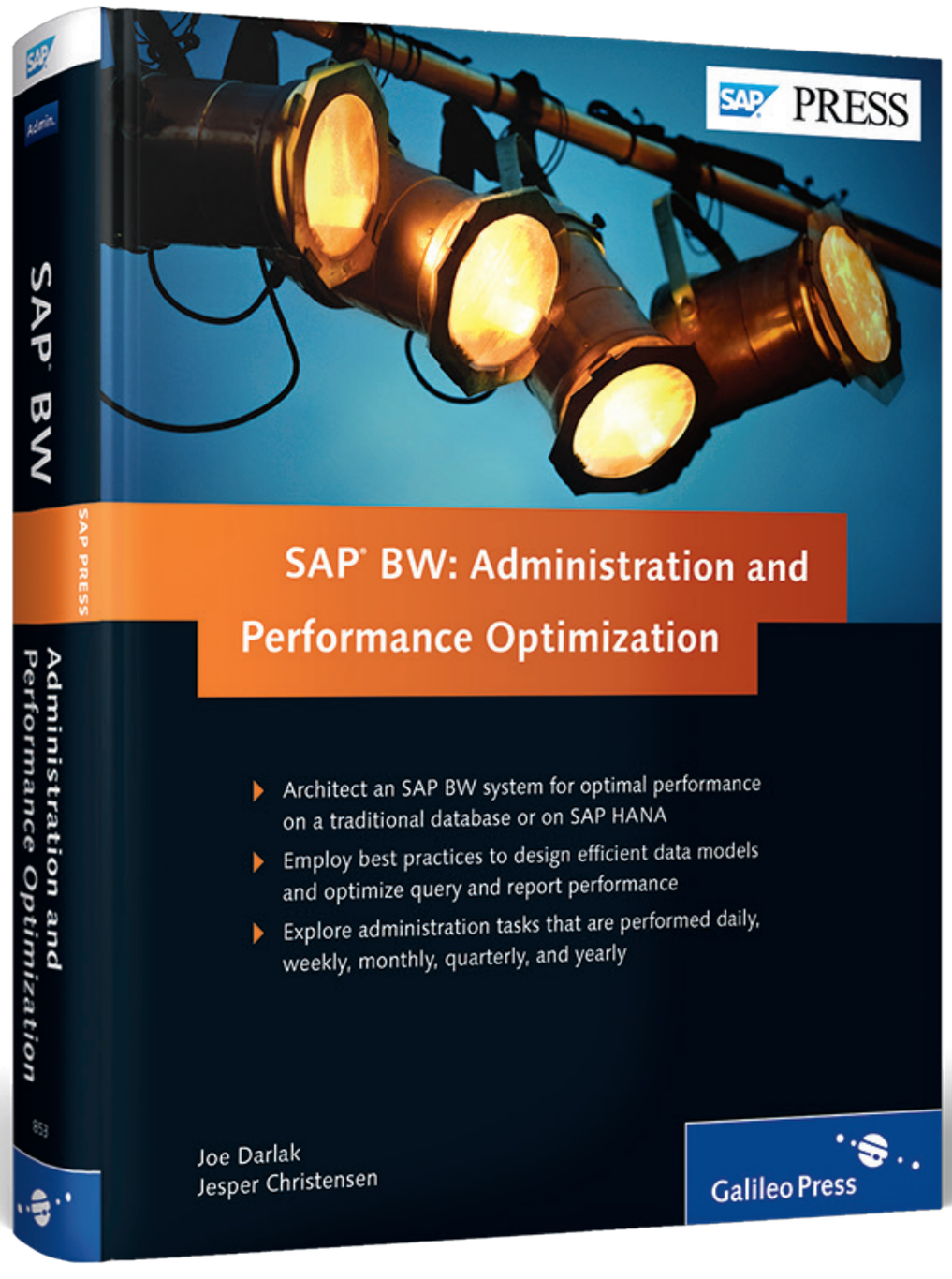


Reading Sample

This reading selection offers the weekly administration and performance tuning tasks to help you keep your database small and your system operating within healthy limits. The book provides tasks for daily, weekly, monthly, quarterly, and yearly completion.

Joe Darlak, Jesper Christensen

SAP BW: Administration and Performance Optimization

652 Pages, 2014, €79.95
ISBN 978-1-59229-853-2

www.sap-press.de/3341

Weekly monitoring and administration can prevent small issues from growing into critical issues. Consistent housekeeping keeps the database small and the system operating within healthy limits.

8 Weekly Tasks

This chapter will split its focus between weekly administration tasks (Section 8.1) and weekly performance tuning tasks (Section 8.2).

All of these tasks should be performed by a combination of BW Basis, BW Technical, and/or data load monitoring sub-teams within the support organization. While some of these tasks only need to be performed once weekly, others, such as process chain monitoring, may need constant supervision. We recommend that you consider any opportunity to automate weekly tasks, such as by leveraging SAP Solution Manager to configure thresholds and send alerts by e-mail.

Reporting of performance or systematic issues for weekly checks should be handled the same way as for issues with daily checks. Any sensitive or critical checks should be executed first so that the technical team is knowledgeable and aware of the system's operational health.

Let's first look at weekly administration tasks.

8.1 Weekly Administration Tasks

The first technical resource in the office every Monday should conduct routine weekly administration checks manually as a quick health check of the system's weekend processing. If the system is monitored globally, 24/7, the weekly checks should be scheduled at a time when corrective action is most likely to be taken so that problems can be fixed immediately. Over time, thresholds for normal behavior will be apparent; these thresholds should be documented, and any abnormal behavior or deviations above or below normal should be noted and investigated.

In this section, we will review the following weekly administration tasks:

- Review EarlyWatch Alert
- Rebuild BWA indexes
- Clean up PSA and change logs
- Clean up application logs and trace files
- Execute housekeeping tasks
- Execute RSRV consistency checks

The majority of these administration tasks, such as rebuilding BWA indexes, cleaning up logs, and housekeeping tasks, ensure that your system is operationally healthy and stable enough to support future demand. The remaining administration tasks, such as reviewing the EarlyWatch Alert and executing Transaction RSRV consistency checks, arm you with information about the relative health of your system. Depending on the complexity of your system or the expected resolution, acting on the information provided may need to be handled in a more structured way, such as a project or simply planned in advance as part of the monthly, quarterly, or yearly administration tasks. The first weekly task should always be to check the EarlyWatch Alert for any information.

8.1.1 Reviewing SAP EarlyWatch Alert

The SAP EarlyWatch Alert (EWA) is a free preventive service included in the standard maintenance agreement with SAP. It is designed to help customers take rapid action before potential problems lead to unplanned downtime. The EWA provides customers with information about the stability and performance of their SAP systems, and it focuses on the following aspects:

- Server analysis
- Database analysis
- Configuration analysis
- Application analysis
- Workload analysis

The SAP EarlyWatch Alert is fully integrated with SAP Solution Manager, but it must be activated and scheduled. The EWA collects performance data by running background jobs in each SAP production system. The collected data is sent

from each satellite system to the central SAP Solution Manager system for processing and evaluation. The resulting reports provide historical trends of each system's performance to aid in the analysis of performance related issues, as seen in Figure 8.1.

2 Performance Indicators

The following table shows the relevant performance indicators in various system areas.

Area	Indicators	Value	Trend
System Performance	Active Users	785	↘
	Avg. Response Time in Dialog Task	1253 ms	↘
	Max. Dialog Steps per Hour	5351	↘
	Avg. Response Time at Peak Dialog Hour	1054 ms	↗
	Avg. Availability per Week	83 %	→
	Average Response Time in RFC Task	6116 ms	↗
	Max. number of RFCs per hour	64436	↘
	Avg. RFC response time at peak work hour	6129 ms	↗
Hardware Capacity	Max. CPU Utilization on DB Server	41 %	↘
	Max. CPU Utilization on Appl. Server	60 %	→
Database Performance	Avg. DB Request Time in Dialog Task	783 ms	↗
	Avg. DB Request Time in Update Task	39 ms	↗
	Average DB time for RFC	760 ms	→
Database Space Management	DB Size	34626.60 GB	→
	DB Growth Last Month	1713.12- GB	→

Figure 8.1 Historical Trends Shown in Performance Indicators of the EWA

Recommendation

Further analysis services are based on the resulting EWA for each satellite system, such as EarlyWatch Alert for Solutions and Service Level Reporting. See SAP Note 1040343 for more information on EarlyWatch Alert for Solutions.

