SAP Database Administration
Guide for SAP NetWeaver on
IBM DB2 UDB for z/OS

Release 640
Document Version 4.0: April 20, 2005
Copyright

© Copyright 2004 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.
Icons in Body Text

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>Caution</td>
</tr>
<tr>
<td></td>
<td>Example</td>
</tr>
<tr>
<td></td>
<td>Note</td>
</tr>
<tr>
<td></td>
<td>Recommendation</td>
</tr>
<tr>
<td></td>
<td>Syntax</td>
</tr>
</tbody>
</table>

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see Help on Help → General Information Classes and Information Classes for Business Information Warehouse on the first page of any version of SAP Library.

Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Example text</em></td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options.</td>
</tr>
<tr>
<td></td>
<td>Cross-references to other documentation.</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles.</td>
</tr>
<tr>
<td>EXAMPLE TEXT</td>
<td>Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td><em>&lt;Example text&gt;</em></td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td>EXAMPLE TEXT</td>
<td>Keys on the keyboard, for example, F2 or ENTER.</td>
</tr>
</tbody>
</table>
35.2 Static Properties Passed to DB2 ................................................................. 134
35.3 Dynamic SAP Properties Passed to DB2 .................................................. 135
35.4 Analysis of Accounting Records ............................................................... 138

Monitoring and Performance ............................................................................. 139
36 Setup of Database Performance Monitoring with rfcoscol ................................ 140
36.1 Installation of rfcoscl ............................................................................... 141
36.2 Manual Installation of rfcoscl on z/OS ...................................................... 142
36.3 Configuration of rfcoscl ........................................................................... 144
36.4 Starting rfcoscl Manually ........................................................................ 145
36.5 Stopping rfcoscl Manually ...................................................................... 147
36.6 SAP System Configuration of the RFC Connection .................................... 148
36.7 Maintaining rfcoscl in Transaction DB2 .................................................... 152
36.8 Troubleshooting for rfcoscl ..................................................................... 156

37 Database Performance Monitor ..................................................................... 157
37.1 DB2 Subsystem Activity ............................................................................. 159
37.2 Lock Waits ............................................................................................... 161
37.3 DBM1 Virtual Storage .............................................................................. 162
37.4 Global Times ............................................................................................ 163
37.5 Thread Activity ......................................................................................... 164
37.6 Cached Statement Statistics ...................................................................... 166
37.7 Data Set Statistics .................................................................................... 168
37.8 Installation Parameters ............................................................................. 169
37.9 PTF Status ............................................................................................... 170
37.10 Automatic Start of DB Traces ................................................................. 171
37.11 DB Alert Router ..................................................................................... 172

38 Database Tools ............................................................................................. 175
38.1 Proactive Redirection of Application Servers to Other DB2 Subsystems ........ 176
38.2 Data Sharing Topology ............................................................................ 177
38.3 DB2 System Catalog ................................................................................ 178
38.4 z/OS System Log ..................................................................................... 179
38.5 DB2 Commands ....................................................................................... 181
38.6 Buffer Pool Tuning .................................................................................. 182
38.7 EXPLAIN ................................................................................................. 189
38.8 DB System Check .................................................................................... 192
38.9 DBSL Trace ............................................................................................. 193

