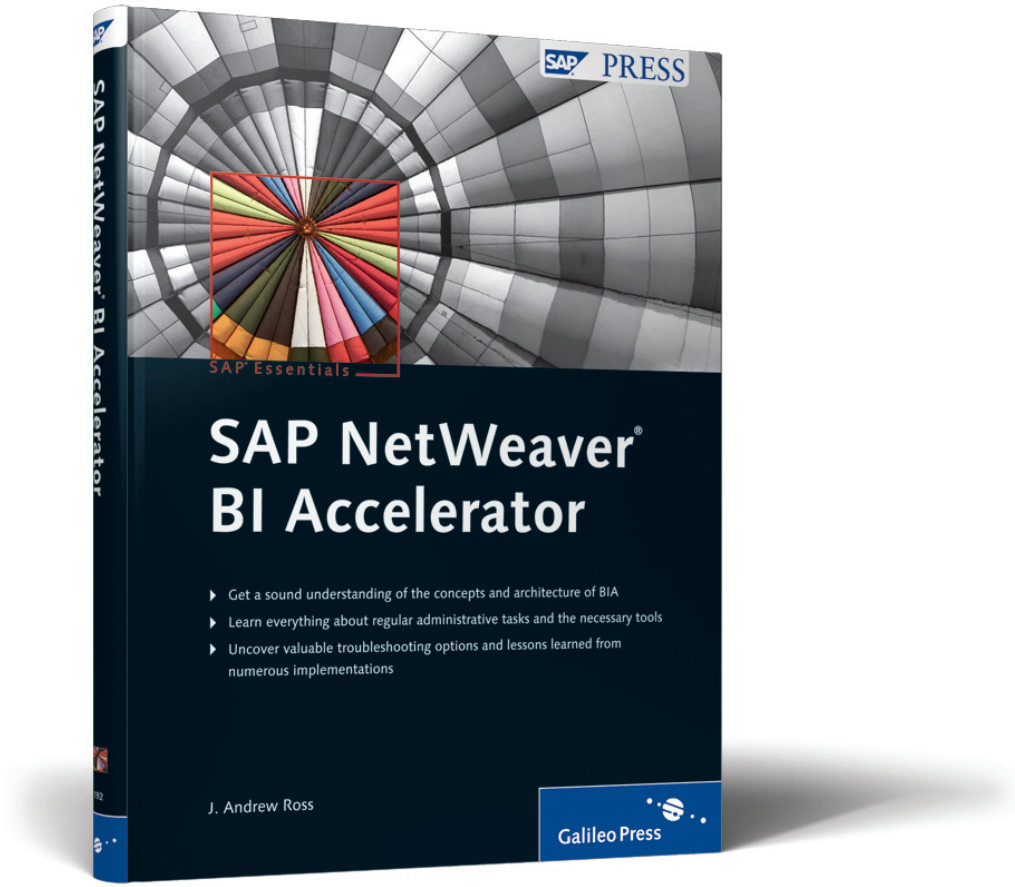
J. Andrew Ross

SAP® NetWeaver BI Accelerator



Galileo Press

Bonn • Boston

Contents

1 Introduction 9

2 Impact 17

2.1 The Platform Revolution 20
2.2 Business Impact 23
2.2.1 SAP NetWeaver 23
2.2.2 SAP NetWeaver Business Intelligence 25
2.2.3 The BI Accelerator Value Proposition 28
2.2.4 Setting the Right Expectations 35
2.3 Technical Impact 36
2.3.1 The Aggregation Challenge 37
2.3.2 Services and Appliances 38
2.3.3 An Inexpensive Solution 40
2.3.4 Ways to Respond to a Query 42
2.4 Summary 43

3 Architecture 45

3.1 The Appliance: TREX Engine Inside 47
3.1.1 Hardware Specification 48
3.1.2 TREX Client-Server Architecture 49
3.1.3 Index Server and BIA Engine 49
3.1.4 Name Server 51
3.1.5 RFC Server 52
3.1.6 Alert Server 53
3.1.7 Additional TREX Servers 54
3.1.8 Attached Storage 55
3.2 Availability 57
3.2.1 Tasks Requiring Planned Downtime 57
3.2.2 Redundancy and Backup Servers 58
3.2.3 Backup and Recovery 62
3.2.4 Disaster Tolerance 66
3.3 Scalability 68
3.3.1 Vertical Partitioning 69

3.3.2 Splitting Indexes 70
3.3.3 Cloning BIA to New Blades 71
3.3.4 Sizing and Size Restrictions 71
3.4 Minimized Administration 73
3.5 Summary 73

4 Administration 75

4.1 Main Administration Transactions 77
4.1.1 The BI Accelerator Monitor 77
4.1.2 Data Warehousing Workbench 84
4.1.3 BIA Index Maintenance Transaction 84
4.1.4 BIA Index Check and Repair Transaction 84
4.1.5 TREX Administration Transaction 87
4.2 Initial Tasks 88
4.2.1 Delivery and Installation 89
4.2.2 Maintaining the RFC Connection 91
4.2.3 Creating BIA Indexes 95
4.2.4 Configuring Alerts 111
4.2.5 Going Live 120
4.3 Regular Tasks 121
4.3.1 Index Maintenance 121
4.3.2 Monitoring Load 123
4.3.3 Creating Backups 125
4.4 Occasional Tasks 127
4.4.1 Cloning a BIA Instance 127
4.4.2 Updating a BIA Instance 129
4.4.3 Reorganizing the BIA Indexes 130
4.5 Summary 131

5 Advanced Administration 133

5.1 The TREX Standalone Administration Tool 133
5.2 Exceptional Tasks 137
5.2.1 Repairing the RFC Connection 138
5.2.2 Consistency Checks 144
5.2.3 Check Sets for BIA Indexes 154
5.2.4 Index Repair Programs 156
5.2.5 Impact of Metadata Changes on the Accelerator 157

5.2.6 Rebuilding Indexes 158
5.2.7 Performing a Recovery 161
5.3 Optimization Tasks 163
5.3.1 Index Landscape and Reorganization 163
5.3.2 Performance Checks 172
5.3.3 Delta Indexing 175
5.3.4 Statistics for BIA Index Maintenance 175
5.3.5 Indexing and Index Parameters 179
5.3.6 Tips and Tricks for High Performance 181
5.4 Collaborative Tasks 183
5.4.1 Setting Up the Service Connection 184
5.4.2 Landscape Overview 186
5.4.3 Tracing 188
5.5 Summary 199

6 Success Stories 201

6.1 Scalability: Project Jupiter 201
6.2 Colgate-Palmolive Company 205
6.3 Bayer AG 209
6.4 German Defense Forces 210
6.5 T-Mobile UK 211
6.6 Rabobank Group 212
6.7 Kimberly-Clark Corporation 213
6.8 WACKER Group 215
6.9 Summary 216

7 Technical Details 219

7.1 Vertical Decomposition 219
7.2 Parallelization 221
7.3 Smart Compression 223
7.4 SAP NetWeaver BI InfoCubes 227
7.5 BIA Indexes 229
7.5.1 TREX Metamodel 229
7.5.2 BIA Index Metadata 231
7.6 Query Execution 233
7.6.1 Query Plans 233
7.6.2 Plan Operations 235

7.6.3 Multiquery Optimization 235
7.7 Deploying the Accelerator 236
7.8 Summary 238

8 References 239

A Glossary 243

B The Author 253

Index 255

5 Advanced Administration

The administration tasks described so far have been part of the required skill set for any administrator of an SAP NetWeaver BI system with a BI accelerator attached. The tasks described in this section are more advanced and require additional skill to perform safely.

For users with sufficient experience, the TREX standalone administration tool offers a wealth of opportunities for deeper technical interaction with the TREX software within the accelerator. If a company requires SAP help to manage issues with its accelerator installation, an administrator can grant SAP service engineers remote access through the company firewall to the accelerator blades and thus allow them to use the TREX standalone administration tool over the Internet.

Section 5.1 introduces the TREX standalone administration tool. Section 5.2 describes some exceptional tasks such as repairing the connection to the BI system, checking and repairing the indexes, and performing a recovery. Section 5.3 describes some optimization tasks, including landscape reorganization, performance checks, and adjusting indexing parameters. Section 5.4 describes some tasks related to any necessary collaboration with SAP service experts, such as setting up a service connection, checking the BIA landscape, and creating trace files. Section 5.5 summarizes the chapter.

5.1 The TREX Standalone Administration Tool

For fully detailed interaction with the TREX engine at the heart of the SAP NetWeaver BI Accelerator, there is no better tool than the TREX standalone administration tool. This is a graphical tool supporting point-and-click and drag-and-drop interaction. Most administrative tasks can be handled through the SAP Transactions RSDDBIAMON and TREXADMIN, but some more advanced tasks require use of the TREX standalone administration tool. This tool was built by and for TREX developers, to enable them to work more easily with their own growing creation, and for this reason it offers powerful features that can be dangerous in

the hands of a beginner. The tool does not include the safeguards that a beginner might expect, and therefore it should not be made accessible to anyone who is not qualified to use it safely.

The TREX standalone administration tool supports a huge range of interactions and can display enough system information to enable technical experts to do whatever may be necessary to resolve any TREX issues that can arise in a productive BIA installation. If remote access to the tool is granted, SAP service engineers can use it to perform any troubleshooting task in TREX.

To start the TREX standalone administration tool:

1. Open a command-line window on one of the BIA hosts.
2. Change to user `<sid>adm`.
3. Go to directory `/usr/sap/<SAPSID>/TRX<NN>/`.
4. Execute command `TREXAdmin.sh`.

The default host on which to start the tool is the BIA blade hosting the active master name server. This is normally the first blade numerically in the set of BIA blades because the installation automatically assigns consecutive numbers to the blades, and the default master starts on the first blade. However, the tool can be started from any blade in the landscape and display the same information.

The TREX standalone administration tool opens at the view displayed when it was last closed. However, the natural view to display first is the Summary view (see Figure 5.1). The Status line above the Status Details panel tells you what you need to know first. In this case, all the alert icons are green (square). Above the Status line are some version and platform details, and the Status Details panel displays the more major check results such as memory consumption, number and total size of indexes, and whether traces are on or off.

After the Summary view, the first view to check in the TREX administration tool is perhaps the Services view (see Figure 5.2). All the main services on all the hosts in the landscape are listed. In this case, there are four blades in the landscape, and on each blade, there are name server, index server, RFC server, and "other" (alert server) services running. The green (square) icons down the left side show that all the services have status OK. The bars toward the center show that at the time of this screen capture, there is very little CPU activity or memory consumption on the blades.