After the initial configuration and activation in SAP Solution Manager, the first EWA session is automatically created for each system marked for monitoring. By default, all subsequent EWA sessions are scheduled once a week, on Mondays. If the overall rating of the EWA is red, as can be seen by item ❶ in Figure 8.2, the service results are automatically sent to SAP Support. If the overall rating is yellow ❷ or green, results are sent to SAP Support only once every four weeks. It is also possible to have the reports e-mailed automatically to specific e-mail addresses.

If the EWA issues a red or yellow rating, the reasons for the rating are specified right below the rating in the alert overview (see Figure 8.3). More information is provided in the subsequent sections where the alerts were evaluated. Investigate the root cause of the issues and take corrective action.

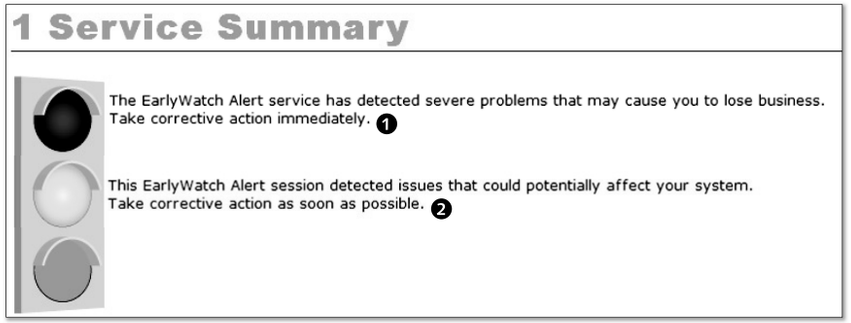


Figure 8.2 Red and Yellow Rating Messages in the EWA

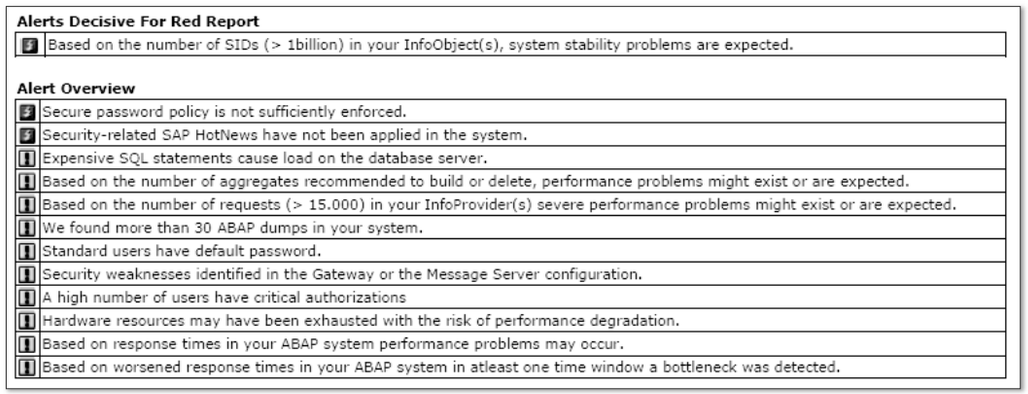


Figure 8.3 The Alert Overview's Justification for the EWA Report Rating

In our experience, many customers ignore EWA reports because the rating does not reflect a specific business need or use case. For example, if the report is always red because of a known and accepted issue, the monitoring team becomes falsely accustomed to the red rating and does not recognize when critical issues do indeed arise. This is a common behavior, but it is extremely reckless! In many cases, the thresholds for specific alerts and their ratings can be adjusted in SAP Solution Manager. If the issue cannot be resolved or is deemed an accepted risk, adjust the rating so that the monitoring team is not lulled into inaction.

In SAP BW systems, the EWA provides a section that analyzes the design, administration, and management of BW-specific areas (see Section 16 in Table 8.1, which lists all the EWA sections and subsections for an SAP BW 7.30 system). The BW Checks section has three subsections:

- **BW Administration and Design**
  This subsection reports important KPIs specific for SAP BW; identifies the largest SAP BW objects; analyzes the number and types of each InfoProvider; analyzes configuration and design settings that could impact performance; analyzes partitioning, aggregates, and BWA; identifies important SAP Notes that should be applied; reviews number-range buffering; analyzes SAP BW statistics; and reviews SAP BW-specific workloads.
- **BW Reporting and Planning**
  This subsection analyzes SAP BW runtime statistics and frontend distribution, checks query profiles, identifies the most popular queries and poorest-performing queries, analyzes query definitions, and analyzes OLAP cache usage.
- **BW Warehouse Management**
  This subsection analyzes dataload statistics; identifies top DTPs, largest requests, and top InfoProviders by load requests and upload volume; and reviews process chain and change run performance.

Section	Title	Subsections
1	Service Summary	▸ Alert Overview ▸ Check Overview
2	Performance Indicators	▸ Performance KPIs
3	Landscape Overview	▸ Products and Components ▸ Servers ▸ Hardware Configuration
4	Service Preparation and Service Data Quality	▸ Service Data Control Center ▸ ST-PI and ST-A/PI Plug-ins ▸ Service Preparation Check ▸ Performance DB ▸ Landscape Service Information ▸ Hardware Capacity Data
5	Software Configuration	▸ Application Release Maintenance ▸ ABAP SP Maintenance ▸ Java SP Maintenance ▸ Database Maintenance ▸ Operating System Maintenance ▸ SAP Kernel Release

Table 8.1 EWA Sections for an SAP BW 7.30 System on Oracle

Section	Title	Subsections
6	Hardware Capacity	▸ Overview System
7	Workload Overview	▸ Workload by Users ▸ Workload by Task Types ▸ Top Applications
8	Performance Overview	▸ Performance Evaluation
9	Trend Analysis	▸ Response Time Trends ▸ Application Profile
10	SAP System Operating	▸ Update Errors ▸ Program Errors (ABAP Dumps)
11	Security	▸ SAP Security Notes ▸ Default Passwords—Standard Users ▸ Password Policy ▸ Gateway and Message Server Security ▸ Users with Critical Authorizations
12	Software Change Management	▸ SAP NetWeaver AS ABAP ▸ SAP NetWeaver AS JAVA
13	Database Performance	▸ Load per User ▸ I/O Performance ▸ Database Parameters
14	Database Administration	▸ Database Growth ▸ Database Release
15	Java System Data	▸ Java VM Heap Size ▸ Availability of Performance Data
16	BW Checks	▸ BW Administration and Design ▸ BW Reporting and Planning ▸ BW Warehouse Management
17	Expensive SQL Statements	▸ Analysis of DB SQL Cache

Table 8.1 EWA Sections for an SAP BW 7.30 System on Oracle (Cont.)

Section	Title	Subsections
18	Trend Analysis	▸ System Activity ▸ System Operation and Hardware Capacity
19	Appendix	▸ Analysis of Aggregates

Table 8.1 EWA Sections for an SAP BW 7.30 System on Oracle (Cont.)

The EWA is an essential tool for any technical team supporting and monitoring an SAP BW system. It is always the first document that should be requested to assess the status of an existing system because it provides an overview supported by the exact details of every facet of the system, and it usually provides specific clues to solve performance issues and preserve operational health.

By routinely running, monitoring, and acting on the EWA, you can increase system stability and performance for your entire solution landscape. For this reason, we strongly recommend activating and monitoring the EWA for all productive systems.

8.1.2 Rebuilding BWA Indexes

You should routinely monitor BWA indexes to ensure that they are using the appropriate amount of BWA memory. Over time, BWA indexes can grow and become much larger than fact tables in the BI system. Any InfoCube that is fully reloaded after data is selectively deleted (not dropped) is a prime scenario in which the BWA index could double in size after every reload. Another slower growth scenario is an InfoCube, which is compressed with the elimination of zeroes. In both of these scenarios, the deleted or eliminated data still remains in the BWA index but is not used for reporting. As a result, the BWA index consumes more memory than necessary.