39 DB2 Connect Diagnostics ............................................................................. 194
39.1 Network Turnaround Time ...................................................................... 196
39.2 Statistic Counters .................................................................................... 197
39.3 Measuring Network Turnaround Time ..................................................... 199
39.4 CLI Tracing ............................................................................................. 200
40 Tables and Indexes Monitoring ................................................................. 201
  40.1 Requirements .................................................................................. 202
  40.2 Tables and Indexes Monitor .............................................................. 203
  40.3 Checking Consistency .................................................................... 207
  40.4 Database Space Statistics ............................................................... 208
  40.5 Detailed Information on Tablespaces .............................................. 209
  40.6 Detailed Information on Indexes ...................................................... 210
  40.7 Detailed Information on Tables ....................................................... 211
  40.8 Extent Monitor ............................................................................... 212
  40.9 Volume Free Space ......................................................................... 214
  40.10 Troubleshooting for DB2 Refresh .................................................. 215
41 CCMS Monitor Set ............................................................................. 217
42 Switching on WLM Management ....................................................... 222
Database Administration ........................................................................ 225
43 Backup and Recovery Options ........................................................... 226
  43.1 Backup ........................................................................................... 228
  43.2 Online Backup ............................................................................... 229
  43.3 Offline Backup .............................................................................. 231
  43.4 Backup Procedure Recommendations ........................................... 233
  43.5 CCMS Backup Monitor ................................................................. 235
  43.6 Recovery ........................................................................................ 237
  43.7 Recovery to the Current State ....................................................... 239
  43.8 Recovery to a Prior Point in Time ................................................. 240
44 JES Interface ...................................................................................... 246
  44.1 Profile Parameters ......................................................................... 247
  44.2 SYSPLEX Failover Support and Connection Profile ....................... 248
  44.3 Prerequisites ................................................................................. 253
  44.4 Listing z/OS jobs ........................................................................... 254
  44.5 Creating and Saving z/OS JCL Jobs .............................................. 255
  44.6 Changing z/OS JCL Jobs or Stored Procedure Utility Statements .... 256
  44.7 Deleting z/OS JCL Jobs ................................................................. 257
  44.8 Displaying JCL Jobs and Stored Procedure Utility Statements .......... 258
  44.9 Entering a TSO Password ............................................................. 259
  44.10 Creating and Changing an Individual Jobcard ............................ 260
  44.11 Submitting z/OS JCL Jobs Asynchronously .................................. 261
  44.12 Checking the Status of the Job ..................................................... 262
  44.13 Deleting the Job Output on z/OS and on the Client ...................... 263
  44.14 Displaying the Job Output ......................................................... 264
  44.15 Cleaning Up the JES joblog Directory ....................................... 265
45 DBA Planning Calendar ...................................................................... 266
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>45.1</td>
<td>Preparations for Using the DBA Planning Calendar</td>
<td>267</td>
</tr>
<tr>
<td>45.2</td>
<td>Plannable Actions</td>
<td>268</td>
</tr>
<tr>
<td>45.3</td>
<td>Basic Functions</td>
<td>269</td>
</tr>
<tr>
<td>45.4</td>
<td>How the DBA Planning Calendar Works</td>
<td>270</td>
</tr>
<tr>
<td>45.4.1</td>
<td>Security Considerations for Stored Procedures</td>
<td>271</td>
</tr>
<tr>
<td>45.4.2</td>
<td>Executing a DBA Action</td>
<td>272</td>
</tr>
<tr>
<td>45.5</td>
<td>Using the DBA Planning Calendar</td>
<td>277</td>
</tr>
<tr>
<td>45.6</td>
<td>Backup</td>
<td>278</td>
</tr>
<tr>
<td>45.7</td>
<td>Recovery</td>
<td>281</td>
</tr>
<tr>
<td>45.8</td>
<td>Rebuild Index</td>
<td>282</td>
</tr>
<tr>
<td>45.9</td>
<td>Reorganization</td>
<td>283</td>
</tr>
<tr>
<td>45.10</td>
<td>Update Statistics(RUNSTATS)</td>
<td>285</td>
</tr>
<tr>
<td>45.11</td>
<td>Updating the Table and Index Monitor</td>
<td>287</td>
</tr>
<tr>
<td>45.12</td>
<td>Restarting Jobs</td>
<td>288</td>
</tr>
<tr>
<td>45.13</td>
<td>Displaying the DB2 Utility Status</td>
<td>289</td>
</tr>
<tr>
<td>45.14</td>
<td>Checking the Results of Actions</td>
<td>290</td>
</tr>
<tr>
<td>45.15</td>
<td>Adjusting the Jobs</td>
<td>291</td>
</tr>
<tr>
<td>45.16</td>
<td>Central DBA Planning Calendar</td>
<td>292</td>
</tr>
<tr>
<td></td>
<td>Storage Management</td>
<td>294</td>
</tr>
<tr>
<td>46</td>
<td>Changing the Database Layout</td>
<td>295</td>
</tr>
<tr>
<td>46.1</td>
<td>Creating Tables and Indexes</td>
<td>296</td>
</tr>
<tr>
<td>46.2</td>
<td>Modifying Tables and Indexes</td>
<td>297</td>
</tr>
<tr>
<td>46.3</td>
<td>Moving Tables</td>
<td>299</td>
</tr>
<tr>
<td>46.4</td>
<td>Rules for Self-Defined Objects</td>
<td>300</td>
</tr>
<tr>
<td>47</td>
<td>Storage Parameters</td>
<td>301</td>
</tr>
<tr>
<td>47.1</td>
<td>General Overview</td>
<td>302</td>
</tr>
<tr>
<td>47.2</td>
<td>Editing Storage Parameters</td>
<td>303</td>
</tr>
<tr>
<td>47.3</td>
<td>Changing the Source of Storage Parameters</td>
<td>305</td>
</tr>
<tr>
<td>47.4</td>
<td>Special Actions</td>
<td>306</td>
</tr>
<tr>
<td>47.5</td>
<td>Directly Changing Storage Attributes</td>
<td>307</td>
</tr>
<tr>
<td>47.6</td>
<td>Moving Tables to Existing Tablespaces</td>
<td>308</td>
</tr>
<tr>
<td>47.7</td>
<td>Isolating and Combining Tables</td>
<td>309</td>
</tr>
<tr>
<td>47.8</td>
<td>Partitioning Tables</td>
<td>310</td>
</tr>
<tr>
<td>47.9</td>
<td>Handling Large Tables</td>
<td>313</td>
</tr>
<tr>
<td>47.10</td>
<td>Mass Processing</td>
<td>314</td>
</tr>
<tr>
<td>47.11</td>
<td>Incremental Conversion</td>
<td>318</td>
</tr>
<tr>
<td>47.12</td>
<td>Troubleshooting</td>
<td>319</td>
</tr>
<tr>
<td>48</td>
<td>Space Management</td>
<td>320</td>
</tr>
<tr>
<td>48.1</td>
<td>Adding Volume Space</td>
<td>321</td>
</tr>
<tr>
<td>48.2</td>
<td>Data Compression</td>
<td>322</td>
</tr>
</tbody>
</table>
Additional Information ........................................................................................................ 324
49 Transaction Codes .................................................................................................... 325
50 Database Layout ....................................................................................................... 327
  50.1 General Overview ................................................................................................ 328
    50.1.1 Requirements .............................................................................................. 329
    50.1.2 SAP Technical Settings .............................................................................. 330
    50.1.3 Basic Concept ............................................................................................ 332
  50.2 Mapping Between the ABAP Dictionary and DB2 .............................................. 333
    50.2.1 DB2 Objects ............................................................................................... 334
    50.2.2 Examples .................................................................................................. 338
    50.2.3 Compatibility ............................................................................................ 340
    50.3 ...................................................................................................................... 342
  51 Environment Variables ............................................................................................ 344
  52 DB2 Bind and Grant Jobs ........................................................................................ 346
SAP Database Administration Guide: IBM DB2
Universal Database for z/OS
General Information
1 About this Guide

For the most up-to-date version of this guide, see the documentation in SAP Service Marketplace at

service.sap.com/instguides.

Some sections use information provided by the SAP documentation Planning Guide: z/OS Configuration for SAP on IBM DB2 UDB for z/OS. This documentation is referred to as SAP Planning Guide for z/OS in this documentation.

For more information on functions not described in this documentation, see the SAP Online Documentation. Choose Help → SAP Library from the main menu in your SAP System. Alternatively, you can access the help files on the Online Documentation CD, supplied in the installation package.
2 Required Knowledge

You need to be familiar with the following:

- z/OS (UNIX System Services, JCL, TSO)
- DB2 administration, SQL, SPUFI, DB2 utilities, such as **REORG** and **RUNSTATS**
- Operating system of the application server (AIX, Windows, LINUX or z/OS USS)
- The SAP system (ABAP Dictionary, conversion of tables, CCMS)
## 3 Terminology

The following table lists terms that are used with a specific meaning in this guide.