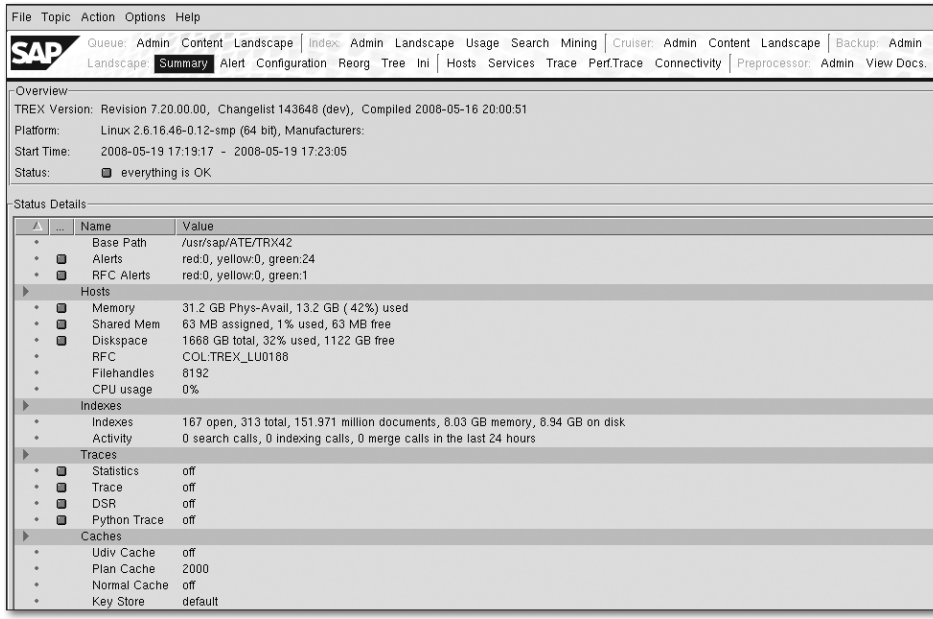


Figure 5.1 TREX Admin Tool, Summary

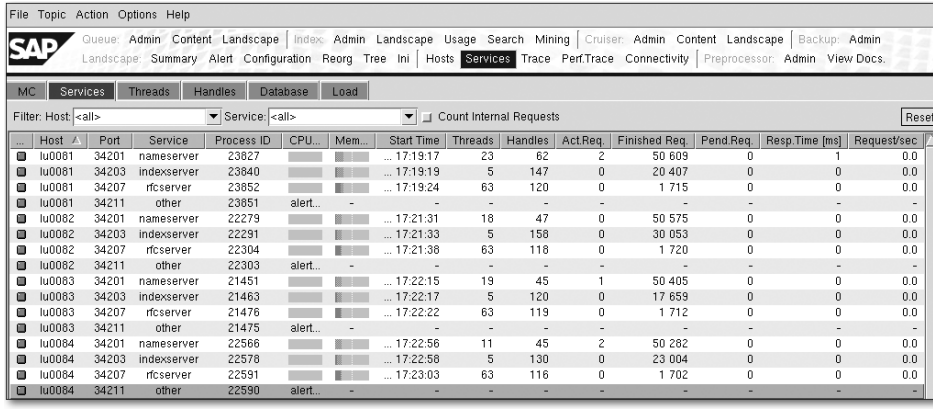


Figure 5.2 TREX Admin Tool, Services

To configure the BIA landscape, choose the CONFIGURATION view (see Figure 5.3).

The SCENARIO panel at the upper left enables you to select usage of backup index servers (the other options are inapplicable for the accelerator).

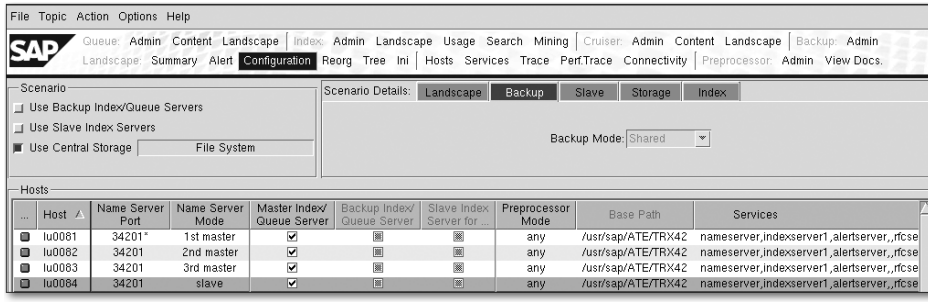


Figure 5.3 TREX Admin Tool, Configuration

In the SCENARIO DETAILS panel toward the right, the BACKUP tab offers a drop-down menu for selecting a backup mode. The available modes are:

- SHARED (one backup for all masters)
- DEDICATED (one backup for each master)
- MULTIPLEXED (one backup for several masters)
- MUTUAL (each master is backup for another master)

The INDEX tab enables you to change the size threshold for splitting a large BIA index.

Below the tabs, the HOSTS panel includes more detailed information and indicates with an asterisk which blade hosts the currently active master name server. Use of the word "slave" is an inheritance from the text search world and has no meaning for the accelerator.

Instead of configuring the alert server via Transaction TREXADMIN, you can do so from the TREX standalone administration tool by selecting the ALERTS view, and then choosing ALERT SERVER CONFIGURATION (see Figure 5.4). Both these alternatives offer the same functional capabilities.

In the top panel of the configuration dialog box, you can enter an e-mail address to which alerts above the given threshold (trace level) should be sent. In the middle panel, you can override the default set of checks that run regularly, at times specified by the code in the middle column, and you can change how often they run by editing the displayed cron job specification (for which there is a nearby CRONTAB HELP button in case you forget the format). In the bottom panel, you can change the selection of individual checks that run in the selected check set.

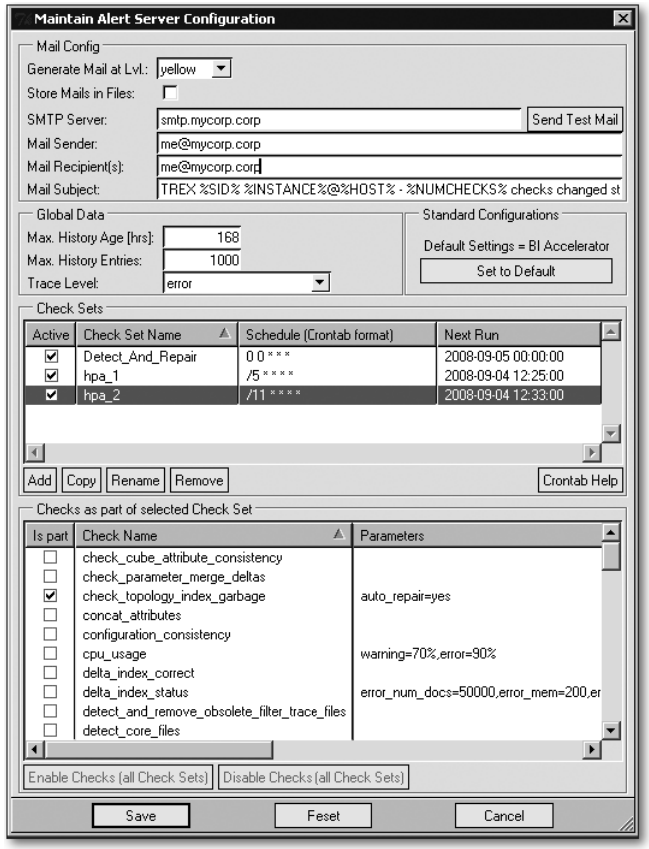


Figure 5.4 TREX Admin Tool, Alert Server Configuration

You should ensure that the BIA blades can reach the SMTP server you specify as the destination for e-mail alert messages. Each TREX installation runs its own alert server, and the results from the checks run by each of the alert server processes are consolidated on the blade that hosts the active master name server so the active master sends the alert messages, but this can only succeed if TREX can access the SMTP server.

5.2 Exceptional Tasks

As an administrator, you should be able to handle not only the tasks that arise during routine operation of the SAP NetWeaver BI Accelerator, but also certain

tasks you would need to perform only in exceptional circumstances. Here we review some of these tasks.

5.2.1 Repairing the RFC Connection

If the RFC connection between the BIA blades and the SAP NetWeaver BI system is broken for any reason, the situation can be diagnosed and in most cases repaired using the TREX administration tool. The tool offers full support for creating and editing RFC destinations, pinging hosts to locate network problems, reporting in detail on any issues that arise, and automatically repairing broken connections.

Alerts are color-coded to highlight any that require manual intervention. In general, red alerts indicate an error status that the tool cannot repair by itself, while yellow alerts indicate issues that the tool can resolve automatically. The red alerts indicate situations that require human action. The yellow alerts will be repaired in the next check run if the RFC check in the alert server is activated (where the check runs by default every 5 minutes).

You can create a new RFC destination in the SAP NetWeaver BI system for the accelerator by using Transaction SM59, as described in Section 4.2.2. The connection type is TCP/IP, and TREX runs as a registered server program. The program ID is unimportant because the accelerator overwrites it automatically with a correct ID.

To ensure that you receive sufficient warning of any RFC problems that may arise, you can check that the TREX alert server is configured to perform regular automatic RFC connection checks and to send you an e-mail if a problem is sufficiently serious.

To do so, in the TREX standalone administration tool Alert view, choose Alert Server Configuration to display the dialog box shown earlier in Figure 5.4. There you can check that your e-mail contact information is entered correctly, and you can review the checks that have been selected. To receive RFC warnings, you should ensure that in one of the selected check sets, the check `rfc_connection` has been flagged.

To handle matters related to the RFC communication between the accelerator and the BI system, choose the Connectivity view in the TREX standalone administration tool, as shown in Figure 5.5.

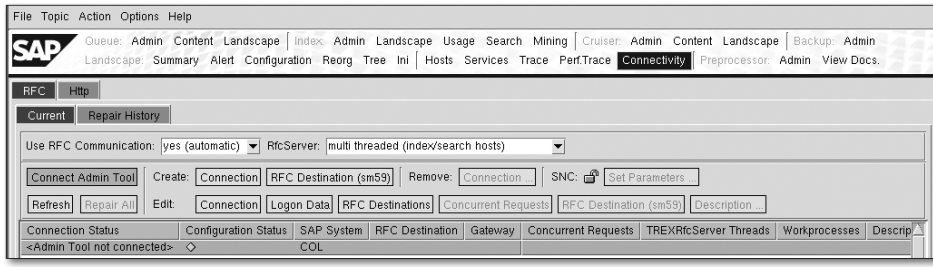


Figure 5.5 TREX Admin Tool, Connectivity View

The view opens to show the RFC tab with its CURRENT tab showing. The tool indicates any recommended next action by highlighting the relevant button with a different color. In the case shown, the administration tool has not yet connected itself to the BI system, as indicated by the gray (diamond) icon, and the recommended next action is to choose the CONNECT ADMIN TOOL button.