To identify these indexes, execute the BI and BWA table comparison check in Transaction RSRV on a periodic basis (see ❶ in Figure 8.4) and rebuild indexes for which there is a significant deviation. In addition to index growth, you should also adjust or rebuild indexes after changes are made to the relevant InfoProvider. If delta indexes are used, routinely merge them with the main index.

Regardless of an index size compared to the fact table, you would be prudent to rebuild the entire index routinely to ensure that memory is being released from the delta indexes. As seen in ❷ in Figure 8.4, Transaction RSRV has a BWA repair

utility; it can be scheduled routinely to delete and rebuild all BWA indexes. Due to potentially long run times and the impact on query performance, you should schedule this utility during a period of low system usage by end users when possible.

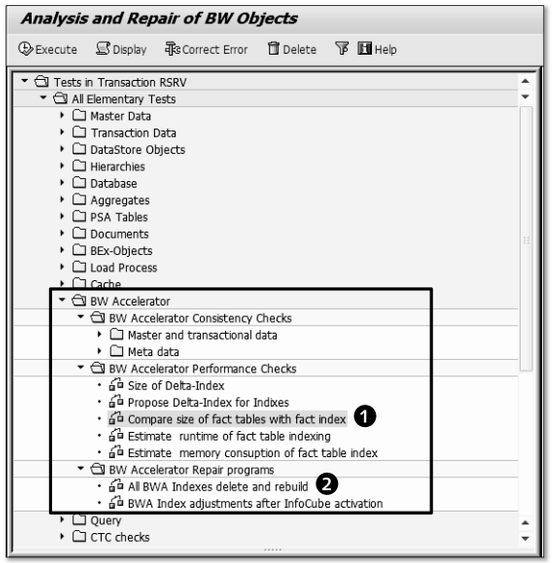


Figure 8.4 BWA Consistency Checks (Transaction RSRV)

However, if the rebuild time for all indexes exceeds any weekly available outage window, it is best to divide indexes into groups and schedule the rebuild of each group of indexes in an alternating manner. For example, in a scenario with two groups, A and B, the rebuilds of one group (A) can be scheduled during the first and second weeks of every month, and the rebuilds of the second group (B) can be scheduled during the third and fourth weeks. When necessary, individual index rebuilds can be deleted and re-created manually in Transaction RSDDB (see Figure 8.5).

Use Transaction RSDDBIAMON2 to access BWA and monitor the indexes on it. As indexes grow in size, you may find that you need to reorganize or redistribute indexes across the blades in the appliance. This capability can be launched from the BW Accelerator Monitor `RSDDBIAMON2` as displayed in ❶ of Figure 8.6. The same messages from the monitor can also be seen in the system checks in Transaction RSRV ❷.

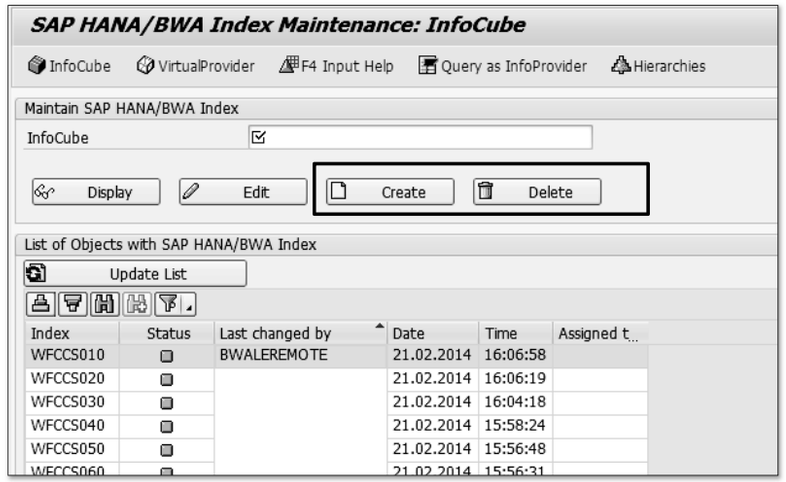


Figure 8.5 Manually Rebuild BWA Indexes (Transaction RSDDB)

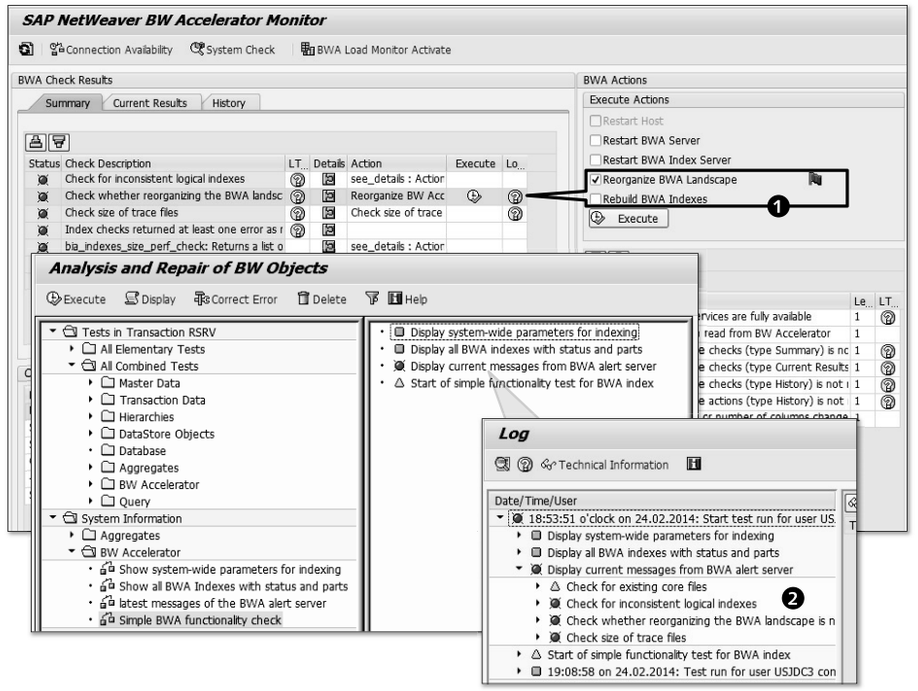


Figure 8.6 BWA System Information (Transactions RSRV and RSDDBIAMON2)

There are additional Transaction RSRV consistency checks available for BWA indexes, which should be used in conjunction with BWA alerts. BWA alerts can be configured via the TREX Admin tool on the BWA itself. These checks can also be executed directly from the index maintenance screen in Transaction RSDDB (see Figure 8.7). These consistency checks should be scheduled on a routine basis.

