistics 320, 321
proposing from the query definition 320, 323
rebuild 351
rollup 302, 332, 336, 337
rollup hierarchy 340, 341
switching off 343
ALE → Application Link Enabling
Alloc fault rate 124
Allocated memory 142
Allocation retries 124
Application Link Enabling 378
Application server 118
Asynchronous RFC 377
Attribute 93, 96, 98

B
B*-tree index 114
BAPI 26, 372, 378
BAPI interface 31
Binary search 193, 194
Binary tree 194
BRCONNECT 231, 234
B-tree 195, 196
Buffer pool 120
Business Application Programming Interface → transfer techniques
Business content 181, 372
BW benchmark 70
BW statistics 165, 166, 182, 319
BW system load 173

C
Cache mode 255
of an InfoCube 258, 267
Cache parameters 252
global cache parameters 252
Cache purging 259
Cache structure 250
Calling program 389
Calls 124
Cardinality 114, 197, 205
Cell editors 291
Change log 67, 412
Change run 348, 349
delta process 351
parallel 353
serial 353
Character 29
Characteristic 93
Characteristics hierarchy 105
Commit 124
Communication interfaces 371, 372
Communication structure 374, 399
Communication techniques 376
Compression 92, 357, 359
Computer 118
Context change 49, 50
Cost-based optimizer 227, 228
CPU bottleneck 146
CPU capacity 146
CPU load distribution 76
CPU sizing 69
CPU time 160
CPU utilization 146
CUA buffer 53

D
Data buffer 120
Data buffer quality 122
Data cube → data buffer
Data cube 19, 22
Data extraction 26, 371, 373, 374, 398
memory requirement 398
Data extraction from flat files 373
Data extraction from multidimensional databases 373
Data extraction from relational databases 373
Data extraction from third-party systems 374
Data IDoc 379, 392, 398
Data package 357, 379, 384, 392, 393, 399
Data package number 380
Data package size 392, 396
maintenance 396
Data staging 371
Data transformation 399
Data warehouse 17, 21
Data warehouse architecture 21
Data warehouse frontend 24, 31
Database buffer 120, 130
Database instance 118
Database monitor 119
Database optimizer 189, 191, 227
cost-based optimizer 227
rule-based optimizer 227
Database server 118
Database time 159
DataMart interface 372
DataSource 374, 375, 400
DB Connect 26, 372, 373
DB parameter 136
DB runtime 178
DB statistics 189, 228
administration 231
BRCONNECT 231, 234
DBA Planning Calendar 239
statistics for InfoCubes 233
UPDATE STATISTICS 231, 236, 239
db_block_buffers 121
db_block_size 121, 124
db_cache_size 121
DBA Planning Calendar 137, 239
Debugging options 268
DIAG protocol 40, 283, 284
Dialog process 383, 384, 415
Dialog work process 53, 54, 55
Dimension ID 90, 201, 407
Dimension table 29, 90, 92, 199, 407
Dispatcher 156

E
E fact table 92, 201, 358
eM/blocksize_KB 57
eM/initial_size_MB 50, 54, 57, 141
Enqueue time 158, 159
Enterprise Core Component 15
ETL process 22, 371
Exception aggregation 114, 315
reference characteristic 315
Exception cells → cell editors
Execution plan 191, 207, 208
full table scan 192, 211
index full scan 211
index range scan 193, 212
index unique scan 193, 212
operation types in execution plans 211
reading the execution plan 211
star join execution plan 207
structure 209
Expert mode → user mode
Export/import buffer 140
Extended memory 140, 154
External mode 48
Extract structure 374, 375
Extraction 371, 378
Extraction structure 26, 372
Extractor program 372
Extractors 375
  application-specific 375

F
Fact table 29, 92, 199, 357, 406
  E fact table 92, 201
  F fact table 92, 201, 358
Flat aggregate → line item aggregate
Flat file 26, 372, 373, 397
Frontend network time 283, 287
Frontend runtime 180
Full table scan 128, 192, 211

G
Generic key buffer 53
GUI time 157, 283
GUI time → individual statistical records

H
Hard disk capacity 64
Hardware sizing → sizing
Heap memory 140, 154
Heavy queries 73
Hierarchy 105, 107, 116
Hierarchy and attribute change runs 346
  monitoring 350
  parameterizing 348
Hierarchy leaves 105
Hierarchy node 105
Hierarchy root 105
High user 73
Histogram 228
Hit 121
Hit ratio 121, 140
Hit ratio → SAP buffer quality