If the TREX standalone administration tool attempts to connect itself to the BI system, but the required connection data is missing in the accelerator, the tool will show a message highlighted in yellow as shown in Figure 5.6.

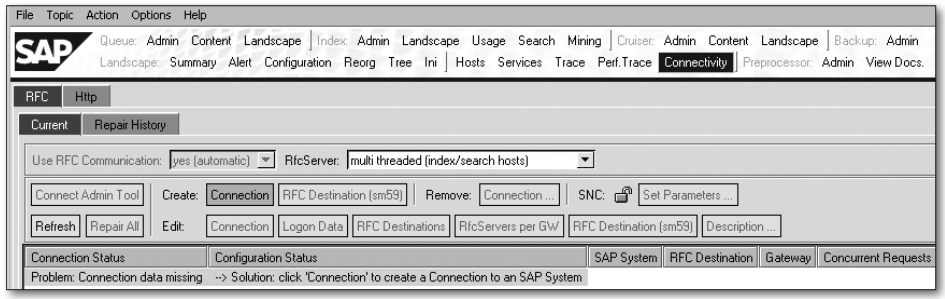


Figure 5.6 TREX Admin Tool, Connection Data Missing

The recommended next action is to create a connection. This is indicated by the highlighting of the button you should choose next to create a connection. When you do so, the dialog box shown in Figure 5.7 appears.

This dialog box enables you to create a connection to the specified SAP system on the assumption that a correctly configured RFC destination exists in that system. Within TREX, the connection data you enter in the dialog box is stored in the *saprfc.ini* file, and the logon data you enter is encrypted and stored in *topology.ini*.

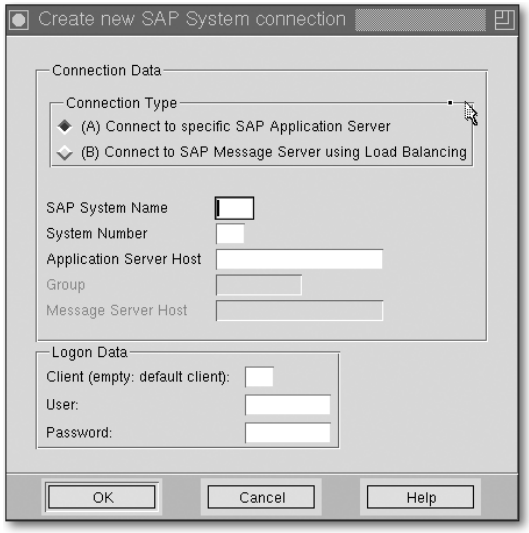


Figure 5.7 TREX Admin Tool, Create New SAP System Connection

When you set up an RFC destination, you need to specify whether your accelerator connects to your SAP NetWeaver BI system landscape via local gateways or a central gateway (see Figure 5.8). Each TREX instance has one RFC server process, which spawns as many threads as required.

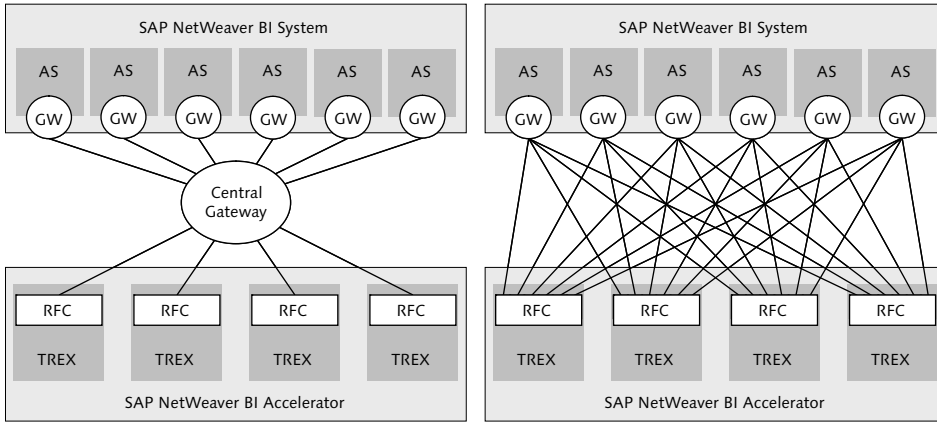


Figure 5.8 Central and Local SAP Gateways

A central gateway between the local gateways on the application servers and the RFC servers in TREX appears to offer simpler traffic flow because each node at

either end communicates with only a single node represented by the central gateway but has the disadvantage that this central node is a single point of failure and can be a bottleneck in a heavy load scenario. For this reason, we recommend the use of local gateways for communication with the accelerator.

In the local gateway configuration, each application server communicates directly through an RFC server with every TREX instance in the BIA landscape, and vice versa. As Figure 5.8 shows, this can give rise to a large number of RFC server threads on each BIA host, but no extra administration is required. The administrator simply leaves the gateway option fields blank when setting up the RFC destination. TREX then starts as many RFC server threads as needed.

If a suitable RFC destination does not yet exist in the target SAP system, you can create one, either in Transaction SM59 or in the TREX standalone administration tool.

If TREX does not find a suitable RFC destination in the target SAP system, the tool prompts you (with highlighted buttons) either to enter into TREX the information about an existing RFC destination that is defined in a connected SAP system (see Figure 5.9) and store the information in the TREX *topology.ini* file or to create a new RFC destination in a connected SAP system (see Figure 5.10). If you create a new destination, you do not need to enter a program ID because TREX generates one automatically for use at both ends of the connection. Again, the information is stored locally in the TREX *topology.ini* file. As soon as the destination has been created, a "created successfully" information box appears.

Creating a new RFC destination from TREX as shown in Figure 5.10 is equivalent to doing so in the SAP system using Transaction SM59. Both ways of setting up an RFC destination require the same information.

Connecting the TREX administration tool to the SAP system is just the first step. Once the tool has successfully connected, it determines whether all the BIA blade server hosts can communicate correctly with the SAP system. If not, for example, because an application server has been removed from the landscape or added to it, the tool may display information similar to that shown in Figure 5.11, with error details highlighted in red, other unresolved issues in yellow, and a yellow (warning triangle) status icon for the affected system. The problems highlighted in yellow can be solved by the automatic repair capabilities built into the administration tool. In this case, the tool prompts you with highlighting to choose the Repair All button. Because the RFC check runs automatically in the background,

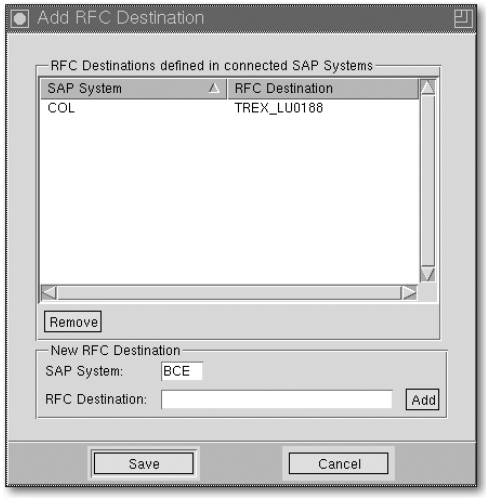


Figure 5.9 TREX Admin Tool, Add RFC Destination

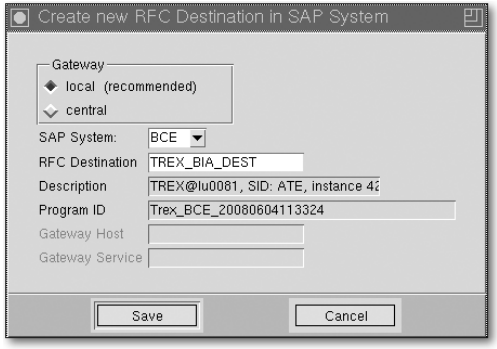


Figure 5.10 TREX Admin Tool, Create New RFC Destination

you can simply wait to let the repair run, and check that it has done by clicking Refresh as often as necessary, but clicking the highlighted Repair All button can speed up the repair if you are in a hurry.

Fortunately, the problems shown here were not too serious, and TREX solved them all in seconds, as shown in Figure 5.12. Now the tool shows a green (square) configuration status icon and highlights the connection details in green to indicate that all is well. Generally, the tool offers clear visual feedback in this way to ensure that the administrator knows as exactly as possible what to do next.

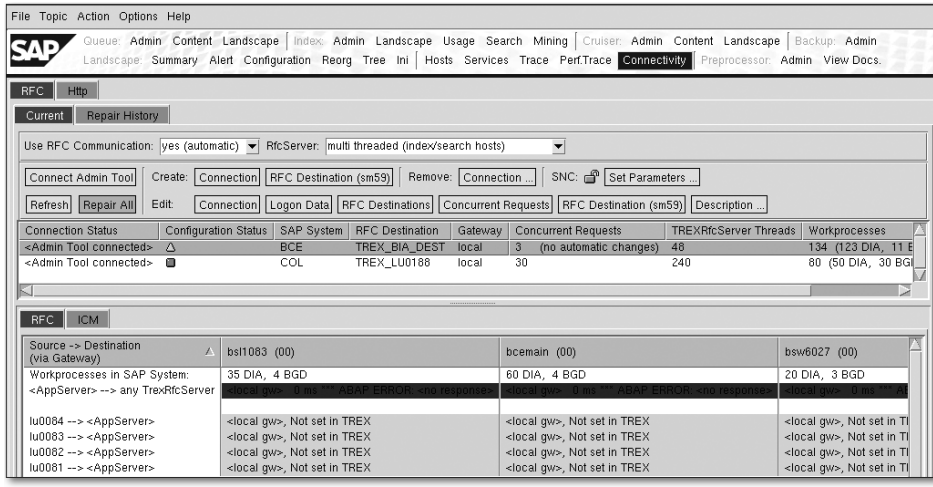


Figure 5.11 TREX Admin Tool, Repair Connectivity

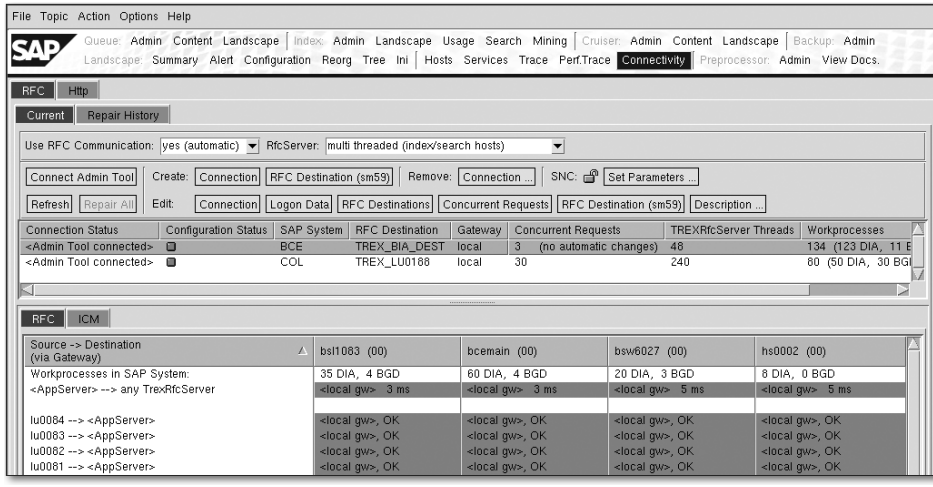


Figure 5.12 TREX Admin Tool, Connectivity Repaired

The TREX administration tool includes a great deal more functionality for the automatic repair of connectivity issues, but the user experience is always similar to that illustrated here, and an experienced administrator should have no difficulty with it.