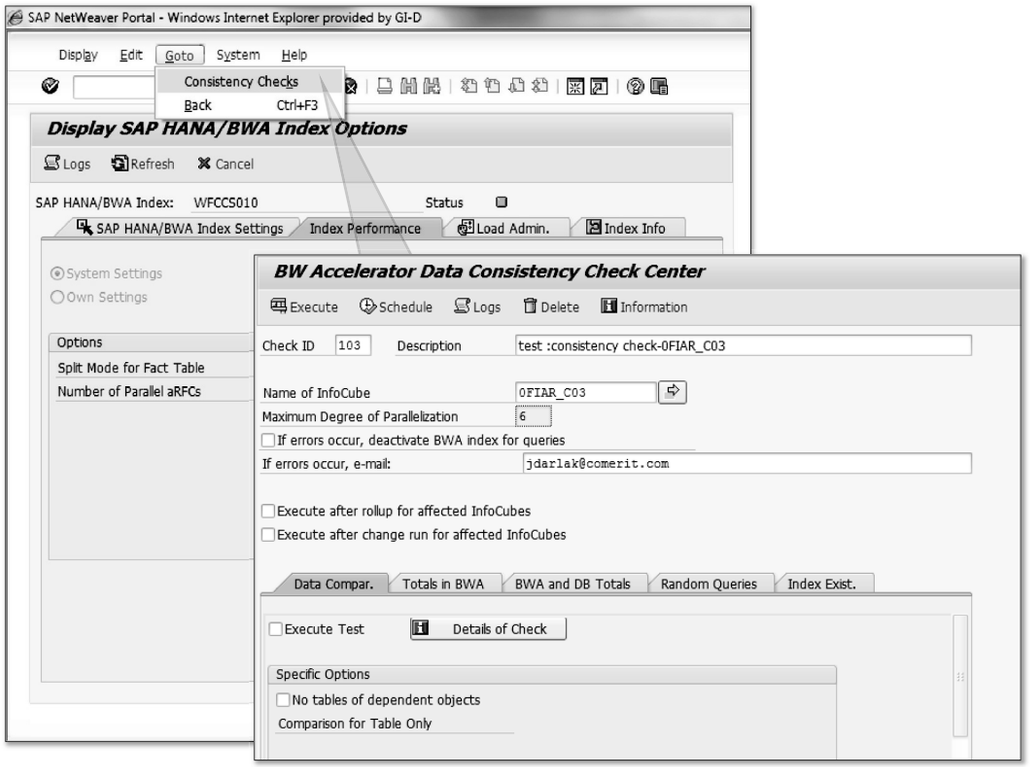


Figure 8.7 BWA Index Consistency Checks (Transaction RSDDB)

8.1.3 Cleaning PSA and Change Logs

During the load processing of large volumes of master and transaction data into SAP BW, there is usually a significant amount of storage consumed by temporary transaction data and system-related metadata. Routine housekeeping activities should remove unused, unwanted, and unneeded data. Regularly performing these activities ensures optimum utilization of system resources and increased system performance.

According to SAP Data Volume Management (DVM) Service, data deletion is one of the four recommended methodologies to check potential database growth. (The other methodologies are avoidance, summarization, and archiving.) Recall from Chapter 2 that one-third of the storage space in the average BW database is temporary data held in PSA and DSO change log tables.

Large PSA and DSO change log tables impact data load performance, increase the downtime for maintenance, and increase the cost of data storage. The size of the PSA and change logs can be explained by the fact that entries in these tables are never updated or overwritten. The only operations on these tables are inserts or deletions. For example, full loads on a periodic basis increase the size of the PSA table much faster than the actual data target, which is either overwritten in the case of a DSO or InfoObject, or dropped and reloaded in the case of an InfoCube. For delta loads, changes to previously extracted records are overwritten in DSOs and InfoObjects or compressed into a single record in InfoCubes (assuming that compression occurs routinely).

From a strategic perspective, the only reasons to retain entries in the PSA tables are to facilitate error resolution and to mitigate the need to re-extract data from sources that are either poor performing or delta capable. In the case of full loads, only the latest PSA entry should be retained, and all other entries can be deleted once the latest entry is successfully loaded into the PSA. For delta loads, a retention period for PSA data should be set based on the risk of data loss (i.e., data should not be deleted from the PSA before it is loaded to all subsequent data targets). In most cases, 15 days is more than sufficient to mitigate this risk. In many cases, a full repair load can be executed to recover any lost deltas without a significant performance impact, so this risk is usually applicable for delta DataSources that require the population of setup tables in the source system, such as LIS DataSources in the logistics cockpit, before full repairs can be carried out.

The deletion strategy for PSA tables should differentiate between master data and transaction data, full loads and delta loads, and DataSources and setup tables. Change logs should be treated like delta loads for transaction data. An additional consideration should be the periodicity of the data load from the source. For example, if a full InfoPackage is loaded monthly, the previous month's load should be retained in the PSA, so the retention period should be 31 days. Table 8.2 illustrates a sample deletion strategy for daily loads in a typical BW system.

Table Type	DataSource	Extraction Mode	Delete Entries Older Than…
PSA	Master data	Full	1 day
PSA	Master data	Delta	3 days
PSA	Transaction data	Full	1 day
PSA	Transaction data	Delta	8 days
PSA	Transaction data with setup table	Delta	15 days
Change log	Transaction data	Delta	8 days

Table 8.2 Sample Deletion Strategy for Daily Load PSA and Change Log Tables

The best way to delete data from the PSA and change log tables is by scheduling the relevant deletion process variants in the process chains after data has been loaded successfully (see Figure 8.8).

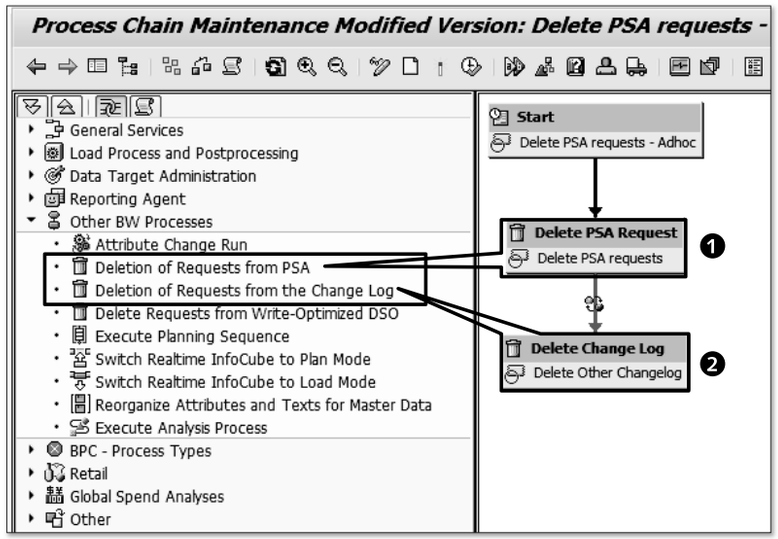


Figure 8.8 Process Chain Variants for Deleting Temporary Data (Transaction RSA11)

These process variants enable flexible deletion using selection criteria, and the parameter screens accept parameters with patterns on DataSource (for PSA tables) and DSOs (for change logs), as illustrated by ❶ in Figure 8.9. The number of days

to retain each PSA can be specified in the column identified by ❷, and the deletion can be limited to only those requests that are successfully updated ❸.

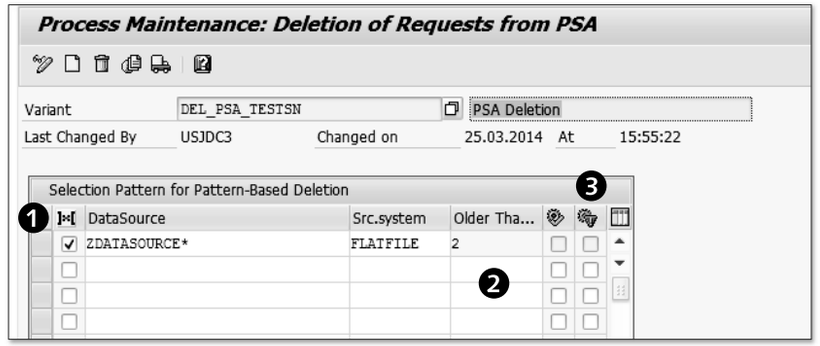


Figure 8.9 Process Variant for Deletion of Requests from the PSA

Even though the acceptance of patterns in the selection criteria simplifies maintenance of the deletion jobs, PSA and change log tables are still too easily omitted from cleansing. To combat this, custom programs published on SCN can help identify those tables with entries that are "unmanaged" or otherwise excluded from existing deletion variants.

For one-time use, PSA tables can be deleted using either program `RSPSADEL1` or function module `RSATREE_PSA_DELETE_BATCH`. We do *not* recommend that you use these programs routinely because PSA deletions should not be scheduled while data is being loaded. Therefore, it is best to schedule the relevant process variants directly in the process chains after data has been loaded. This helps mitigate the risk of database contention by ensuring that these conflicting jobs do not overlap.

Schedule a weekly PSA and change log deletion process chain to "catch" any data loads for which deletions are not scheduled as part of the load process chains. This helps mitigate the risk that a PSA table or change log could grow exponentially, thereby keeping the database growth in check and optimizing system resources.