<table>
<thead>
<tr>
<th>Term</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>DB2 database objects are referred to as <strong>database</strong>.</td>
</tr>
<tr>
<td>DB2</td>
<td>Within this guide, DB2 UDB for z/OS is mostly referred to as <strong>DB2</strong>.</td>
</tr>
<tr>
<td>SAP database</td>
<td>The complete data belonging to one SAP system is called <strong>SAP database</strong>.</td>
</tr>
<tr>
<td>Stogroup</td>
<td>DB2 UDB for z/OS storage groups are called <strong>stogroups</strong> in this guide to distinguish them from SMS storage groups in the Storage Management Subsystem (SMS).</td>
</tr>
</tbody>
</table>
4 New Features

SAP Web Application Server Release 6.40
The following features are changed or new with release 6.40:

- DB2 Version 8 is required for the SAP Web AS Release 6.40. Many new features of DB2 V8 are exploited and described in this guide:
  - Table controlled partitioning
  - System level backup and recovery
  - Automatic extent management
  - Accounting and WLM exploitation
  - Transaction level accounting and workload management
  - IFI enhancements
  - Index only access for VARCHAR

- Unicode is now supported.

- DB2 Connect is used to connect non-z/OS application servers to DB2 (enhancements in the tool db2radm to setup DB2 Connect). ICLI is no longer used. For the Java stack, the DB2 JDBC Universal Driver—which is delivered with DB2 Connect—is used as connectivity.

- Stored procedures have replaced the usage of FTP for JCL job submission.

- New features in the planning calendar (transaction DB13)
  - Exploitation of inline statistics and utilities TEMPLATE, MODIFY RECOVERY and MODIFY STATISTICS
  - Protection of changed utility skeletons possible
  - New design

- Recommendations for DB2 V8 system parameters (ZPARMs).

SAP Web Application Server Release 6.30
The SAP Database Administration Guide Release 6.20 is also valid for use with the SAP Web Application Server Release 6.30.

SAP Web Application Server Release 6.20
The following features are included in this guide for Release 6.20:

- MCOD allows you to place multiple SAP systems in one database. The database administration tools take this into account.

- The DB Performance Monitor uses rfcoscol to gather performance data.

- ICLI Alert Router has been dropped. rfcoscol serves as the database alert router.

- New features of DB2 V7, such as Real Time Statistics and the corresponding stored procedure DSNACCOR are included.

- The work processes of SAP application servers can be actively redirected to a different DB2 system. Transactions ST04 and DB2 enable you to initiate such failovers.

- The SAP system aims to ensure that the DB2 Accounting trace classes 2 and 3 and the performance trace IFCID 318 are always switched on.
• Changes within the database layout
  
  One database can now hold one multi-table tablespace (XSAP) or up to 100 single-
  table tablespaces. This significantly reduces the number of databases needed for an
  SAP system.

**SAP Web Application Server Release 6.10**

The following features are included in this guide for Release 6.10:

• Changes within the database layout:
  
  o LOB tablespaces, auxiliary tables and indexes to support LOB (large object)
    data types
  
  o Schema / creator other than SAPR3

• Modifications to the mass processing tool RSDB2MAS and the PTF check tool

• New transaction DB13C Central DBA Planning Calendar

• New ICLI client and DBSL traces

• DB2 PM no longer supported

• SAP Performance Monitor uses ICLI Performance Services to gather performance
  data.

**SAP R/3 Release 4.6C**

The following features are included in this guide for Release 4.6C:

• Significant modifications to the mass processing tool RSDB2MAS and the PTF check
  tool

• Reduced number of data sets through using DB2’s new storage attribute **DEFINE
  YES/NO**

**SAP R/3 Release 4.6B**

The following features are included in this guide for Release 4.6B:

• Transaction DB2 and central navigation tool

• PTF check tool

**SAP R/3 Release 4.6A**

The following features are included in this guide for Release 4.6A

• Direct adjustment of storage parameters in the database using transaction SE14.
  Instead of a table conversion, SQL statements of the form **ALTER INDEX or ALTER
  TABLESPACE** are performed, if possible.

• Improved DB2 code page support (the special characters “#” and “$” are no longer
  used for tablespace and database names). For more information, see SAP Note
  116750.

• Usage of DB2’s new buffer pools BP8K0 and BP16K0

**SAP R/3 Release 4.5B**

The following features are included in this guide for Release 4.5B

• Report **RSDB2MAS**
  
  This report supports the mass processing function in the ABAP Dictionary. It allows you
  to select:
Empty and non-buffered tables for transfer into a multi-table tablespace
Non-empty, non-buffered tables for transfer into a tablespace of their own.
The selected tables can be entered for mass processing. See the section on "Mass Processing" for more information.

Within transaction SE14:
Move tables to existing tablespaces. See the section "Moving Tables to Existing Tablespaces" for more information.
Directly access index storage parameters. See the subsection "Display/Edit Index Parameters" for more information.
Basic Operations

This section covers basic DB2 administrative tasks.
5 Maintaining the DB2 Subsystem

To maintain the DB2 subsystem that is not part of a data sharing group:

1. Stop all SAP dialog instances
2. Stop the SAP central instance
3. Perform maintenance actions. Stop and start the DB2 subsystem if needed.
4. Start all SAP instances

To maintain a DB2 subsystem that is part of a data sharing group:

5. Redirect the work processes of all SAP instances, which are connected to that DB2 subsystem, to different members. These members need to be capable of handling the additional load.
6. Stop all rfcoscol processes connected to the DB2 subsystem
7. Perform maintenance actions. Stop and start the DB2 subsystem if needed.
8. Redirect the moved work processes back to the DB2 subsystem
9. Start rfcoscol

The redirection of work processes is described in Monitoring and Performance → Database Tools → Proactive Redirection of Application Servers to Other DB2 Subsys [Page 176].
The SAP system provides a central navigation tool, which simplifies access to DB2-specific transactions and reports. You can access the central navigation tool by calling transaction DB2.

The tool is described in the following sections:

- Performance Tuning [Page 21]
- Storage Management [Page 24]
- Checks and Settings [Page 26]
- Traces and Logs [Page 28]
6.1 Performance Tuning

When you call transaction DB2, the screen Database Administration: DB2 UDB for z/OS - Performance Tuning appears.

From this screen, you can select functions related to performance analysis and tuning. The functions are grouped into four areas: Analysis, Tools, Alerts and Network.

Performance Tuning

The following tables give an overview of the functions, their alternative transactions, and give a reference to where you can find more detailed information on these functions in this documentation.