5.2.2 Consistency Checks

To ensure that the data in your BIA indexes are consistent with the InfoCube data in the BI system, you can run a variety of consistency checks. The main launch point for such checks is the BI Accelerator Data Consistency Check Center.

From the BIA monitor, choose Goto • Consistency Checks. This opens the BI Accelerator Data Consistency Check Center. From this center, you can execute consistency checks, schedule these checks, and view the logs of checks that have run (see Figure 5.13).

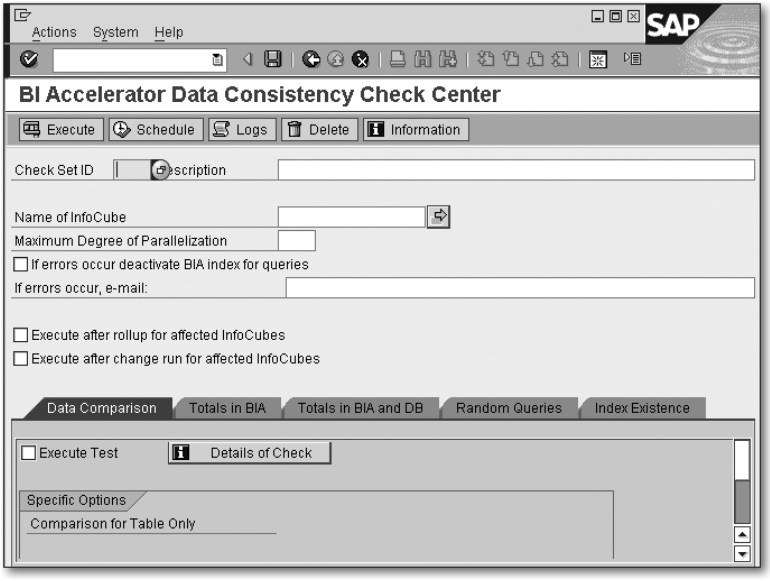


Figure 5.13 BI Accelerator Data Consistency Check Center

The BI Accelerator Data Consistency Check Center offers the following tabs from which you can execute the checks:

- Data Comparison
  Enables you to compare the contents of each individual table in an InfoCube with the contents of the corresponding index, record by record. This check is only suitable for relatively small tables and indexes.
- Totals in BIA
  Enables you to check key figure sums internally by executing a query on the BIA index using all key figures. Then the BI system executes a query for each

characteristic and navigation attribute occurring in the InfoCube that aggregates over all key figures. All the characteristics and navigation attributes that exist in the InfoCube are then included individually in the drilldown, and the totals are calculated. The system compares the result with the result of the first query. This test checks the completeness of the join paths to the fact tables. If the test shows that the data is incorrect, you need to rebuild the BIA index and its master data indexes.

- Totals in BIA and DB
  Enables you to compare key figure sums on the database against and in the BIA index. The system executes a query for each characteristic and navigation attribute occurring in the InfoCube that aggregates over all key figures on the BIA index and the database. Then it sums individually over all the characteristics and navigation attributes occurring in the InfoCube and compares the results from the database and the accelerator. The runtime of the test can be long. If it shows that the data is incorrect, you need to rebuild the BIA index and its master data indexes.
- Random Queries
  Enables you to checks for consistency by executing random queries, reading the data once from the database and once from the accelerator. The results should be the same. However, they can differ if the InfoCube data has been changed between execution of the query on the database and in the accelerator. You can verify the results as described later in this section. To repair the index, you need to rebuild it.
- Index Existence
  Enables you to checks whether indexes have been created for all the (relevant) tables in the star schema for the InfoCube. The test is very fast. If an index is missing, the BIA index needs to be rebuilt.

As an alternative to working from the BI Accelerator Data Consistency Check Center, you can run some fast index checks from the BIA monitor. Or you can run the more thorough checks in Transaction RSRV, to check that the indexes are complete and consistent.

To execute one or more sets of automatic BIA index checks in the BIA monitor, choose Index Checks • BI Accelerator.

This offers the following further options:

- Index Checks Schedule/Deschedule
  Toggles the index checks schedule on or off.
- Execute/Display/Change checks
  Displays a dialog box inviting you to flag the check set or sets you want to run, and then execute (see Figure 5.14). Once the checks have run, the results are displayed. In the run shown in Figure 5.15, 11 checks ran smoothly, and no red or yellow alerts were generated.

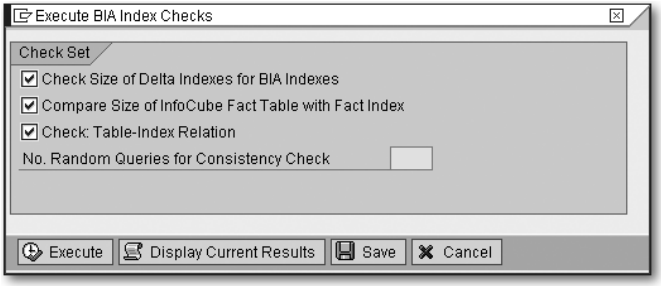


Figure 5.14 BIA Monitor, Execute Index Checks

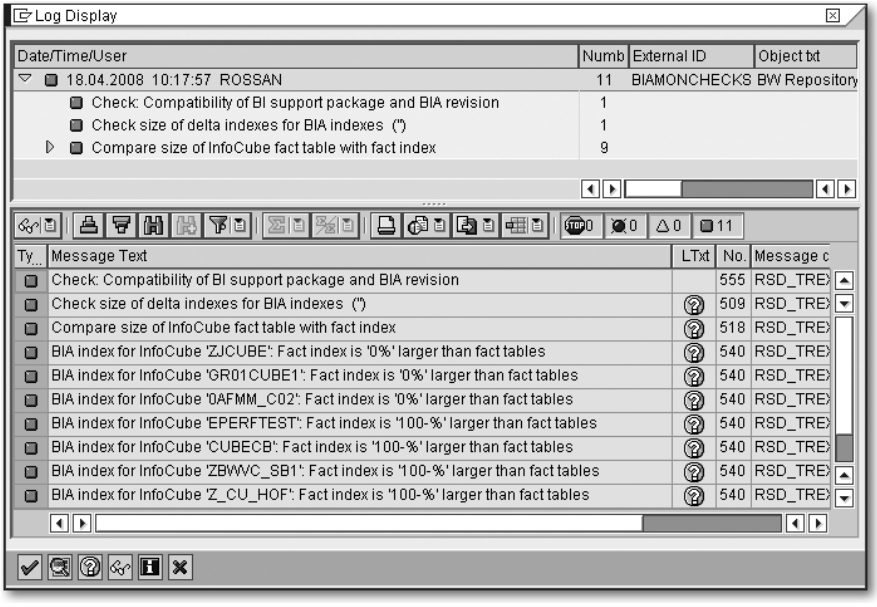


Figure 5.15 BIA Monitor, Index Check Results

The default set of index checks currently includes three checks. Check 1 looks at the size of delta indexes for BIA indexes, check 2 compares the size of InfoCube fact tables with their fact indexes, and check 3 looks at the relation between tables and indexes. The checks run quickly and provide useful information about performance and consistency. They can also be found and run in the analysis and repair environment (Transaction RSRV, see Section 5.2.6).

To execute selected elementary or combined checks in Transaction RSRV, drill down in the check set tree displayed on the left side of the screen to find the accelerator checks that you wish to run and double-click on them. This causes their names to appear on the right side of the screen (see Figure 5.16). Alternatively, you can drag them to the right and drop them there. Then, for each check on the right, you need to set the relevant parameters (this generally means specifying the InfoCube or InfoCubes to be checked). To do this, click on the lines at right to open a dialog box (see Figure 5.17) where you can enter the required values. When you are ready, choose EXECUTE and wait for the results (see Figure 5.18). These are displayed as a detailed list with a colored icon for each line to show at a glance whether all went well.