8.1.4 Cleaning Application Logs

In all SAP systems, there are many log and trace tables that, if left untended, grow over time. Keeping these tables cleansed on a weekly basis ensures optimal performance of the ABAP system overall. This section will cover routine housekeeping jobs that should be executed routinely to minimize the impact of log and trace growth over time.

All application logs are written to table BALDAT—this table is usually one of the largest system tables in most systems. Ironically, many routine housekeeping jobs generate more application logs, which can consume more storage capacity than the housekeeping jobs free up. For example, every request that is deleted from the PSA or change log generates an application log. Therefore, it is important to clean application logs as part of every housekeeping effort. Before cleansing, consider the utility that application logs serve during root cause analysis; some application logs should be retained long enough to be helpful when investigating system events.

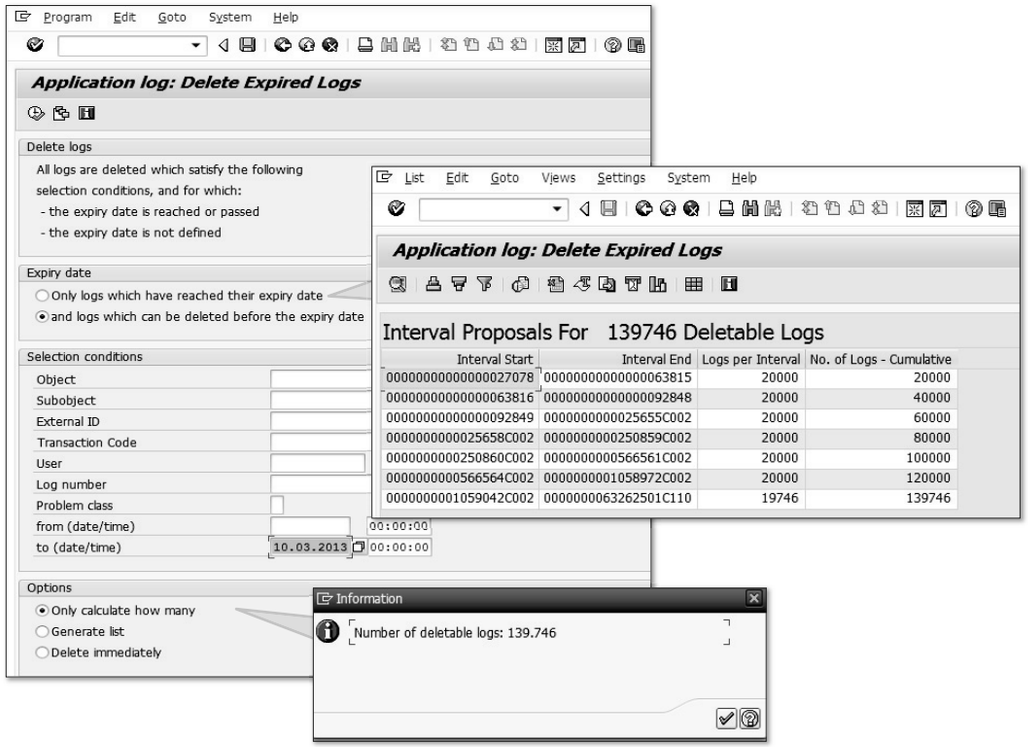


Figure 8.10 Deletion of Expired Logs (Transaction SLG2)

Regardless of their potential utility, at some point, application logs should be cleansed to reclaim that storage capacity and improve the overall performance of administering the system. Remember—the fewer logs in the table, the quicker it is to search for them.

To cleanup application logs, execute Transaction SLG2 or run report SBAL_DELETE, enter an appropriate date from which to cleanse logs, and select the checkbox to delete logs that can be deleted before their expiry date, as shown in in Figure 8.10. Use the option ONLY CALCULATE HOW MANY to show how many logs can be deleted. Schedule report SBAL_DELETE to delete the application logs on a weekly basis.

8.1.5 Executing BW Housekeeping Task List

As covered in Chapter 1, the task list SAP_BW_HOUSEKEEPING is available for scheduling routine housekeeping tasks that should be executed weekly in all SAP BW systems. By default, the housekeeping task list contains tasks that do the following:

- Repair indexes on InfoCube fact table(s) at the Data Dictionary level
- Re-assign requests written into the incorrect PSA partition
- Ensure request consistencies throughout the PSA landscape
- Ensure that partitioned tables are correctly indexed for the PSA
- Verify DataSource segments' assignment to PSA
- Delete entries that are no longer required in table RSIXW
- Reorganize and delete bookmark IDs and view IDs
- Delete RSTT traces
- Delete BW statistical data
- Check BW metadata with regard to the DDIC
- Clear all OLAP Cache parameters

Note

The coding for the SAP_BW_HOUSEKEEPING task list can be implemented via SAP Note 1829728 for SAP BW 7.x systems.

You can execute and schedule the housekeeping task via Transaction STC01, as shown in ❶ in Figure 8.11. Some of the tasks in this task list require that parameters be entered; these tasks each contain a "text editing" icon in the parameter column, as illustrated by ❷.

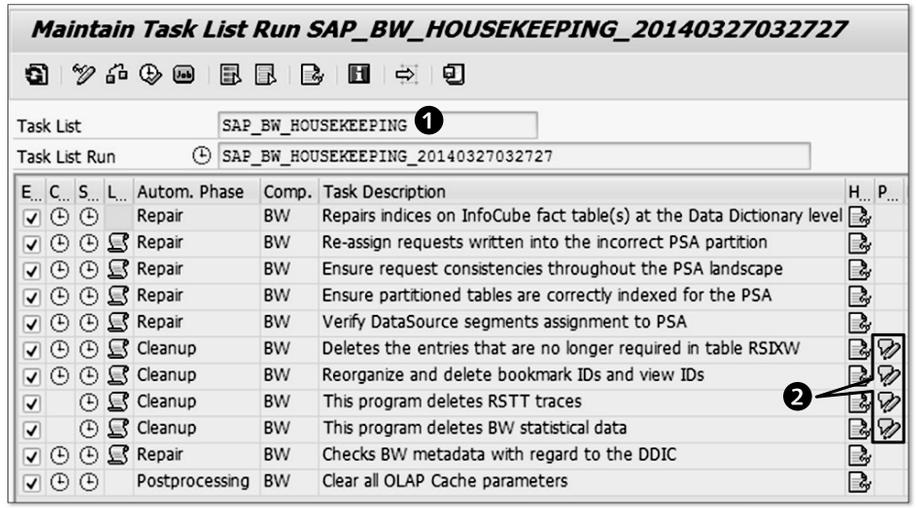


Figure 8.11 Task List SAP_BW_HOUSEKEEPING (Transaction STC01)

Enter the parameter values based on the data-retention strategy and save the entries as a variant so that the same values can be used when scheduling the task list. To check the status or logs of any current or previous task list execution, use Transaction STC02.

8.1.6 Executing Other Housekeeping Tasks

In addition to cleansing application logs and executing the SAP BW housekeeping task list, you should execute the jobs listed in Table 8.3 on a weekly basis. In most cases, the report can be scheduled, but the results should be monitored for exceptions or errors.