I
I/O bottleneck 148
ICF service 41
ICM 40, 41
ICM handlers 41
ICM monitor 44
ICM profile parameters 43
IDoc 378, 379, 387
IDoc status 391
IEMon.exe 287
Import/export buffer 53
Index 23, 190
  administration 216
  bitmap index 197, 205, 214, 215
  B-tree index 195, 196, 203, 205, 215
  check 216
  clustering index 203
  create 225
  delete 220
  detailed analysis of index quality 223
  encode vector index 202
  first-level analysis of index quality 222
  fragmentation 222
  index quality 222
  index schema 201
  indexing schema 199
  missing indices 218
  on master data tables 225
  P index 191, 202, 360
  primary index 190
  radix index 202
  reorganization 221, 222, 225
  repair 222
  secondary index 191
  table index 194
  UNIQUE index 191
Index full scan 211
Index range scan 193, 212
Index unique scan 193, 212
Individual statistical records 284
Info IDoc 378, 379, 386, 398
Info IDoc status 390
InfoCube 19, 29
  compression 357, 359
  E fact table 358
  F fact table 358
  structure-specific properties 367
InfoObject 26
InfoProviders 28
Information Broadcasting 31
InfoSet 28, 29
InfoSet query 29, 414
InfoSource 374, 375, 399
Intermediate documents 378, 387
Internal mode 48
Internet Communication Framework 40
Internet Communication Framework service 41
Internet Communication Manager 40, 41

K
Key figure 27

L
Leaf block 197
Leaf rows 197
Line item aggregate 317
Line item dimension 92, 94, 95, 112, 114, 205, 317
Load request 385, 386, 387
detail data 389
header data 386
overall status 388
status information 387
Load time 157
Local memory 47
Lock logics 339
Log buffer 123
Logical unit of work 377
Long table scans 129
Low user 73
LUW → Logical unit of work

M
Main memory 142
Master data 29, 93, 96, 409
Master data table 95, 98
Medium queries 73
Medium user 73
Memory management monitor 51
Memory sizing 77
Memory/Disk sort 129
Metadata management 30
Metadata repository 30
Mode context 48
MONI 162
Monitoring 385
Multidimensional data storage 19
MultiProvider 28, 29, 166, 294
heterogeneous MultiProviders 295
homogeneous MultiProviders 295
parallel processing 295, 296
sequential processing 295
turning off parallel processing 297
MultiProviders queries on MultiProviders 166
mySAP ERP 15

N
Nametab buffer 52
Navigation attribute 102, 115
No. of roundtrips → individual statistical records
Non-cumulative key figure 114
Non-dialog work process 53, 55
Normalized form 21
Number range 407
Number range buffer 407
Number range object 407

O
ODBO interface 31
ODS object 67, 411, 414
activation queue 67, 412
active data 68, 412
change log 67, 412
OLAP 19
global cache 245
invalidating the OLAP cache 246
local cache 245
OLAP cache 178, 179, 247, 251
OLAP cache monitor 247
OLAP data storage 19
OLAP processor 165, 166, 173, 174
OLAP runtime 179
OLAP BAPI 31
OLAP cache 245
warming up 277
OLAP processor 243
OLAP Properties for InfoProviders 257
OLAP system 20, 21
OLE DB for OLAP 31
OLTP 20
OLTP system 20, 21
Online Analytical Processing → OLAP
Online Transaction Processing 20
Open hub service 372
Operating system collector
SAPOSCHOL 145
Operating system monitor 145
Optimization mode 267

P
P index 360
Package dimension → dimension table
Page memory 154
Paging area 140
Paging rate 147
Parses 125
Partitioning 23, 215, 295, 360
at application level 295, 366
at database level 361
configuration 364
hash partitioning 361
logical partitioning 23
maximum number of partitions 365
partitions 23
range partitioning 361
setting the data distribution 367
Partitioning characteristic 215, 360, 363, 367
Performance trace 404
Persistence mode 254, 255
Persistent Staging Area → PSA
PGA → Program Global Area
Physical reads 121
PRIV mode 54
Processing time 160
Program buffer 53, 140
Program Global Area 120, 130
PSA 379, 380, 381, 382, 383, 392
partitioning 382
processing options 383
PSA table 68, 380
PSAPTEMP 133

Q
qRFC → queued RFC
Query monitor 191, 204, 207, 230, 260, 268, 272, 326
documenting options 268, 326
performance info 272
Query properties
optimization mode 267
read mode 261, 263, 264, 265, 266
Query trace 273
Query types 73
QueryCube 31
Queued RFC 377

R
rdisp/PG_MAXFS 52
rdisp/PG_SHM 52
rdisp/ROLL_MAXFS 56
rdisp/ROLL_SHM 56, 141
Read mode 116
of a query 292
of an InfoCube 258, 266, 267
Reads 121
Recursive calls 124
Redo log buffer 123
Remote function call → RFC
RemoteCube 28, 29
RemoteCube with service 28, 29
Reporting agent 277
scheduling package 279
Repository buffer 52
Request 339
Request ID 92, 334, 336, 337, 357, 380
Response time in an SAP system
database time 159
dispatcher wait time 159
enqueue time 159
execution time 159
load time 159
processing time 160
roll wait time 160
roll-in 159
rollout 160
RFC 283, 376
asynchronous RFC 377
queued RFC 377
synchronous RFC 377
transactional RFC 377
Roll area 140
Roll memory 141, 154
Index