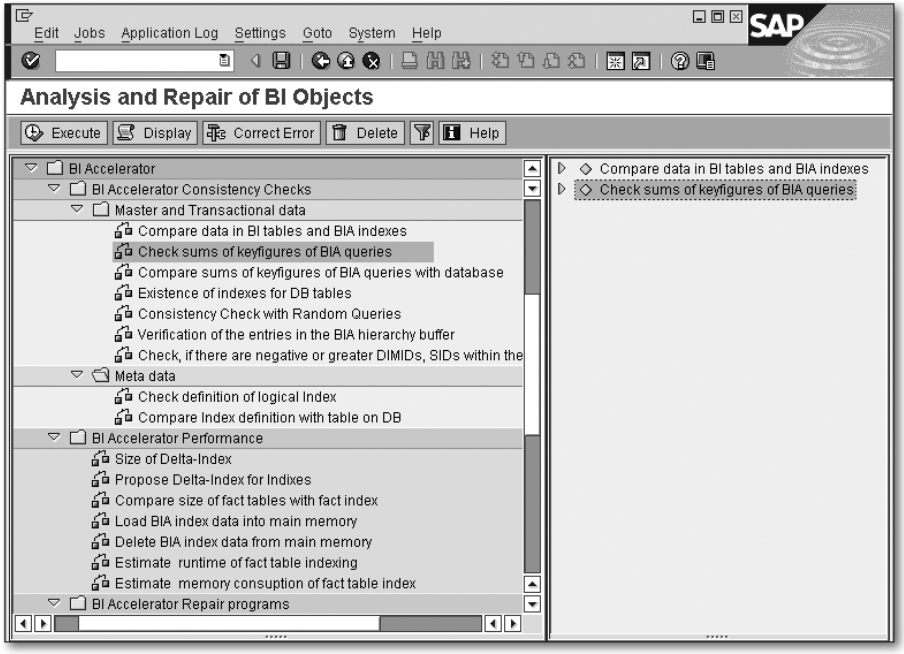


Figure 5.16 Transaction RSRV, Select Checks to Run

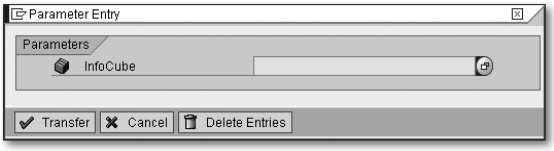


Figure 5.17 Transaction RSRV, Dialog Box for Parameter Entry

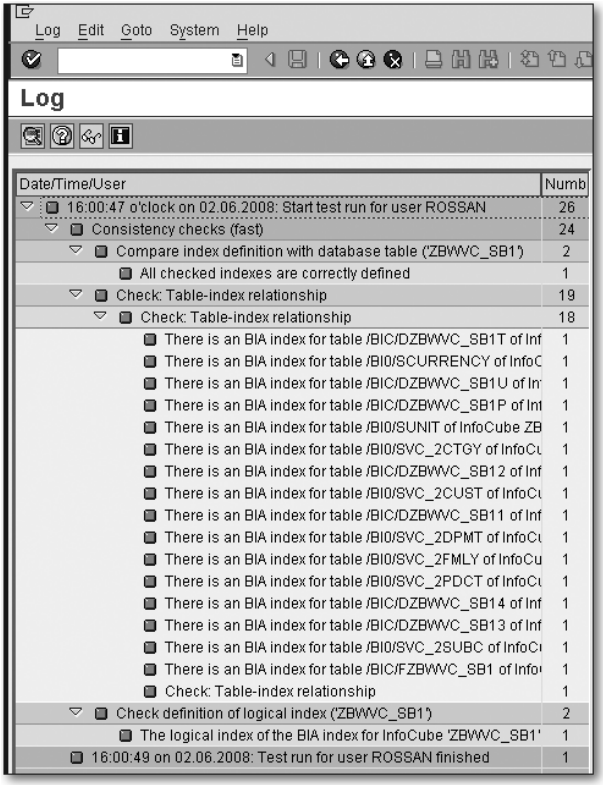


Figure 5.18 Transaction RSRV, Check Results

When to Run Checks: A Matrix

With so many checks, so many indexes to check, and so many possible reasons to run a check, it may seem impossible to follow a regular plan here. However, a little preparation can help. Draw a matrix with checks and reasons to run checks as its columns and rows, and then fill it out for yourself in view of your index landscape and your own priorities.

Table 5.1 shows a possible matrix. Here the 6 consistency checks listed below the matrix are numbered in the column heads and mapped to the 13 reasons to run checks listed in the rows. Check 1 is a query load generated by the RSTT trace tool described in Section 5.4.3, and checks 2–6 are listed with brief descriptions in Section 5.2.2.

As an example of how to read this matrix, the dots in it specify that after a roll-up or a change run, you plan to run the checks 2–6, and after hardware changes or performance shortfalls, you plan to run checks 1, 4, and 5.

The pattern of dots in this matrix is just san example, and as an administrator, you should plot your own matrix. Ideally, you should also insert more detail, for example, by adding a line specifying how often in the routine case you plan to run the check and a line indicating approximately how much time on average the check takes to run.

Consistency Check	1	2	3	4	5	6
Roll-up		•	•	•	•	•
Change run		•	•	•	•	•
Request deletion		•	•	•	•	•
Selective deletion		•	•	•	•	•
Metadata changes				•	•	•
Rebuild all indexes		•	•	•	•	•
Usage of delta index				•	•	•
Repair master data index	•	•	•	•	•	•
SPS/SP update	•	•	•	•	•	•
Revision update	•	•	•	•	•	•
Incorrect data		•	•	•	•	•
Hardware changes	•			•	•	
Performance issues	•			•	•	

1. Transaction RSTT generated query load
2. Compare data in BI tables and accelerator indexes
3. Check sums of key figures of accelerator queries
4. Check for consistency using random queries
5. Check existence of indexes for DB tables
6. Check definition of logical index

Table 5.1 Example of an Index Check Matrix

Recommended best practice is to regularly check the data in your accelerator and compare it with the data in the database. You can do so from the BIA monitor by choosing Goto • Consistency Checks, which opens the BI Accelerator Data Consistency Check Center (see Figure 5.13 shown earlier).

To minimize the system load and runtime for the consistency checks, use the following hints in a three-step approach:

1. **Check the facts**
   The fact indexes usually contain the most data and therefore take longest to check. To reduce the runtime of the check Totals in BIA and DB, set the Drilldown with InfoObject Only option to a characteristic with few attributes such as CALYEAR (see Figure 5.19).

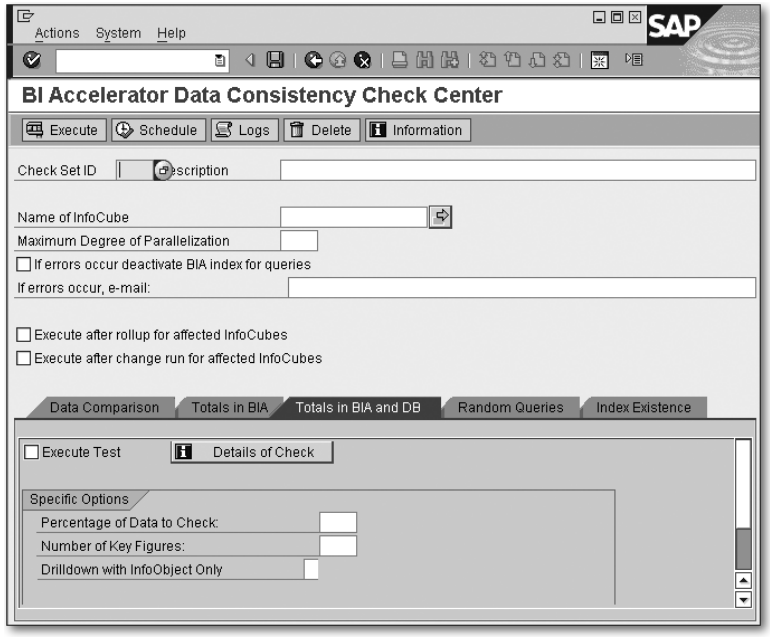


Figure 5.19 Data Consistency Check Center, Totals Tab in BIA and DB

If the InfoCube contains many key figures, you can reduce the load on the accelerator by reducing the number of key figures. If the runtime of this check is still too long, try reducing the percentage of data to be checked. A key figure overflow occurs during the check if the key figure type cannot contain the sum of all values.

2. **Check the completeness of the star schema indexes**
   These indexes can be very large, and we do not recommend a regular completeness check because it is too expensive.

   Instead, use the Totals in BIA check to find incorrect or missing records in the fact tables. Execute all joins of the extended star schema, and compare the results as a complete aggregation on the fact table. This acts like a filter because if there are incorrect or missing records in one of the indexes, the result of the aggregation on the fact table is different from the reference result.

   Again, if the InfoCube contains many key figures, you can reduce the load by reducing the number of key figures, and if a key figure overflow occurs, you can reduce the percentage of data checked.

3. **Check data consistency in complex situations**
   You can check the BIA data using random queries with complex conditions. The performance of this check depends on the performance of the query in the database. If your database still has aggregates for the InfoCube you want to check, then some of the randomly generated queries can be processed efficiently in the database, and in this case the performance of the check will be better.

For the details behind these hints, see SAP Note 1095886.

Master Data and Transaction Data

We now look one by one at the consistency checks available in the analysis and repair Transaction RSRV.

Compare data in BI tables and BIA indexes

This check compares the contents of each individual table with the contents of the corresponding index, record by record. It is only suitable for tables or indexes that do not contain a large amount of data, such as dimension tables, certain S tables, and X and Y tables. Fact tables are normally too large for this check. If a table contains 10,000 records or more, it is not checked.

In some situations, the content of the indexes of the BIA index may differ from the content of the corresponding database table. This may be the case if requests have been deleted from the InfoCube or if an InfoCube has been compressed.

Check sums of key figures of BIA queries

This check first executes a query on the BIA index, which is aggregated using all key figures, then includes all the characteristics and navigation attributes that exist in the InfoCube individually in drilldowns, and calculates the totals. The results are now compared to check the completeness of the join paths to the fact tables.

The runtime of the check depends on the number of characteristics and attributes and on the size of the fact table. If the test shows that the data is incorrect, you need to rebuild the BIA index and the master data indexes.

Check sums of key figures of BIA queries with database

This check executes highly aggregated queries and compares the results from the database with those from the BIA index. For large InfoCubes, the runtime may be long.

Check existence of indexes for database tables

This test checks whether all the (relevant) indexes for a given InfoCube have been created on the BIA server. Its runtime is very fast. If the test reveals that an index is missing, the BIA index needs to be rebuilt.

Check for consistency using random queries

This check executes random queries without persisting them. The system reads the data once from the database and once from the accelerator, and then compares the results. If the results differ, an error message is output.

The results can be different if the data of the InfoCube is changed between execution of the query on the database and in the accelerator.

To verify the results, execute program `RSDRT_INFOPROV_RANDOM_QUERIES` with the following parameters:

- InfoProvider: Name of the InfoCube
- Number of queries: 10

- Starting value: Same as used by the random generator
- Trace comparison: X

You can leave all other values unchanged. If the results are the same as from the check, you need to rebuild the BIA index.

Verify the entries in the BIA hierarchy buffer

When queries are executed in hierarchies, the hierarchy nodes are expanded to the relevant leaves and the expansion saved in a temporary index in the accelerator. The hierarchy buffer manages expanded hierarchies according to an LRU (least recently used) algorithm.

This check verifies whether all temporary indexes in the hierarchy buffer contain correct data. If the hierarchy buffer contains incorrect entries, do not delete the hierarchy buffer but send a customer message to SAP Service. If you urgently need to continue work, you can delete the entire hierarchy buffer, but this will make it harder to troubleshoot the error.