Report/Transaction	Administration Task
RSBATCH_DEL_MSG_PARM_DTPTEMP (use parameters `DEL_MSG` = 3, `DEL_PAR` = 3 and `DEL_DTP` = `X`)	Delete BW background management messages and parameters
RSSM_ERRORLOG_CLEANUP	Reorganize PSA error log
RSAR_PSA_CLEANUP_DEFINITION	Check and clean PSA tables

Table 8.3 Additional Housekeeping Jobs

Report/Transaction	Administration Task
RSB_ANALYZE_ERRORLOG	Analyze DTP error log
RSBM_ERRORLOG_DELETE	Delete inconsistent Error DTPs
RSPC_INSTANCE_CLEANUP	Delete old Process Chain logs
RSPC_LOG_DELETE	Delete old Process Chain logs
RS_FIND_JOBS_WITHOUT_VARIANT	Delete jobs without variant
RSBTCDEL2	Delete old job logs
RSSODFRE	Reorganize hidden folder Office Documents
RSBCS_REORG	Reorganize SAP Office/Business Workplace documents
RSTBPDEL	Check and clean up table change logs (DBTABLOG)
Transaction SM58	Check and delete old tRFC Queues
SAP_DROP_TMPTABLES	Delete temporary BW database tables
RSPO1041 (first run the consistency check from TemSe Data Storage in Transaction SP12)	Reorganize TemSe and Spool
RSTS0024	Delete orphaned job log
RSTT_TRACE_DELETE	Delete BW RSTT Traces
RSAN_UT_RESULT_DROP_RESULTS	Purge intermediary results from APD
RSBPSTDE	Delete statistic data from job runtime statistics
RSM13002	Delete update requests
RSARFCER	Delete old tRFC entries
SWNC_COLLECTOR_CLEAN_SWNCMONI	Delete old Transaction ST03N source system data
RSAR_PSA_NEWDS_MAPPING_CHECK	Check and repair inconsistent PSA request/tables
Transaction RSECADMIN	Delete authorization logs and disable log recording for all users

Table 8.3 Additional Housekeeping Jobs (Cont.)

As an alternative, create a custom task list in Transaction STC01 to include as many of these tasks as feasible, or enhance the default housekeeping task list.

8.1.7 Executing RSRV Consistency Checks

In Chapter 3, we covered elementary Transaction RSRV checks for master data and transaction data, which are perfect for investigating specific issues. Scheduling so many elementary tests for each InfoProvider would be a laborious burden and likely not worth the effort. Fortunately for you, SAP has provided combined tests (see Figure 8.12), which bundle many of the important elementary tests for ease of scheduling. These combined tests should be evaluated and scheduled for routine batch processing.

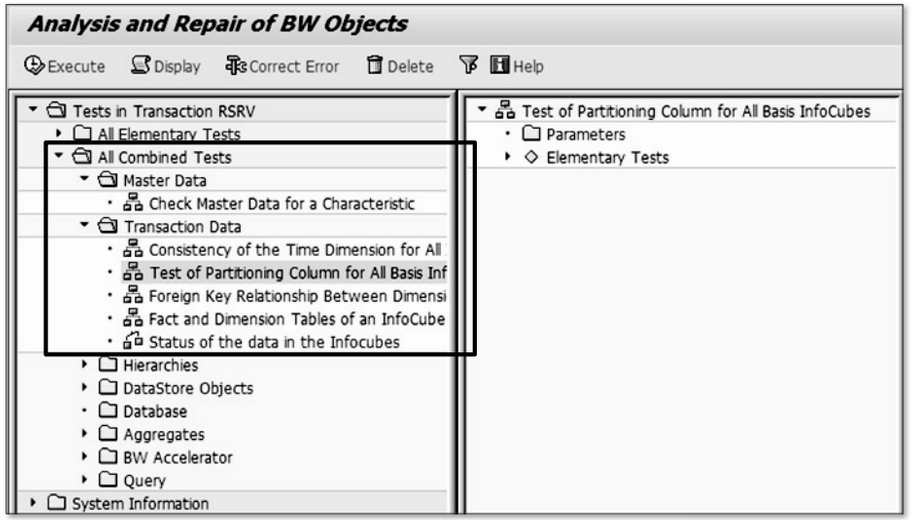


Figure 8.12 Combined Tests (Transaction RSRV)

Batch scheduling the combined tests proves valuable only if the logs are analyzed for errors or warnings. Therefore, you should analyze the logs using Transaction SLG1 for any combined tests that are deemed valuable to schedule on a weekly basis, as shown in Figure 8.13. The results of combined checks can also be monitored in the CCMS, as covered in Chapter 6.

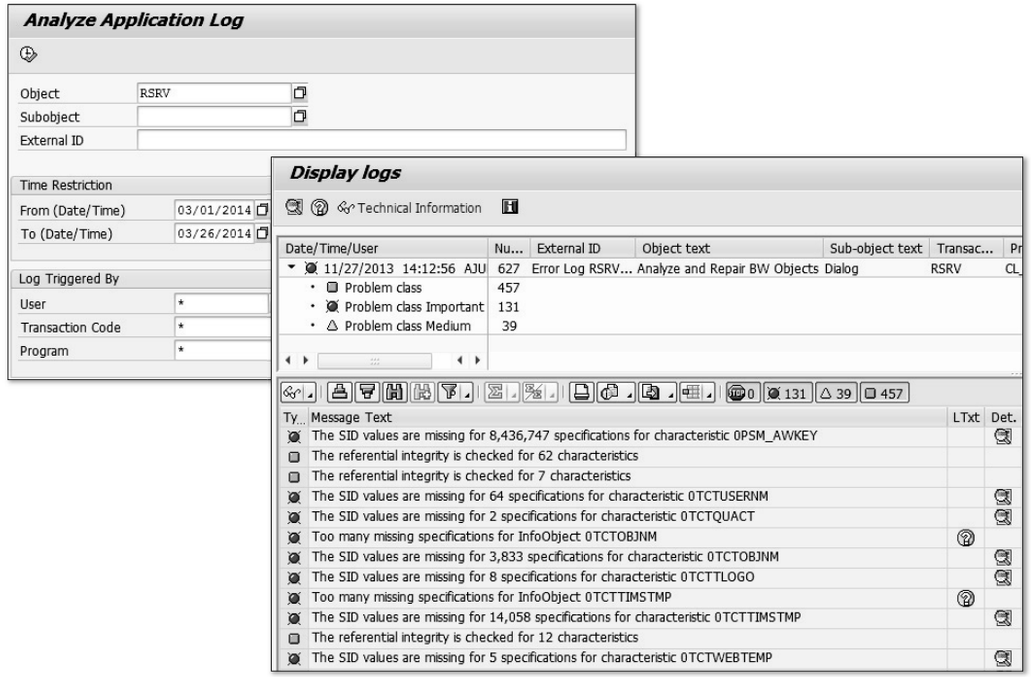


Figure 8.13 Viewing Application Logs (Transaction SLG1)

8.2 Weekly Performance Tuning

Let's take a look at tasks that are critical to maintaining optimal performance in the system for both end users who are running reports and support teams who are executing data loads and administering the data warehouse. We'll examine the following:

- Monitoring BI statistics
- Compressing InfoCubes
- Rebuilding DB indexes and statistics
- Monitoring cache usage

The first task is monitoring BI statistics.

8.2.1 Monitoring BI Statistics

SAP BW records runtime statistics for processes and events in SAP BEx, the Online Analytic Processor (OLAP), and warehouse management. The system records runtimes for statistical events by calculating the difference between the start and end times and subtracting the times for other events called from within each event. Technical BI Content delivers predefined SAP BW objects for analyzing statistics data.

The BW Administration Cockpit provides a central point of entry for monitoring the performance of SAP BW systems. Based on the technical BI Content, this enables runtime analysis and status of SAP BW objects and processes in the SAP Enterprise Portal or using SAP BusinessObjects Dashboards.

Note

For more information, see SAP Note 934848, "Collective note: (FAQ) BI Administration Cockpit)."

SAP BW statistics are recorded for the following areas:

- SAP BEx query runtimes (actions on the front end and in the Online Analytic Processor, including BW Integrated Planning)
- Runtimes of sub-processes in SAP HANA/BWA index maintenance
- Runtimes of sub-processes of analytic indexes
- Runtimes for performing planning functions
- Runtimes and status of data transfer and data processing in the Data Warehouse

Note

The system always records runtime and status statistics of data loading and data processing but does not, by default, record statistics for the DTPs. These are deactivated in the default setting and can be activated in table RSDDSTATOBJLEVEL. For more information, see SAP Note 966964.

Depending on the area, the system records statistics in tables RSDDSTAT* or UPC_STATISTIC*. For statistical analyses in warehouse management processes, technical BI Content uses tables RSMDATASTATE, RSMDATASTATE_EXT, RSDDSTATDTP, RSBKREQUEST, RSBSOURCEPROP, RSPCLOGCHAIN, and RSPCPROCESSLOG.

The data from these statistics tables is stored in technical BI Content InfoProviders, which are then assigned to one of the MultiProviders listed in Table 8.4. Technical BI Content queries and web templates provide ready-made views of the information for flexible analysis.