Performance Tuning - Analysis

<table>
<thead>
<tr>
<th>Function</th>
<th>Alternative Transaction</th>
<th>Detailed Information Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Subsystem Activity</td>
<td>ST04</td>
<td>[DB2 Subsystem Activity][Page 159]</td>
</tr>
<tr>
<td>Thread Activity</td>
<td>ST04</td>
<td>[Thread Activity][Page 164]</td>
</tr>
<tr>
<td>Cache Statement Statistics</td>
<td>ST04</td>
<td>[Cached Statements Statistics][Page 166]</td>
</tr>
<tr>
<td>Global Times</td>
<td>ST04</td>
<td>[Global Times][Page 163]</td>
</tr>
<tr>
<td>Data Set Statistics</td>
<td>ST04</td>
<td>[Data Set Statistics][Page 168]</td>
</tr>
<tr>
<td>Installation Parameters</td>
<td>ST04</td>
<td>[Installation Parameters][Page 169]</td>
</tr>
</tbody>
</table>

SAP DBA Guide for z/OS 640
### Performance Tuning - Tools

<table>
<thead>
<tr>
<th>Function</th>
<th>Alternative Transaction</th>
<th>Detailed Information Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Catalog Browser</td>
<td>ST04</td>
<td>DB2 System Catalog [Page 178]</td>
</tr>
<tr>
<td>z/OS System Log</td>
<td>ST04</td>
<td>z/OS System Log [Page 179]</td>
</tr>
<tr>
<td>DB2 Commands</td>
<td>ST04</td>
<td>DB2 Commands [Page 181]</td>
</tr>
<tr>
<td>Buffer Pool Tuning</td>
<td>ST04</td>
<td>Buffer Pool Tuning [Page 182]</td>
</tr>
<tr>
<td>Update Statistics</td>
<td>DB20</td>
<td>Update Statistics</td>
</tr>
<tr>
<td>Explain Statement</td>
<td>ST05</td>
<td>EXPLAIN</td>
</tr>
<tr>
<td>DBA Planning Calendar</td>
<td>DB13</td>
<td>DBA Planning Calendar [Page 266]</td>
</tr>
<tr>
<td>Central Planning Calendar</td>
<td>DB13C</td>
<td>Central DBA Planning Calendar</td>
</tr>
</tbody>
</table>

### Performance Tuning - Alerts

<table>
<thead>
<tr>
<th>Function</th>
<th>Alternative Transaction</th>
<th>Detailed Information Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long-Running TX</td>
<td>ST04</td>
<td>DB Alert Router</td>
</tr>
<tr>
<td>Deadlocks</td>
<td>ST04</td>
<td>DB Alert Router</td>
</tr>
<tr>
<td>Timeouts</td>
<td>ST04</td>
<td>DB Alert Router</td>
</tr>
<tr>
<td>Lock Escalations</td>
<td>ST04</td>
<td>DB Alert Router</td>
</tr>
<tr>
<td>Extents</td>
<td>ST04</td>
<td>Extent Monitor</td>
</tr>
<tr>
<td>Active Log Shortage</td>
<td>ST04</td>
<td>DB Alert Router</td>
</tr>
<tr>
<td>CCMS Monitor Set</td>
<td>RZ20</td>
<td>CCMS Monitor Set</td>
</tr>
</tbody>
</table>

### Performance Tuning - Network

<table>
<thead>
<tr>
<th>Function</th>
<th>Alternative Transaction</th>
<th>Detailed Information Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Sharing Topology</td>
<td>ST04</td>
<td>Database Tools</td>
</tr>
<tr>
<td>Active DB Connections</td>
<td>ST04</td>
<td>Database Tools</td>
</tr>
<tr>
<td>DB Connection List</td>
<td>ST04</td>
<td>Database Tools</td>
</tr>
<tr>
<td>DB Alert Settings</td>
<td>ST04</td>
<td>DB Alert Router</td>
</tr>
<tr>
<td>DB2 Connect Diagnostics</td>
<td>ST04</td>
<td>DB2 Connect Diagnostics</td>
</tr>
<tr>
<td>DB2 Ping</td>
<td>ST04</td>
<td>DB2 Connect Diagnostics</td>
</tr>
</tbody>
</table>
6.2 Storage Management

You call transaction DB2 and select Storage Mgt.

You can select functions on this screen to monitor and change storage attributes.

The functions are grouped into three areas: Database objects, Status, and Tools.

The following tables give an overview of the functions, their alternative transactions, and give a reference to where you can find more detailed information on these functions in this documentation.

### Storage Mgt. - Database Objects

<table>
<thead>
<tr>
<th>Function</th>
<th>Alternative Transaction</th>
<th>Detailed Information Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit</td>
<td>SE14</td>
<td>[Storage Parameters][Page 301]</td>
</tr>
</tbody>
</table>

### Storage Mgt. - Status

<table>
<thead>
<tr>
<th>Function</th>
<th>Alternative Transaction</th>
<th>Detailed Information Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backups</td>
<td>DB12</td>
<td>[Backup and Recovery Options][Page 226]</td>
</tr>
<tr>
<td>Tables and Indexes</td>
<td>DB02</td>
<td>[Tables and Indexes Monitoring][Page 201]</td>
</tr>
</tbody>
</table>

---

**SAP DBA Guide for z/OS**

640

24
### Storage Mgt. - Tools

<table>
<thead>
<tr>
<th>Function</th>
<th>Alternative Transaction</th>
<th>Detailed Information Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBA Planning Calendar</td>
<td>DB13</td>
<td>DBA Planning Calendar [Page 266]</td>
</tr>
<tr>
<td>DB2 Catalog Browser</td>
<td>DB2C</td>
<td>DB2 System Catalog [Page 178]</td>
</tr>
<tr>
<td>DB2 Commands</td>
<td>ST04</td>
<td>DB2 Commands [Page 181]</td>
</tr>
<tr>
<td>Mass Processing</td>
<td>SE38 with RSDB2MAS</td>
<td>Mass Processing [Page 314]</td>
</tr>
<tr>
<td>Empty DB Objects</td>
<td>SE38 with RSDB2CLN</td>
<td>Self-explanatory report</td>
</tr>
</tbody>
</table>
### 6.3 Checks and Settings

You call transaction DB2 and select *Checks/settings*.

This screen combines check utilities and transactions to modify and display settings.

These functions are grouped into four areas: *Checks, DB2 Subsystem Settings, and SAP Settings and DB Performance Monitor*.