Metadata

Check definition of logical index

This check compares the definitions of each of the table indexes in a BIA index with the current versions of the database tables. It checks whether the number, name, and type of the table fields in the database match the definition for the index on the BIA server. If you do not specify an InfoCube, the system executes the test for all InfoCubes that have a BIA index.

If a table definition has been changed, the system deletes the old index, creates a new index with the current definition, and fills it. All BIA indexes that use this index are set to inactive and become unavailable for reporting during this time. The period of unavailability depends on the size of the table that needs to be re-indexed.

Compare index definition with table on database

The system checks the logical index of a BIA index. The logical index contains the metadata for the BIA index, such as the join conditions and the names of the

fields, and may change if the InfoCube is changed. If you do not specify an InfoCube, the system executes the test for all InfoCubes that have a BIA index.

If the logical index has been changed, the system deletes the old index and creates a new index with the correct definition. The system temporarily sets the BIA index to inactive, and the index is unavailable for reporting during this time.

Find indexes with status unknown

The system checks whether BIA indexes contain indexes that have the status unknown. This only occurs in exceptional cases when the Commit Optimize call terminates during indexing. Because in this case it is not clear whether the previously indexed data is available, the affected indexes are rebuilt in repair mode.

5.2.3 Check Sets for BIA Indexes

From the BI Accelerator Data Consistency Check Center, you can create and schedule check sets. To access the center from the BIA monitor, choose Goto • Consistency Checks. This displays the Data Consistency Check Center (see Figure 5.13 shown earlier). On this screen, you can schedule and run checks of the index data on the BIA server, view the logs of checks that have run, and group certain checks to form check sets.

Procedure for Creating a New Check Set

To create a new check set, follow these steps:

1. Give the check set a description.
2. Specify the InfoCubes corresponding to the accelerator indexes for which the check set is to be executed. Input help is available.
3. Specify the maximum degree *N* of parallelization for background processing, if the checks are to run in background. The system starts up to *N* simultaneous dialog processes, with one for each InfoCube.
4. If necessary, set the indicator to deactivate an accelerator index for queries if errors occur. If this indicator is set, the accelerator index is inactivated (so that it cannot be used for queries) as soon as the check set reports incorrect data in the accelerator index. This prevents the accelerator from using incorrect data for reporting. In some circumstances, a check can report incorrect data even

though the data is correct. Then deactivation is unnecessary, but it is still better than using incorrect data for queries.

5. If you want an e-mail to be sent if an error occurs (if incorrect data is reported), enter the address of the recipient in the relevant field.
6. If the check set is to be executed immediately after the roll-up of new requests to an InfoCube, set the relevant indicator. The check set is then still part of the process (this is relevant for integration into a process chain), but the lock on the process is no longer valid so that other processes are not interrupted. The check set is not executed for all InfoCubes, but only for the InfoCube for which the data was rolled up.
7. If the check set is to be executed immediately after the change run, set the relevant indicator. As before, the check set is still part of the process, but the lock on the process is no longer valid. The check set is only executed for the InfoCubes whose accelerator index was adjusted in the change run.
8. Each tab in the screen controls a test, as described in Section 5.2.2. Select the checks you want for your check set, and select the relevant options.
9. Save the check set. A check set ID is allocated and displayed.

Displaying and Changing a Check Set

To display an existing check set, select it from the input help for the CHECK SET ID field (see Figure 5.20). You can change the parameter values of the selected check set and save it again. The CHECK SET ID stays the same.

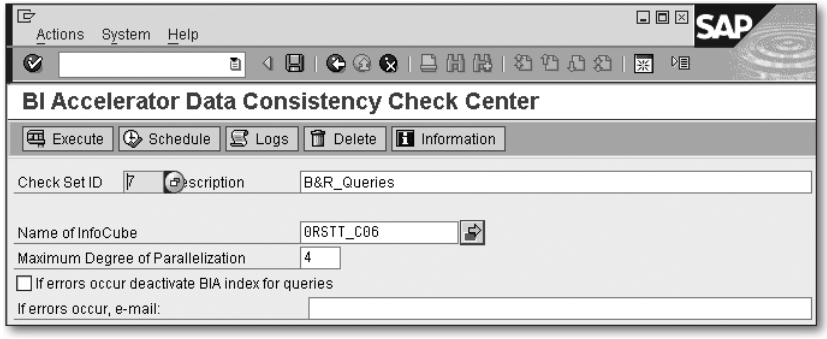


Figure 5.20 Display a Check Set

To delete a check set, select it, choose DELETE, and confirm at the prompt.

Executing a Check Set

To execute a check set, just select it, and choose Execute. The checks are executed in the dialog (and not in parallel). If you have just created the check set, you do not even have to save it. When the check is complete, the system displays the results in the application log.

To schedule a check set, choose Schedule. This opens the Start Time dialog box. Here you can schedule the check set to run once or periodically in the background. You need to save your entries to set the schedule. The name of the scheduled job is `BW_TR_RSDDTREX_INDEX_CHECK`.

You can also execute a check set by using program `RSDDTREX_INDEX_CHECK`.

To do this, you need the check set ID, or you can select the check set from the input help. You can also use this program to add a check set to process chains. To call the logs, choose Logs.

5.2.4 Index Repair Programs

The following are the index repair programs:

- Delete and rebuild all BIA indexes
  This repair first deletes and then recreates and fills all the BIA indexes in the accelerator. This is an extremely drastic action that can put your accelerator under full load for many hours and will make it unavailable for user requests during that time. In exceptional circumstances, if a critical error occurs, you may need to execute this action for a successful restart with consistent data, but before you run it you should consider the consequences carefully.
- BIA index adjustments after InfoCube activation
  If an InfoCube is changed, for example, by adding a key figure, the accelerator does not automatically adjust the BIA index because the adjustment may take a long time (see Section 5.2.5). This repair writes information about any changes to the log and makes the required changes in repair mode. If you execute this repair, we recommend that you run it as a background job.
- Rebuild all master data indexes of a BIA index
  We strongly recommend that you do not execute this repair. It deletes and rebuilds all master data indexes of a BIA index. These master data indexes are used by other BIA indexes, so the repair may result in terminations or poor performance during query execution. This repair also requires various data

loading processes to be locked. The repair is only for use in cases where there is incorrect data in the master data indexes of a BIA index. Such problems are serious. If you find such a problem, you should open a customer message. SAP service will analyze the problem, determine which index contains incorrect data, and rebuild the index using a program that is not released for general use. As of Support Package 18, this repair is no longer available in Transaction RSRV.

5.2.5 Impact of Metadata Changes on the Accelerator

If you change the metadata for an InfoCube that has a BIA index, the BIA index may need to be adjusted as well. In most cases, this adjustment occurs automatically as soon as the InfoCube is activated. The system compares the current metadata of the BIA index with the metadata from the newly changed InfoCube, and any discrepancies trigger adjustment actions. If the metadata changes for the InfoCube are transported to a system, the comparison and the adjustment occur in a post-activation step for the transport.

There are rare cases where the adjustment is not automatic. If you change the structural design of the InfoCube by adding or deleting a key figure or a characteristic, this change requires a complete rebuild of the BIA index. The rebuild is not performed automatically because the process may have a long runtime and impose significant load on the system. Instead, the BIA index is simply deleted. In this case, you need to schedule the rebuild manually to run at a convenient time.

For example, if you add a new navigational attribute to an InfoObject, this does not have an immediate impact on any BIA index, so no adjustment is triggered when you activate the InfoObject. The index data and metadata in the BIA index needs to be adjusted only if the navigational attribute is turned on for an InfoCube. In that case, the logical index of the BIA index, which holds the metadata, is re-created for this InfoCube. Then the master data indexes need to be dropped and re-created with the data for the new navigational attribute.

To ensure consistency, the system rebuilds not just a single index but all the master data indexes for the InfoObject, which means all the S, X, and Y indexes. Rebuilding the master data indexes can be a time-consuming process, too, so if the size of the table exceeds a preset limit (the default value is 50,000 lines), the rebuild runs in a separate background process.

If this adjustment process fails for any reason, you can repair the BIA index in Transaction RSRV by running the repair program BIA INDEX ADJUSTMENTS AFTER INFOCUBE ACTIVATION. This is equivalent to running program RSDDTREX_INDEX_ADJUST. The program compares the metadata and then triggers any necessary adjustments. For performance reasons, you should run the program as a background job.

5.2.6 Rebuilding Indexes

If you find that a BI query reads data from the BI accelerator and outputs incorrect data or data that is different from the results returned when the database is read, you should do some preliminary troubleshooting before you approach SAP service experts for help.

First, make sure the problem is not caused by the database. You can do this by running the query twice in Transaction RSRT in debug mode, once on the database and once with the accelerator. If the results are different, make sure the latest database patches are applied, and check for SAP Notes on the patches. If you still have a problem, mention this test result in your customer message to SAP.

To enable SAP experts to solve the problem effectively, you need to tell them both which query delivered the wrong data and all information needed to reproduce the problem. You should specify the row and column of at least one cell that contains incorrect data, together with the name of the key figure and the characteristic values. If possible, you should provide a second query that shows the discrepancy when compared with the first. If data disappear during navigation, you should describe this navigation and create an OLAP trace. It is helpful to reduce the queries to the absolute minimum of selected values. For further details, see SAP Notes 995364, 1060387, and 1095886.

In exceptionally rare cases, BIA indexes can become corrupt and require rebuilding. A set of BIA checks can be activated to check the indexes, for example, to compare the index data with the BI table data or to check that the lists of indexes and tables correspond. If there is a problem with an index, the index can be deactivated, and any individual table within an index can be rebuilt as required to render the index fully functional again. All the functionality for checking, rebuilding, and rechecking the indexes is highly automated.

In Transaction RSRV (see Figure 4.7 shown earlier), you can analyze and repair BI objects such as BIA indexes.

Checks are available for:

- Testing for inconsistencies between the data in the InfoCube on the database and the data in the BIA index — node BI Accelerator Consistency Checks
- Testing whether an accelerator index is running with optimal performance — node BI Accelerator Performance Checks
- Completely or partially building or rebuilding all BIA indexes or a specific BIA index — node BI Accelerator Repair Programs

The exactness and duration of each of these checks vary. For most purposes, you would run data consistency checks from the BI Accelerator Data Consistency Check Center (see Section 5.2.2) rather than from Transaction RSRV.

In the BIA monitor, you can specify that the system is to run a small number of tests on a daily basis. You do this by choosing BI Accelerator • Execute/Display Index Checks.