Rollback 124
Roll-in 50, 159
Rollout 50, 160
Rollup 332, 337
  manual 337
  monitoring 344
Rollup hierarchy 340, 341
Rollup jobs 345
Root block 197
Roundtrip 157, 283, 284
RSDDSTAT 167
RSMO monitor 385, 386, 392, 401
Rule-based optimizer 227
Run schedule 328

S
SAP Basis system 35
SAP benchmark 70
SAP BEx Analyzer 282, 285
SAP buffer 52, 138
SAP buffer quality 140
SAP Business Explorer 31
  Analyzer 31
  Query Designer 31
  Web Analyzer 31
  Web Application Designer 31
SAP BW system load 182
SAP BW system load analysis 164
SAP BW system load monitor 164, 174
SAP calendar buffer 53
SAP ECC 15
SAP enqueues 158
SAP executable buffer → program buffer
SAP Extended Memory 48, 50, 53, 54
SAP GoingLive Check 63
SAP GUI 282, 283, 289
SAP GUI buffer 53
SAP Heap Memory 50, 53
SAP instance 118
SAP memory areas 47, 51, 137, 140
SAP memory management 48
SAP memory management monitor 137, 138, 141, 142
SAP memory management system 53
SAP Paging Memory 52
SAP performance monitor 117
SAP profile parameters 51, 56, 58
SAP Quick Sizer 62, 86
SAP RemoteCube 28, 29
SAP roll area 49, 50
  local SAP roll area 49
  shared SAP roll area 49
SAP roll file 49
SAP Roll Memory 48, 53
SAP Service Marketplace 62, 63, 87, 88
SAP system load analysis 163
SAP Web Application Server 35, 38, 40
SAP work process 148, 397
SAP work process monitor 150
SAP work process types 149, 162
SAP workload analysis 155
SAPS 70, 75
Screen buffer 53
Search algorithm → binary search
SEM InfoCube → transactional InfoCubes
Service engineer mode → user mode
SGA 130
SGA → System Global Area
Shared buffer 53
Shared cursor cache → shared pool
Shared memory 47
Shared pool 120, 122, 125
Shared SQL area → shared pool
shared_pool_size 123, 125
Short table scans 128
SID key 90, 201, 410, 414
SID table 101, 409, 410, 414
Single record table buffer 53
Sizing 61
  CPU 69, 71, 72
  data volume increase 66
  dimension tables 65
  fact table 66
  hard disk capacity 64, 69
  InfoCubes 65
  memory sizing 77, 78, 80
  ODS object 67, 68
  PSA tables 68
  SAP GoingLive Check 63
  SAP Quick Sizer 62, 86
  SAPS 70
  T-shirt sizing 63
Sizing process 62
Slowly changing dimensions 91
Sort buffer → Program Global Area
SP stacks → support package stacks
SQL trace 404
Standard ODS object → ODS object
Star schema 29, 89, 199
Support Package Manager 37
Support package stacks 36
Support packages 35, 36
Swap space 51, 144
Swapping 51
Swaps 140
Synchronous RFC 377
System Global Area 120, 130
System Load Monitor 173

T
Table buffer 53
Table scan 128
Technical content 181
for extraction 184
for SAP BW objects 185
transferring technical content 183
Terminal-in message → individual statistical records
Terminal-out message → individual statistical records
Texts 95
Time dimension → dimension table
Time-constant attributes 97
Time-constant navigation attributes 102
Time-dependent attributes 99
Time-dependent navigation attributes 103
Transaction data 409
Transactional InfoCubes 214
Transactional RFC 377
Transfer methods 379, 380
IDoc 379
PSA 379, 383
Transfer rules 375, 399
Transfer structure 374, 399
Transfer techniques 371, 378
  Application Link Enabling 378
  BAPI 374
  intermediate documents 378
Transformation 371
tRFC 377, 379, 383
T-shirt sizing 63

U
UD Connect 26, 372, 373
UNION operation 29
Unit dimension → dimension table
Update rules 375
Update simulation 401
User calls → calls
User context 48, 49, 50, 54, 156
User mode 161, 162
User session 152
User types 73

V
VirtualProvider 28, 29

W
Warehouse management 371
Web frontend network time 287
Work process → SAP work process
Work process monitor 150
Workload Monitor 160, 161, 162, 163

X
XML data 26, 373
XMLA 31
XMLA interface 31

Z
ztta/roll_area 49, 54, 55, 56
ztta/roll_extension 50, 54, 57
ztta/roll_first 49, 54, 56