---

#### Database Administration: DB2 UDB for z/OS

<table>
<thead>
<tr>
<th>Database Objects</th>
<th>Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>@ Table □ Index</td>
<td>DBA Planning Calendar</td>
</tr>
<tr>
<td></td>
<td>DB2 Catalog Browser</td>
</tr>
<tr>
<td></td>
<td>DB2 Commands</td>
</tr>
<tr>
<td></td>
<td>Mass Processing</td>
</tr>
<tr>
<td></td>
<td>Empty DB objects</td>
</tr>
</tbody>
</table>

---

The following tables contain an overview of the functions, their alternative transactions, and a reference to where you can find more detailed information on these functions in this documentation.

### Checks/Settings - Checks

<table>
<thead>
<tr>
<th>Function</th>
<th>Alternative Transaction</th>
<th>Detailed Information Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dictionary – DB2 Catalog</td>
<td>DB02</td>
<td>Checking Consistency [Page 207]</td>
</tr>
<tr>
<td>Missing Unique Indexes</td>
<td>DB02</td>
<td>Checking Consistency [Page 207]</td>
</tr>
<tr>
<td>DB2 Subsystem</td>
<td>DB16</td>
<td>DB System Check [Page 192]</td>
</tr>
<tr>
<td>Missing Fixes</td>
<td>SE38 with RSDB2FIX</td>
<td>Automated PTF Check [Page 31]</td>
</tr>
<tr>
<td>SAP System Check</td>
<td>SICK</td>
<td>Checking Consistency [Page 207]</td>
</tr>
</tbody>
</table>
### Checks/Settings - DB2 Subsystem Settings

<table>
<thead>
<tr>
<th>Function</th>
<th>Alternative Transaction</th>
<th>Detailed Information Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintain Check Parameters</td>
<td>DB17</td>
<td>DB System Check [Page 192]</td>
</tr>
<tr>
<td>Installation Parameters</td>
<td>ST04</td>
<td>Installation Parameters [Page 169]</td>
</tr>
</tbody>
</table>

### Checks/Settings - SAP Settings

<table>
<thead>
<tr>
<th>Function</th>
<th>Alternative Transaction</th>
<th>Detailed Information Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCL Submission Service</td>
<td>DB2J</td>
<td>JES Interface</td>
</tr>
<tr>
<td>DB Alert Router</td>
<td>ST04</td>
<td>DB Alert Router</td>
</tr>
</tbody>
</table>

### Checks/Settings – DB Performance Monitor

<table>
<thead>
<tr>
<th>Function</th>
<th>Alternative Transaction</th>
<th>Detailed Information Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP Collector Settings</td>
<td></td>
<td>DB Performance Monitor</td>
</tr>
</tbody>
</table>
6.4 Traces and Logs

You call transaction DB2 and select Traces/logs.

From this screen, you can access different traces and logs. This is particularly useful if you have to analyze the cause of short dumps and database problems.

These functions are grouped into two areas: Traces and Logs.

The following tables contain an overview of the functions, their alternative transactions, and a reference to where you can find more detailed information on these functions in this documentation.

**Traces/Logs - Traces**

<table>
<thead>
<tr>
<th>Function</th>
<th>Alternative Transaction</th>
<th>Detailed Information Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Trace</td>
<td>ST01</td>
<td>–</td>
</tr>
<tr>
<td>DB2 Connect</td>
<td>ST04</td>
<td>DB2 Connect Diagnostics</td>
</tr>
<tr>
<td>ABAP</td>
<td>SE30</td>
<td>–</td>
</tr>
<tr>
<td>SQL</td>
<td>ST05</td>
<td>–</td>
</tr>
<tr>
<td>DBSL</td>
<td>–</td>
<td>DBSL Trace [Page 193]</td>
</tr>
<tr>
<td>IFI Data Collector</td>
<td>–</td>
<td>Performance Monitoring</td>
</tr>
<tr>
<td>IFI DB Trace</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

DB2 Connect, DBSL and IFI DB Trace should only be used by experienced administrators. Tracing can considerably impact the performance of the entire system.

They allow you to trace each SQL statement as processed on different software layers; the DBSL Trace in the SAP db shared library, the DB2 Connect CLI Trace in the DB2 Connect CLI layer, IFI DB Trace on DB2 level.
### Traces/Logs - Logs

<table>
<thead>
<tr>
<th>Function</th>
<th>Alternative Transaction</th>
<th>Detailed Information Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP System Log</td>
<td>SM21</td>
<td>--</td>
</tr>
<tr>
<td>z/OS System Log</td>
<td>ST04</td>
<td>[z/OS System Log][Page 179]</td>
</tr>
</tbody>
</table>
7 Authorization Profiles

The following authorization profiles are delivered as standard for particular tasks in the table and index monitor and the CCMS topics.

<table>
<thead>
<tr>
<th>Authorization Profile</th>
<th>Authorizations Permitted</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_DB2_DBADM</td>
<td>With this authorization, you can:</td>
</tr>
<tr>
<td></td>
<td>• Execute an <code>ALTER</code> on the secondary quantity of a tablespace or index (tables and indexes monitor)</td>
</tr>
<tr>
<td></td>
<td>• Change and delete the JCL Jobs of any user (JES Interface)</td>
</tr>
<tr>
<td></td>
<td>• Change the TSO password of any user (JES Interface)</td>
</tr>
<tr>
<td></td>
<td>• Execute all DB2 commands, create, change and delete new commands, (SAP performance monitor)</td>
</tr>
<tr>
<td></td>
<td>• Execute <code>SELECT</code> on DB2 catalog tables (DB2 catalog browser)</td>
</tr>
<tr>
<td></td>
<td>• Switch an application server to a different DB2 data sharing member</td>
</tr>
<tr>
<td>S_DB2_COMM</td>
<td>With this authorization, you can:</td>
</tr>
<tr>
<td></td>
<td>• Execute, change and delete all DB2 commands, create new commands (SAP performance monitor)</td>
</tr>
<tr>
<td>S_DB2_EXPC</td>
<td>With this authorization, you can:</td>
</tr>
<tr>
<td></td>
<td>• Execute all DB2 commands, create, change and delete new commands, (SAP performance monitor) that have the command user <code>ALLUSER</code></td>
</tr>
<tr>
<td>S_DB2_ALLU</td>
<td>With this authorization, you can:</td>
</tr>
<tr>
<td></td>
<td>• Execute all DB2 commands that have the command user <code>ALLUSER</code> (SAP performance monitor)</td>
</tr>
</tbody>
</table>

You call transaction SU01 to give the appropriate authorization profile to a user.
8 Automated PTF Check

The following sections provide you with information on the automated PTF check:

- PTF Check Overview [Page 32]
- Additional Information Sources [Page 33]
- Technical Details [Page 34]
- Requirements [External]
- PTF Check Setup [Page 36]
- Performing the Check [Page 37]
- Output Analysis [Page 40]
- PTF Check Troubleshooting [Page 41]
- PTF Check Recommendations [Page 42]
8.1 PTF Check Overview

It can be time-consuming to check whether all Authorized Program Analysis Reports (APARs) and Program Temporary Fixes (PTFs) required in SAP Note 81737 have been applied to a z/OS system. You can simplify this task by using a tool that automatically performs all of the following steps:

1. Determination of the release and/or version of all software components (SAP System, SAP kernel, z/OS system, and DB2 subsystem)
2. Extraction of all required PTFs from SAP Note 81737 and current fix level file
3. Determination of the status of all required PTFs within the z/OS system
4. Output of missing PTFs and/or Function Module IDs (FMID)
8.2 Additional Information Sources

The following references provide additional information:

- **SAP Note 81737** lists all PTFs required.
- **SAP Note 183311** covers updates on the PTF check.
- The installation process of rfcoscol is described in Manual Installation of RFCOSCOL on z/OS [Page 142].
8.3 Technical Details

The automated PTF check is based on the assumption that customers administer all z/OS software components using IBM’s System Modification Program Extended (SMP/E). This z/OS program keeps a record of all changes (for example, PTFs) to function modules in the Consolidated Software Inventory (CSI).

SMP/E also provides an interface (GIMAPI) that can be called by application programs to query the contents of the CSI. For more information on SMP/E, see the IBM documentation SMP/E Reference and SMP/E User’s Guide).

rfcoscol by SAP is able to connect to the GIMAPI interface and forward the SMP/E data to any SAP System that is linked to rfcoscol. The PTF check itself is performed by report RSDB2FIX and runs on an SAP System that is called check system in this documentation. The system to be checked is referred to as target system. The target and check system may be identical. However, you can also just choose one single SAP System to be the check system for all other SAP installations at the customer site. This is feasible because the check system connects to the target system and its z/OS host using remote function calls (RFC). Also, it is not required that the check system runs on DB2.

The technical details of the PTF check are illustrated in the following graphic.

PTF Check

SAP Note 81737 is formatted in such a way that it can be used directly as input for the check report. You only have to download it to the application server of the check system. Additionally, SAP and IBM provide a file called fix level file on sapservX that contains a list of PTFs related to the latest put levels. Both files are uploaded by RSDB2FIX.

Alternatively, the check tool is able to automatically retrieve the most recent PTF information kept in SAP Note 81737 and the fix level file directly from SAPNet – SAP Frontend (formerly OSS) if a valid RFC connection exists. In that case, a download of these files to your PC or the application server is not required.
The target system’s kernel release as well as the versions of the DB2 and z/OS software used are determined via RFC. This information combined with the uploaded PTF information (SAP Note 81737 and fix level file) is subsequently used by RSDB2FIX to retrieve a list of required PTFs and FMIDs.

Finally, RSDB2FIX queries the status of each required FMID and PTF employing the rfcoscol connection to GIMAPI and SMP/E. PTFs that are not found with status “applied” or “superseded” are listed as “missing” in the output.

APARs that are not checked, because none of the associated FMIDs can be located in the given CSIs, are also written to the output.
8.4 PTF Check Setup

Before starting the PTF check you have to set up your environment as follows:

1. Check the SMP/E settings.
   The PTF check tool can only check entries in CSI. You have to make sure that these CSI entries reflect exactly the status of the software that is actually running.

2. Select one of your SAP systems as the check system to perform the PTF check.

3. Update the check report RSDB2FIX in your check system if necessary. For more information, see SAP Note 183311.

4. Install and start rfcoscol on z/OS and establish an RFC connection between the check system and rfcoscol. For details, see Manual Installation of RFCOSCOL on z/OS [Page 142].

5. Establish a connection to SAPNet – R/3 Frontend. (Optional)
   a. Log on to the check system and call transaction OSS1.
   b. Choose Parameters → Technical Settings.
   c. Specify and save the Logon settings.
   d. Choose Log on, specify group 1_PUBLIC, and check whether the connection works.
### 8.5 Performing the Check

Once you have completed the preparations described in section [PTF Check Setup](#), the PTF check can be performed.

1. If the check system is not able to connect to SAPNet, you need to transfer **SAP Note 81737** and the current fix level file to a directory on the application server of the check system:
   - a. From SAPNet – R/3 Frontend, display the English version of **SAP Note 81737**.
      - RSDB2FIX cannot process a note file download from SAP Service Marketplace.
   - b. Choose *Note administration*.
   - c. Choose *Note → Download*.
   - d. Choose *Do not copy*.
   - e. Specify data format ASC and file name on your presentation server.
   - f. Choose *transfer*.
   - g. Copy the downloaded file from your presentation server to the central instance of the check system (use FTP with transfer type **ASCII**).
   - h. Download the file *fix<YYMMDD>.txt* from sapservX directory ~ftp/general/R3server/abap/note.0183311 to the application server of the check system (use FTP with transfer type **ASCII**). It contains in machine readable format a list of all PTFs related to the current required service levels.

2. To access the PTF check tool, call one of the following transactions:
   - o Transaction DB2 and choose *Checks/Settings → Missing Fixes*.
   - o Transaction SE38 and execute report RSDB2FIX.

The input screen of check report RSDB2FIX appears.
Fix Check

DB2/390: Fix check

File and Log Names
- SAP Note 81137
- Fix Level File
- Log Name: FIX-M364D DATE & TIME
  - Display Logs

RFC Settings
- SAP System (F3)
- z/OS Host (TCP/IP)
- SAPNet (R3)
  - SAP I/OS
  - RFC Connections (SM19)
  - Online Service System (CO81)

SNAPI Settings
- Data Set
- Zone
  - 1.CSILibrary
  - 2.CSILibrary
  - 3.CSILibrary
  - 4.CSILibrary
  - 5.CSILibrary
  - 6.CSILibrary
  - 7.CSILibrary
  - 8.CSILibrary
3. Enter the following input values:

### Fix Check: Input Screen Values

<table>
<thead>
<tr>
<th>File Name or Setting</th>
<th>Input Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SAP Note 81737</strong></td>
<td>Full path name of the file on the application server that contains SAP Note 81737. See Performing the Check [Page 37], step 1.g.</td>
</tr>
<tr>
<td><strong>Fix level file (Optional)</strong></td>
<td>Full path name of the file on the application server containing the fix level information. See Performing the Check [Page 37], step 1.h.</td>
</tr>
<tr>
<td><strong>Log Name</strong></td>
<td>Name of the log to which the output is written. The pattern &amp;R3&amp;, &amp;DATE&amp; and &amp;TIME&amp; are substituted by the name of the SAP system destination, the date and the time respectively.</td>
</tr>
<tr>
<td><strong>SAPNet</strong></td>
<td>SAPNet’s RFC destination. The default connection SAPOSS is created by using transaction OSS1 (see PTF Check Setup [Page 36], step 9). The transaction can be accessed directly by choosing Online Service System (OSS1).</td>
</tr>
<tr>
<td><strong>SMP/E Settings</strong></td>
<td>The input depends on how SMP/E is configured in your environment. Specify the data set and the zone for at least one CSI library. The sample input in the Fix Check: Input Screen graphic represents an environment where all the DB2 function modules are administered in data set SMPE.DB261.CSI and zone TDB261, whereas the remaining software components (z/OS USS, JES3, VTAM, and so on) are kept in data set SMPE.OS390.CSI and zone TOS390. You cannot check GLOBAL or DLIB zones.</td>
</tr>
</tbody>
</table>

4. Choose Ping to check whether the associated RFC connection works.

5. Save the input as a variant (CTRL-S; use the target system ID as variant name). The PTF check can then be easily repeated.

6. Make sure that no other PTF check is currently running on the target system. If you run PTF checks in parallel, there is no risk of damaging the SMP/E data. However, the results may be incorrect. You can execute report RSDB2FIX online.

7. For output, you can display the result logs by choosing Display logs within the initial selection screen of RSDB2FIX. If RSDB2FIX is executed online, the result log is displayed directly after the check.
8.6 Output Analysis

You analyze the output of the RSDB2FIX program.

All errors, warnings, and check results are reported to the output. If the report completes successfully, you find a list of missing PTFs and FMIDs at the end of the output.

- Below the section Check PTFs the following output might appear:
  - No missing PTFs found.
    All PTFs required for the FMIDs found within the given SMP/E settings have been applied. Nothing needs to be done. However, if the fix level file was not used, only a subset of all required PTFs is checked and there may still be a number of missing PTFs.
  - The following PTFs are missing.
    A list providing information on the missing PTFs and their associated FMID and APAR is given. The list is ordered by FMID and APAR.
    Check the status of these missing PTFs. Maybe they are only needed under certain circumstances. For example, the additional **Required for...** indicates that a PTF is only needed for certain SAP system releases.
    For more information, see SAP Note 81737. Otherwise, apply the PTFs to your z/OS system.

- Below the section Check FMIDs the following output may appear:
  - All APARs checked.
    This means that all APARs (and associated PTFs) could be checked. The SMP/E settings specified in the input screen is complete. Be aware that if the fix level file was not used, only a subset of all required APARs and PTFs is checked.
  - The following APARs cannot be checked because none of the associated FMIDs could be located in the specified CSIs.
    The following list contains all APARs that could not be checked due to the fact that none of the associated FMIDs is active in the given SMP/E settings. It is possible that the APAR refers to a product that is not installed in your environment. This is, for example, the case if you use JES2 and the APAR is related to JES3.
    Check whether your input to RSDB2FIX is incomplete. If this is the case, you should correct the SMP/E settings on the input screen and run RSDB2FIX again.
8.7 PTF Check Troubleshooting

The following list can help you to solve some of the problems that might occur when executing the RSDB2FIX program:

- **Error message:** *Report RSDB2FIX is outdated Please obtain current version. See SAP Note 183311 for details.*
  A hot package is provided.

- **Error message:** *Version of SAP Note 0081737 is outdated. Please use current version.*
  Download the latest version of SAP Note 81737 and use it as input.

- **Error message:** *SMP/E API failed with GIM59605S ** ENQ FAILED OR SHARED USE OF. FOR QUERY PROCESSING. RSDB2FIX could not access SMP/E due to an SMP/E job or user session running in parallel.*

- **Error message:** *SMP/E API failed with GIM44240I GIMVSMSG - THE VSAM ERROR ANALYSIS OCCURRED.*
  Give read access for SMP/E’s CSI data sets to the user that starts RFCOSCOL.
8.8 PTF Check Recommendations

Uploading Input Files from PC
If RSBD2FIX is executed online, the input files can also be uploaded from the presentation server. In that case, you have to download SAP Note 81737 and fix level file to the presentation server and specify the input parameters accordingly.

Omitting Fix Level File
Usually it is sufficient to check the put levels contained in file fix<YYMMDD>.txt only once. After making sure that all PTFs belonging to certain put levels have been applied, you can reduce the PTF check to the PTFs listed in SAP Note 81737. In that case, leave input parameter Fix level file empty. This will reduce execution time.

Scheduling PTF Checks Regularly
Consider defining a background job within transaction SM36 that checks all your systems on a regular basis, for instance once a month. Each target system that is defined as a variant of report RSDB2FIX can form a step within this job. However, you must make sure that RSDB2FIX always uploads the latest versions of SAP Note 81737 and the fix level file. This is guaranteed if RSDB2FIX uploads the information directly from SAPNet.

Checking PTF Level before an Upgrade
The PTF check does not depend on the system’s SAP system release. Therefore, use RSDB2FIX before upgrading a system to check whether all PTFs required for the target release are applied.

Handling a Data Sharing Group
For a data sharing group, two cases can be distinguished. If the software level is identical for all members of a data sharing (DS) group and is managed within one central SMP/E CSI, it is sufficient to perform the PTF check for one member. Otherwise, you must perform a PTF check for each member of the DS group. To do this:

1. Configure and start rfcoscol’s on each DS member.
2. For all rfcoscol’s, establish a dedicated TCP/IP connection (using transaction SM59) to the check system.
3. Execute RSDB2FIX for each DS member. You must specify the TCP/IP connection and the CSI libraries individually for each DS member.

PTF checks cannot be run in parallel.
9 Other Operations and Considerations
9.1 Database Access

To ensure data consistency, all write access to the database must use the SAP interfaces. Native write access is not allowed and can destroy the data consistency.

Native read access is allowed. Confidential data is encrypted (for example, salaries).