Some of the tests work with statistics data. The statistics have to be switched on for the relevant InfoProvider. You make this setting in the statistics properties maintenance screen. On the Data Warehousing Workbench screen, choose Tools • Settings for BI Statistics.

Transaction RSRV currently includes the following groups of consistency checks, performance checks, and repair programs for the accelerator:

- Consistency checks for master and transactional data
- Consistency checks for metadata
- Performance checks
- Repair programs

These tests can be run separately or combined. There are three predefined combinations of tests:

- Consistency checks (detailed)
- Consistency checks (fast)
- Performance

To run one or more checks, proceed as described in Section 5.2.2. If a check discovers a corrupt or missing index, one option is to start Transaction RSDDV and rebuild the entire BIA index containing that index.

If the option of rebuilding the entire BIA index requires too much runtime, or perhaps fails in execution for any reason, another option is available to an SAP Service engineer. This is to identify which individual table index or indexes need to be rebuilt and rebuild them separately. This option requires deep understanding of the SAP NetWeaver BI system landscape and is not supported for use by anyone except SAP Service engineers.

You can find out which table index is causing a problem by looking at the indexing logs. Running the BIA Index Maintenance Wizard in Transaction RSDDV generates application logs.

To view the logs, choose the Application Logs wizard button. A dialog box appears. Select the process logs you wish to see and execute. The screen shown in Figure 5.21 appears, with the Object ("RSDDTREX", BIA index), Subobject ("TAGGRFILL", fill BIA index), and External ID fields already filled. Enter any further information, and execute. The indexing logs are then displayed, with colored icons to indicate the success of the indexing steps.

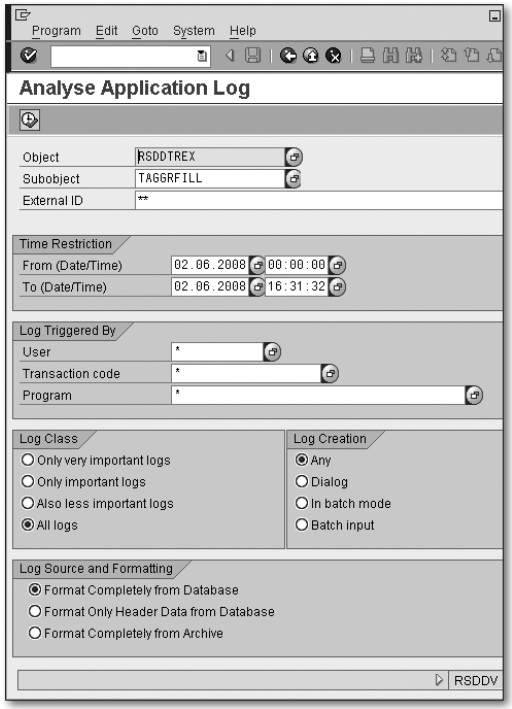
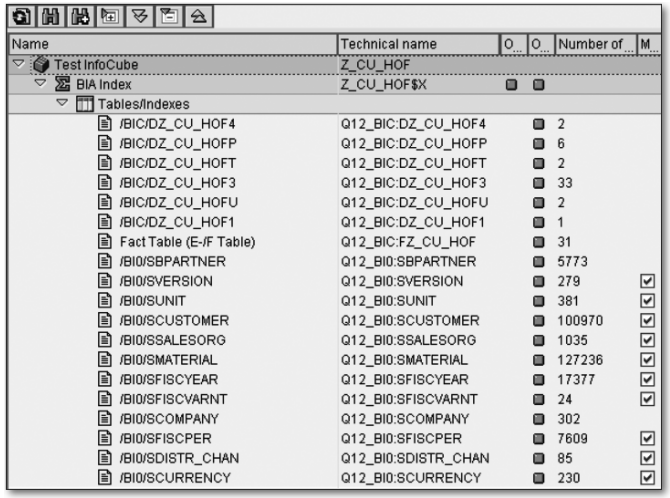


Figure 5.21 Transaction RSDDV, Analyse Application Log

To verify that the BIA indexes now have status green, start Transaction RSDDV, and choose BIA INDEXES. Drilling down on an index displays the details shown in Figure 5.22. In this case, all the table index icons are green, and all is well for this BIA index. The flags on the right indicate that some of the master data tables are shared with other BIA indexes.



Name	Technical name	O...	O...	Number of...	M...
Test InfoCube	Z_CU_HOF				
BIA Index	Z_CU_HOF$X				
Tables/Indexes					
/BIC/DZ_CU_HOF4	Q12_BIC:DZ_CU_HOF4			2	
/BIC/DZ_CU_HOFP	Q12_BIC:DZ_CU_HOFP			6	
/BIC/DZ_CU_HOFT	Q12_BIC:DZ_CU_HOFT			2	
/BIC/DZ_CU_HOF3	Q12_BIC:DZ_CU_HOF3			33	
/BIC/DZ_CU_HOFU	Q12_BIC:DZ_CU_HOFU			2	
/BIC/DZ_CU_HOF1	Q12_BIC:DZ_CU_HOF1			1	
Fact Table (E-/F Table)	Q12_BIC:FZ_CU_HOF			31	
/BI0/SBPARTNER	Q12_BI0:SBPARTNER			5773	
/BI0/SVERSION	Q12_BI0:SVERSION			279	☑
/BI0/SUNIT	Q12_BI0:SUNIT			381	☑
/BI0/SCUSTOMER	Q12_BI0:SCUSTOMER			100970	☑
/BI0/SSALESORG	Q12_BI0:SSALESORG			1035	☑
/BI0/SMATERIAL	Q12_BI0:SMATERIAL			127236	☑
/BI0/SFISCYEAR	Q12_BI0:SFISCYEAR			17377	☑
/BI0/SFISCVARNT	Q12_BI0:SFISCVARNT			24	☑
/BI0/SCOMPANY	Q12_BI0:SCOMPANY			302	
/BI0/SFISCPER	Q12_BI0:SFISCPER			7609	☑
/BI0/SDISTR_CHAN	Q12_BI0:SDISTR_CHAN			85	☑
/BI0/SCURRENCY	Q12_BI0:SCURRENCY			230	☑

Figure 5.22 Transaction RSDDV, Check Index Status

Finally, to confirm that the indexing was successful, you can check the logs again as follows. Start Transaction SLG1, specify OBJECT "RSDDTREX" and SUBOBJECT "TAGGRFILL", and execute (the screen is as shown earlier in Figure 5.21). This will display a set of logs with colored icons indicating the success of the indexing actions. If there were problems here and you wished to drill deeper, you could start Transaction SM21 and study the system log.

5.2.7 Performing a Recovery

During a recovery, no indexing is possible, and any indexing processes that were running before the event that necessitated the recovery are terminated.

A BIA recovery involves importing a saved snapshot to a BIA system. Any other data loads to the accelerator are automatically put on hold during the imports.

A restore from an import deletes all the BIA indexes that have been backed up and imports all the indexes from the backup. The new BIA indexes are distributed

over the BIA servers as before. After the recovery has completed, you may need to trigger a reorganization of the index landscape.

Once the snapshot is imported, the next step is index adjustment. If an index has not been changed because the snapshot was created (which is checked by comparing timestamps), no adjustment is needed. If the index has changed, the adjustment depends on the index type. Fact index requests can be reloaded (if they have not been compressed), dimension indexes are completely rebuilt (except the package dimension index, which is adjusted), and S/X/Y indexes are completely rebuilt.

You can perform a recovery from the BIA monitor as follows:

1. Choose BI Accelerator • Maintenance Functions • Backup and Recovery.
2. If you wish to estimate how long the recovery process will take, select the relevant snapshot, and choose Simulate.
3. Select the snapshot you want to recover, and start the recovery process. The job executes immediately and cannot be scheduled.
4. You can view the job log. A new button is displayed indicating that the recovery process is running. To view the log, click the button.
5. Check the BIA configuration, the initial BIA logs, and the consistency of the BIA indexes.

To ensure that you are prepared to perform a recovery if necessary, you can simulate a recovery. This can also give you an indication of whether it would be faster to re-index all your data from scratch.

If you like, instead of simulating a recovery to decide whether it would be faster than re-indexing, you can evaluate the following factors:

- Age of the snapshot and how many updates have occurred since it was made
- Degree of parallelism and number of CPU cores of the TREX backup server
- Performance of the storage system for the backups
- Bandwidth between the TREX backup server and the storage system

Before performing a BIA recovery, it may be a good idea to check anything in SAP NetWeaver BI that can affect the status of the accelerator:

- Check the InfoCube data to ensure that the roll-up status is up to date.
- Check that all the master data used in BIA have been updated.

Index

A

ABAP 25, 49, 186, 243
ABAP client 53
Activating delta indexing 175
Agassi, Shai 19
Aggregate 26, 33, 36, 41, 43, 76, 120, 237
  Aggregate strategy 38
  Building aggregates 37
Aggregation 17
  Aggregation algorithm 50, 222, 223
Alert server 53, 111, 138
Alert Server Configuration 112
Alert server trace 194
Anchor table 231
Appliance 10, 17, 39, 47, 48, 55, 221
  Appliance model 39
  Backend appliance 42
Assmann, Matthias 212

B

Backup 62, 125
Backup and recovery 62
Backup blade server 57, 60
Backup mode 61
Basket analysis 32
Bauer, Reinhard 216
Bayer AG 209
Bayer Business Services 209
Bayer MaterialScience 209
Best practice for accelerator backups 126
BI analytic processor 120
BI compression 36
BIA 243
BIA engine 49
BIA index 231
BIA index backup 125
BIA Index Maintenance Wizard 98, 176
BIA load monitor 124
BIA snapshot 62, 126, 161, 162
BIA system check 89
Binary large object (blob) 53
Blade server 41, 45, 46, 47, 48, 71, 127, 221, 222
Bundeswehr 210
Business Intelligence 243
Business Objects, an SAP company 26
BusinessObjects
  Enterprise 27
  Polestar 27, 28, 108
  Voyager 27
  Web Intelligence 27
  Xcelsius 27, 35

C

Call center 30
CCMS infrastructure 54
CCMS monitoring 83
CCMS monitoring framework 84
Central gateway 140
Change run 76, 121, 122
Changing a check set 155
Changing trace levels 192
Characteristics 37, 227
Check matrix 148
Check set 154
Client-server paradigm 21
Cloning a BIA instance 71, 127
Colgate-Palmolive 205
Composition Environment 24
Compression 223
  Cluster coding 226
  Difference coding 224
  Golomb coding 224
  Indirect coding 226
  Integer coding 224
  Prefix coding 225
  Run length coding 226
  Sparse coding 226
Compression technique 45, 50, 224, 225
Consistency check 144, 159
Creating a new check set 154
CRM 243
Crystal Reports 27