MultiProvider	Object Description
0TCT_MC01	Front-end and OLAP statistics (aggregated)
0TCT_MC02	Front-end and OLAP statistics (details)
0TCT_MC03	Data manager statistics (details)
0TCT_MC05	OLAP statistics: cache-type memory consumption
0TCT_MC06	OLAP statistics: cache vs. SHM
0TCT_MC07	OLAP statistics: query memory consumption
0TCT_MC11	BI object request status
0TCT_MC12	Process status
0TCT_MC14	Report availability status
0TCT_MC15	Data storages with inconsistent and incomplete data
0TCT_MC21	Process statistics
0TCT_MC22	DTP statistics
0TCT_MC23	InfoPackage statistics
0TCT_MC25	Database volume statistics
0TCT_MC31	BWA statistics: CPU consumption
0TCT_MC32	BWA statistics: InfoProvider memory consumption
0TCT_MCA1	Front-end and OLAP statistics (highly aggregated)
0TCT_MCWS	Workspace MultiProvider

Table 8.4 Technical BI Content MultiProviders

While analyzing the statistics data using the technical BI Content provided is mandatory, customers can and should define their own queries and web templates for specific analysis scenarios. See Figure 8.14 for an example of a simple query on

aggregated OLAP statistics showing the percentage of runtime by component for a rolling 24 months. This data is reported from InfoCube `0TCT_C01` and shows that DB Time has been steadily increasing over time. This indicates that database growth has been adversely impacting query performance, and steps should be taken to slow, stop, or reverse this trend. Potential solutions include compressing InfoCubes, rebuilding DB indexes, and tuning aggregates. If none of these solutions are satisfactory, consider installing BWA (which may be a regretful purchase) or migrating to SAP HANA.

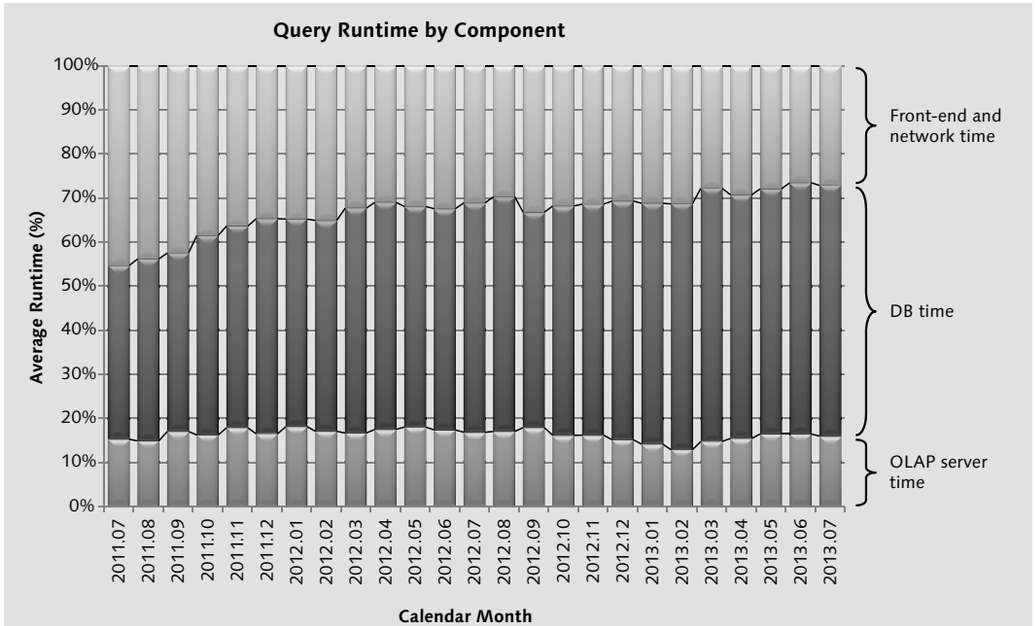


Figure 8.14 Front-End Query Statistics

Statistics queries and web reports should be executed and monitored for exceptions to average runtimes on a weekly basis. You should investigate and resolve any significant exceptions or deviations from normal.

8.2.2 Compressing InfoCubes

During compression, all records with the same key values across the data-package partitions in the F-fact table are aggregated into a single record in the E-fact table. When F-fact table requests are compressed, their respective data partitions are dropped from the F-fact table. In this way, the data volume of the InfoCube can be managed when a high number of delta records results in substantial growth. Compression provides performance benefits from a data loading perspective, as well as from a reporting perspective.

Ideally, you should execute compression after each data load, although it is not necessarily a critical path item. The more frequently an InfoCube is compressed, however, the quicker both queries against it and data loads to it execute. The only reason not to compress data is that the request needs to be dropped from the InfoCube and reloaded from the source. However, because data should always be staged in a DSO before feeding a cube, this reasoning is usually without merit.

Schedule a weekly compression process chain to compress all InfoCubes for which compression is not executed immediately after each data load. In most cases, compression with elimination of zeroes provides the most benefit by deleting all records from the E-fact table for which all key figures equal zero—as long as the zero-records are not required for business logic in reporting. Monitor this process chain exactly like any other process chain. If BWA indexes are also rebuilt weekly, ensure that you execute compression before rebuilding the index.

8.2.3 Rebuilding DB Indexes and Statistics

Like BWA indexes, database indexes can grow and become cumbersome over time. To combat this phenomenon, you should also rebuild DB indexes on a routine basis. While DB indexes can be rebuilt manually when managing InfoCubes ❶, as seen in Figure 8.15, the deletion and rebuilding of indexes should be scheduled via process chains on a weekly basis. You can delete and repair DB indexes in dialog mode or delete and re-create them in a batch.

The database statistics are used by the system to optimize both query and compression performance. Even when the InfoCube is indexed in BWA, it is important to keep the database statistics up to date. It is possible to schedule automatic recalculation of the database statistics after each data load in the process chains. At a minimum, we recommend that you update the statistics after every one million new records have been loaded into an InfoCube.

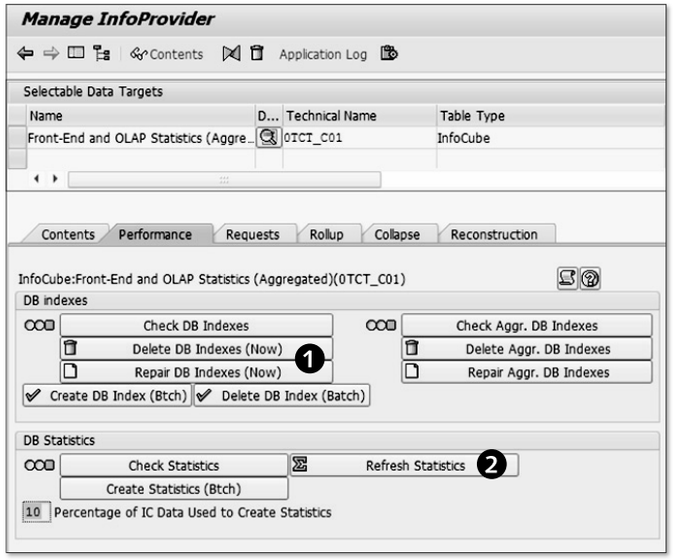


Figure 8.15 DB Index Rebuilding Using the InfoCube Manage Screen

The status of the database statistics can be checked from the InfoCube maintenance screen, as seen by ❷ in Figure 8.15. You can adjust the sample size as needed. The button Check Statistics confirms whether the statistics are up to date (green), are out of date (yellow), or do not exist (red). Use Recalculate Statistics to create or update statistics in the background.

Calculating statistics for InfoProviders should also be scheduled weekly via process chains or directly via an OS-level command for the database installed, where it may be possible to specify thresholds to limit the job to InfoProviders for which only a specific percentage (usually 5-20%) of the base data has changed.

8.2.4 Monitoring Cache Usage

In SAP BW systems, the size of the global cache depends on the shared memory buffer, which is defined by the profile parameter `rsdb/esm/buffersize_kb`. The global cache setting can be maintained in Transaction RSRCACHE and should not exceed 90% of the shared-memory buffer.

The cache monitor can be accessed via Transaction RSRT, as shown in Figure 8.16. Check the size of the global and local cache buffers and the number of objects in their consumption.

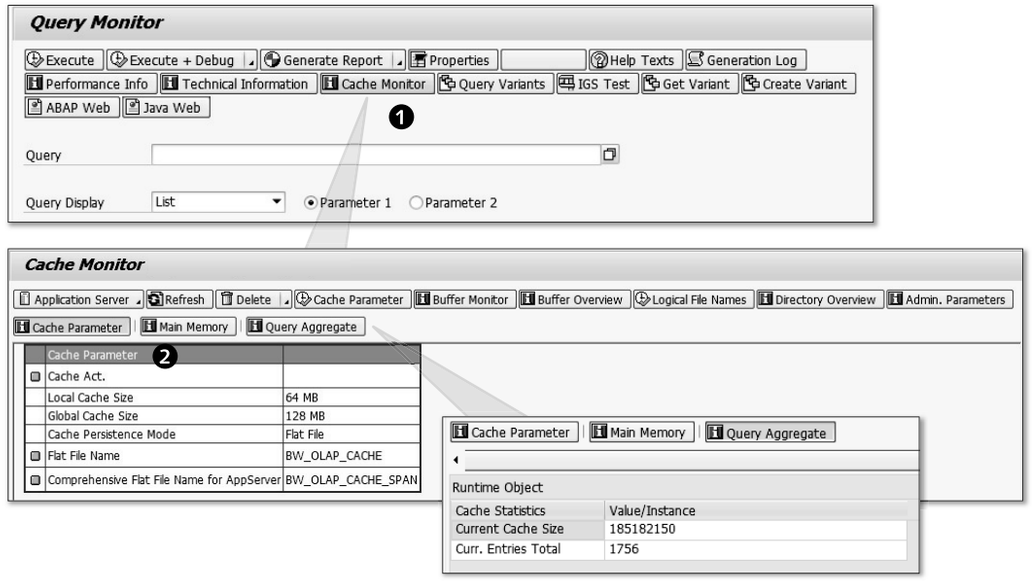


Figure 8.16 Access to the Cache Monitor (Transaction RSRT)

Alternatively, you can access the cache monitor directly via Transaction RSRCACHE, as shown in Figure 8.17. From the hierarchical display of the Main Memory overview, the cache buffer size of every query object can be checked. Identify extraordinarily large query object buffer sizes and remediate, if necessary.

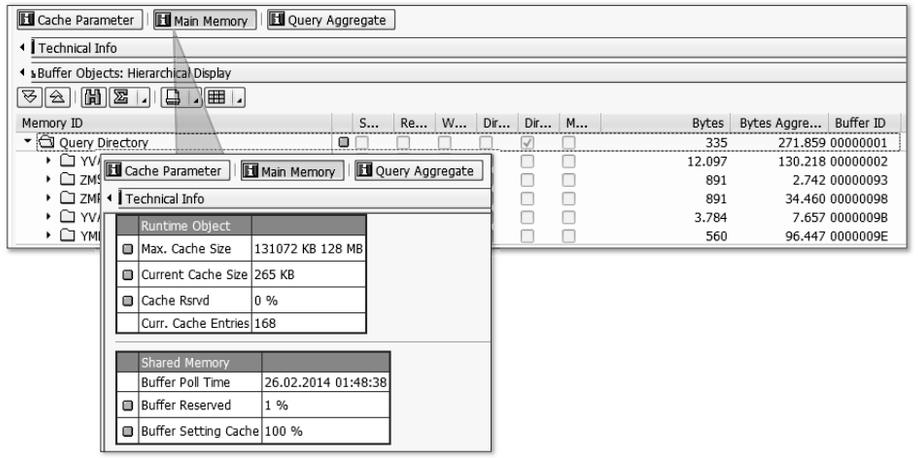


Figure 8.17 Main Memory Usage in the Cache Monitor (Transaction RSRCACHE)

Our next chapter covers monthly administration and performance-tuning tasks to keep your SAP BW system running smoothly.

Contents

PART III Administration Tasks in Your SAP BW System

PART IV Support for Your BI Initiatives

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

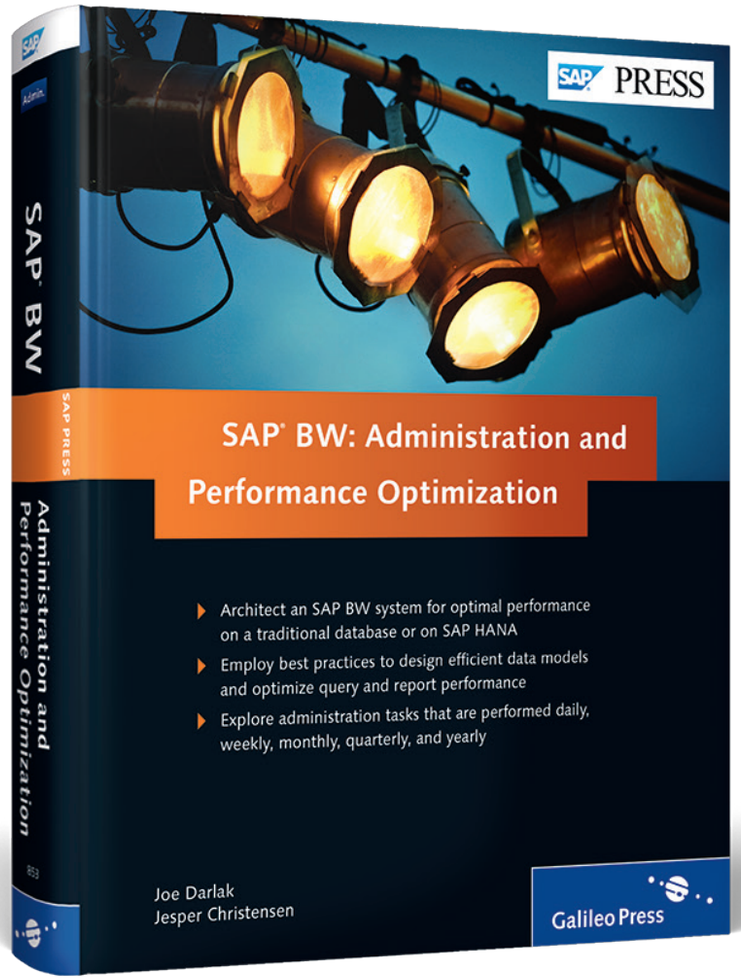
S

T

U

V

W



Joe Darlak, Jesper Christensen

SAP BW: Administration and Performance Optimization

652 Pages, 2014, €79.95
ISBN 978-1-59229-853-2

www.sap-press.de/3341

Joe Darlak, director at COMERIT, Inc., has been successfully implementing SAP BW at clients since 1998, including several large-scale, full-lifecycle global rollouts. He has significant experience increasing the capacity, reliability, and performance of BI systems, defining and implementing processes and controls to stabilize service and mitigate risk, and improving the competency and skill sets of implementation and client service teams. Recently, he has architected one of the largest SAP BW migrations to SAP HANA on record, and has been instrumental in defining best practices for SAP HANA at Fortune 100 clients. Over the past decade, he has presented at multiple conferences on various SAP BW performance tuning topics. He is a former principal consultant in the iAnalytics practice at PricewaterhouseCoopers and holds a masters of business administration from the McCombs School of Business at the University of Texas at Austin.

Jesper Christensen, director of Business Intelligence International at COMERIT, Inc., was part of the SAP BW Regional Implementation group at SAP that initially rolled out SAP Business Warehouse. He is an internationally recognized subject matter expert in business intelligence, a contributing author of international BI journals, and a frequent speaker at SAP and BI-related conferences. His leadership at COMERIT helped Tetra Pak win the Gartner Award of BI Excellence in 2009. Jesper has supported multi-year global projects as the lead architect with companies such as ExxonMobil, Chevron, Statoil, Ericsson, Tetra Pak, Shell, BHP Billiton, and McKesson, among others.