Customer Relationship Management 243
Customer segmentation 32

D

D table 244
Data alignment 31
Data bus bandwidth 70
Data Consistency Check Center 82, 144, 150, 154, 159
Data modeling strategy 36
Data partitioning 45
Data staging 236, 237
Data warehouse 244
Data warehousing 26, 39, 237
Data Warehousing Workbench (DW Workbench) 83, 84, 103, 159, 176, 249
Database 119, 221
Database backup 62
DataStore object 26, 36, 237, 238, 244
Debugging 188
Deleting data 122
Delta index 109, 172
Delta index file 51
Delta index merge 123
Delta indexing 50, 76, 175
Deutsche Telekom AG 211
Dimension identifier 227
Dimension table 227, 244
Disaster recovery 66
Disaster tolerance 66
Display attribute 229, 244

E

E table 244
Ease of reporting 33
EIM 244
E-mail notification 82, 111, 117, 137
Enhanced accelerator index 108
Enhanced star schema 96
Enterprise information management 24, 244
Enterprise resource planning 244
ERP 244
ETL 244
Executing a check set 156
Extended star schema 49
Extraction, transformation, and load (ETL) 244

F

F table 244
Fact table 96, 174, 227, 230, 244
Failover 117
Fault tolerant system 66
FEMS compression 236
File storage system 41
File storage unit 48
Flat accelerator index 96
Fujitsu Siemens 18, 48

G

General Parallel File System (GPFS) 55, 68, 212, 216, 245
German Defense Forces 210
Global indexing parameters 111
Going live 120
Golomb coding 224, 245
GPFS 55, 68, 212, 216, 245
Graphviz 170
GSM cellular telephony 211

H

Hard disk drive 12
Hardware costs 40
Hardware investment 33
Hewlett-Packard 18, 19, 48, 214
HP-UX 19
ProLiant 19
Hierarchy table 122
High availability (HA) 57
Horizontal data partitioning 45
HTML 54, 186
HTTP 53
HTTPS option 53

I

IBM 18, 19, 48, 55, 68, 72
BladeCenter 210, 212, 213, 216
DB2 204, 210
Global Technology Services 212, 215
System Storage 212, 216
Systems Solution 213
TotalStorage 210
WebSphere 24
Index administration 165
Index check 158
Index fill process 177
Index landscape 167
Index reorganization 76, 81, 123, 130, 162, 163, 168, 222
Index splitting threshold 70
Index vector 220
Indexing parameters 111
InfoCube 17, 26, 37, 41, 42, 45, 49, 71, 76, 96, 150, 219, 227, 228, 230, 231, 237, 246
InfoObject 96, 150
InfoProvider 43, 46, 77
Installation script 57
Installer script 127
INT4 246
Intel 10, 18, 48, 50, 212
Itanium CPU 19
Xeon processor 18, 19, 48, 210
Internet Connection Manager (ICM) 53, 95
Inverted index 220, 221

J

Java 25, 49
Java Runtime Environment (JRE) 184
Join condition 166, 230
Join graph 166, 170, 229, 231
Join path 166, 231

K

Key figure 227, 247
Key performance indicator (KPI) 164, 247
KF 247
Kimberly-Clark Corporation 213
Knowledge Management 24, 40, 247
KPI 164, 247

L

Landscape snapshot 186
landscapeOverview.py 186
Least recently used 247
Linear scalability 223
Linux 10, 18, 39, 55, 127, 210
Linux cluster 68
SLES 10 48
Load monitor 124
Local area network (LAN) 88
Local gateway 140
Logical cube index 50, 234
LRU 247

M

Mainframe paradigm 21
Master data 76, 151, 227, 228
Master data identifiers 227, 229
Master data index 157
Master data table 122, 229
Master name server 59, 134
Metadata 153, 231
Metadata changes 157
Microsoft .NET 24
Microsoft PowerPoint 35
Multiquery 235

N

NAS 247
Navigational attribute 157, 227, 229, 247
Network attached storage 247
Network File System (NFS) 55, 248
NFS 55, 248
Nickolai, Phil 214

O

Object RSDDTREX 107, 160, 161
OCFS 248
OCFS mirroring 67

ODS 248
OLAP 25, 248
OLAP cache 43
OLAP trace 188, 189
Online analytical processing 25, 248
Operational data store 26, 248
Oracle Cluster File System (OCFS) 55, 248

P

P table 248
Package dimension table 122
Parallel processing 221
Parameter
  BATCHPARA 179
  FLOAT 180
  NUMPROC 180
  PKGSIZE 180
  SUBPKGSIZE 180
Performance 30
Performance check 159, 172
Performance trace 195
Performance tracing 188
Performance tuning 181
Persistent staging area 248
Phased roll-out 120
Plan operation 233
Planned downtime 57
Preprocessor 54
Program RSDDTREX_INDEX_ADJUST 158
Project Jupiter 19, 20, 72, 201–204
PSA 248
PuTTY 184, 248
Python 49, 186, 188
Python trace 197
Python tracing 188

Q

Q table 248
Query monitor 192
Query plan 50, 233
Queue server 54, 55
QuickSizer 72

R

Rabobank Group 212
Rabobank Nederland 213
Radio frequency tagging 31
RAID 55, 248
RAM 11, 42, 249
Random access memory (RAM) 11, 41, 249
Recovery 62, 125, 161
Relative costs 11
Remote Function Call 249
Reorganization algorithm 163, 171
Repair program 156, 159
Restore 161
Retail sector 30
RFC 249
RFC calls 177
RFC check 138, 141
RFC connection 48, 92, 138
RFC destination 52, 138
RFC server 179
RFC server thread 141
Robustness 31
ROI calculator 34
Roll-up 121
RSA1 83, 84, 103, 249
RSCUSTA 91, 92
RSDDBIAMON 62, 77, 97, 112, 133, 229, 249
RSDDTREXEMAIL 117
RSDDV 82, 83, 84, 97, 98, 249
RSRV 82, 84, 145, 147, 151,249
RSTT 189, 249
Runtime statistics 176
RZ20 83, 249

S

S table 252
SAP Business All-in-One 243
SAP Business ByDesign 243
SAP Business Explorer 189
SAP Business Suite 243
SAP Communities of Innovation 15
SAP Developer Network 15
SAP gateway 179

SAP NetWeaver BI 7.0 26, 46
SAP NetWeaver BI InfoCube 246
SAP NetWeaver EIM 247
SAP NetWeaver Master Data Management (SAP NetWeaver MDM) 24
SAP Note
1002839 182
1018798 182
1047527 183
1058533 186
1060387 158
1074559 183
1091714 183
1093719 183
1095886 151, 158
1132572 183
1149760 183
1156093 80
1157582 182
1160520 182
1163149 183
1169640 183
460652 181
883726 75
917803 72
940635 119
970771 84
988989 120
992064 90
995364 158
998680 190
SAP QuickSizer 72
SAP Service Marketplace 57
SAPPHIRE 2005 19
Saprfc.ini 139
SAProuter 184
SAPROUTTAB 184
Scalability 30, 68
SCM 249
SE16 117, 118, 182, 250
SE38 187, 250
Secure network communication 53
Service connection 184
Service level agreement 30, 66
Service-oriented architectures (SOA) 20
Simplified data modeling 34
Sizing 71, 72
Accelerator sizing report 72
SLG1 161, 250
SM59 91, 138, 141250
Smart compression 223
SMTP server 137
Snapshot 61, 62, 126, 161, 162
SOAP 25
Software as a service 39
Software on demand 39
Sold, Harald 210
Sparse data 224
Split index 163, 222
Stable response time 30
Staged deployment strategy 71
Standard trace 191
Standard tracing 188
Star schema 49, 170, 227, 229, 230
Steinhorst, Jörg 211
Stock keeping unit (SKU) 32
Subobject TAGGRFILL 107, 160, 161
Sun Microsystems 18, 48
Supply chain management 249
Surrogate IDs 227
SUSE Linux 18, 19, 48
Switchover 66
System check 82, 89
System consolidation 33

T

Table
RSDDSTAT_OLAP 182
RSDDSTATTREX 176
RSDDSTATTREXSERV 177
RSDDTREXADMIN 95
RSDDTREXEMAIL 117, 119
RSDDTREXHPAFAIL 118, 119
TCP/IP 52
Technical Operations Manual 15, 75, 91, 95, 182, 236
Telephone companies 31
Threshold for splitting indexes 222
T-Mobile UK 211
Topology.ini 52, 59, 141
Total cost of ownership (TCO) 46

Trace 188
Transaction
RSA1 83, 84, 103, 249
RSCUSTA 91, 92
RSDDBIAMON 62, 77, 97, 112, 133, 229, 249
RSDDV 82, 83, 84, 97, 98, 99, 103, 108, 159, 161, 229, 249
RSRT 96, 158, 182, 192
RSRV 82, 84, 145, 147, 151, 158, 249
RSTT 189, 249
RZ20 83, 249
SE16 117, 118, 182, 250
SE38 187, 250
SLG1 161
SM21 161, 250
SM59 91, 138, 141, 250
TREXADMIN 82, 87, 94, 95, 128, 129, 130, 133, 188, 192, 251
Transaction data 151
TREX 10, 40, 47, 251
ABAP API 49, 179
alert server 53
application programming interface (API) 49
architecture 47
binaries 47, 58, 125
client 179
Cruiser 55
daemon 49, 59, 62
index server 49, 60
metamodel 49, 229, 230, 231
name server 51, 59
preprocessor 54
queue server 55
revisions 57
RFC server 52
services 47
standalone administration tool 133
trace levels 191
updates 57
web server 54
TREXADMIN 82, 87, 94, 95, 125, 129, 130, 133, 188, 192, 251
TREXDaemon.ini 59

U

Updating BIA instance 57, 129
Upgrade 57
Usability 31

V

VA 251
Value proposition 18, 28, 46
van der Horst, Tonnie 213
Vavruska, Pete 215
Vertical data partitioning 45, 68, 219
Vertical decomposition 219
Vertical partitioning 69
View attribute 251
Virgin Mobile 211
Virtual Network Computing (VNC) 184

W

WACKER Group 215
Watchdog process 59
Web server 54
Web Services 21, 22, 23
Windows Terminal Server (WTS) 184
Winter Corporation 19, 20, 201–204, 230
Wireless local area network (WLAN) 24
Wizard 99, 106

X

Xeon → *see Intel*
X table 252
XML 24, 53

Y

Y table 252