⚠️ If you intend to read data with an isolation level that causes locks to be requested (that is, CS, RS, RR), be aware that you might cause contention problems in the SAP system. Even a reader using isolation level UR (uncommitted read) might cause problems in the SAP data dictionary area.
9.2 Hardware Failure

Plans to reduce the effects of unplanned downtime due to hardware, software, or communication failure must be implemented. These failures can range from relatively minor incidents to major disasters, for instance:

- Database processor failure
- DASD or disk failure
- User data error
- Application server connection failure
- Database server/application server connection failure
- Failed processor

In the SAP system environment, planning for these types of failures should be similar to conventional failure recovery planning, taking the following into consideration: Due to the highly integrated nature of the SAP data, any database recovery that is triggered by any type of failure must be performed with care in order to ensure that the recovered data is logically consistent. In a point-in-time recovery, any updated data after the target point-in-time will be lost.

See also:


IBM Documentation High Availability for SAP on zSeries Using Autonomic Computing Technologies.
9.3 DB2 Schema

Throughout this guide, the variable `<SCHEMA>` is used for the DB2 schema of the DB2 objects which belong to the SAP system. In older releases, the hard-coded name `SAPR3` was used.
9.4 MCOD and CCMS

In an MCOD installation, SAP systems let other SAP systems in the same DB2 subsystem automatically know about their existence. This is necessary to ensure proper functioning of database backup jobs that are accomplished via transaction DB13 and accurate space statistics that are provided in transaction DB02. During startup, SAP systems contact and notify the other systems. This function is embedded in RSDB2_COLLECT_HOURLY, a program that runs every hour. The other SAP systems do not have to be started for this notification to take place. SAP systems that are embedded in the DB2 subsystem at a later stage are also immediately advised of this system. If necessary, for example, if the program does not run every hour, you can announce an SAP system manually by calling transaction DB2J.
DB2 Setup

Setting up the DB2 subsystem is an important step in the preparation of an SAP system installation. There is a large number of parameters that have a significant impact on the overall performance and operations of DB2. The following sections only cover those parameters with a special significance for the SAP system. For a complete list and recommendations on which values the parameters should have in your environment, refer to the IBM documentation DB2 for z/OS Installation Guide.

Prerequisites

Make sure you fulfill the following requirements before you begin installing the DB2 subsystem:

- DB2 for z/OS Version 8 (running in “New Function” mode) or later
- DB2 Utilities Suite for z/OS
- Control Center for DB2 Universal Database for z/OS and OS/390 (FMID JDB881D)
- RRS needs to be installed and set up. For more information, see Planning Guide: z/OS Configuration for SAP on IBM DB2 UDB for z/OS.

A coupling facility is not required to implement the RRS system logger. You can use a DASD-only log stream instead.

For additional information, refer to the following IBM documentation:
- For DB2 data-sharing in a SYSPLEX, you must use a coupling facility.
- Stored procedures [Page 64] must be enabled.
- Service Level:

  In general, we recommend that the service levels of DB2 and RRS are at the latest level possible. Contact your IBM representative for more information about service levels. Some PTFs are required for this SAP system release. For the list of PTFs, see SAP Note 81737.

  We provide an easy-to-use tool to check the PTF status. If you already have an SAP system installed on site, you can use it to check whether all required PTFs have been applied (even if the SAP system is not running on DB2). For more information, see SAP Note 183311 or Automated PTF Check [Page 31].
10 Creating the DB2 Subsystem

Since SAP systems are relatively complex and a large number of objects is associated with them, the best choice is to keep non-SAP applications out of DB2 subsystems that are dedicated to SAP systems. For information on considerations for possible system layouts, see the SAP Planning Guide, section “DB2 subsystem and z/OS system”. This gives you the following benefits:

- DB2 parameters can be set to values that ensure efficient SAP system operations
- Resource consumption can be monitored more easily
- Sizing estimates are simplified
- Security handling is facilitated
- Backup and recovery procedures can be implemented more easily
- Be aware that placing multiple SAP systems into the same DB2 subsystem increases the storage requirements. The section Virtual Storage Considerations [Page 98] discusses the storage requirements of individual SAP systems below the 2 GB bar of the DBM1 address space of DB2, which must not be reached. If each SAP system is served by a dedicated data sharing member, you can benefit from the advantages offered by a single database while avoiding any subsystem storage ramifications.
For optimal operations and performance, SAP systems require specific DB2 system parameter settings. These are divided into the following categories:

- **Required settings**
  These settings are necessary for a proper functioning of SAP systems and must not be changed.

- **Highly recommended settings**
  Although they do not influence SAP system functions, these settings are very important for ensuring optimal performance and should not be changed. For example, the setting NPGTHRSH=10 ensures good access paths for statements that access volatile tables.

- **Recommended as initial settings**
  These values are in most cases sufficient when you start using the SAP system. However, they should be adjusted based on the user-specific characteristics of the SAP system.

Note that a single set of recommendations is provided that applies to all SAP systems including BW systems, which facilitates the setup.

Once you have installed the SAP system, the settings can be checked using transaction DB16.

The parameters listed in the following sections have special significance for SAP systems, but they only represent a small subset of all available DB2 system parameters. For a complete list of these parameters, refer to the IBM documentation *DB2 UDB for z/OS Installation Guide*.

We strongly recommend that you use the default values for any DB2 system parameter not explicitly mentioned in the following sections.
# 11.1 DB2 System Parameters with Required Settings

The settings given in the following table are mandatory. They are needed for the SAP system to function properly and must **not** be changed.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCCSID</td>
<td>819</td>
</tr>
<tr>
<td>SCCSID</td>
<td>Can be set to any value where DB2 supports bijective translation between SCCSID and ASCCSID and that supports the EBCDIC invariant character set (for example, 37 is such a CCSID). Note that a value of 500 for SCCSID is intended for Switzerland and Belgium only. The standard SCCSIDs that are NOT supported are 290, 420, 905, and 1026. Of course, SCCSID should not be changed once the SAP database is loaded. If you need to change SCCSID, contact IBM DB2 support for assistance.</td>
</tr>
<tr>
<td>DECIMAL</td>
<td>. (Always use a period (.) and never a comma (,))</td>
</tr>
<tr>
<td>CTHREAD</td>
<td>The setting of CTHREAD is highly dependent on the overall environment size of the SAP systems. CTHREAD should be at least equal to the sum of TSO users and IDBACK.</td>
</tr>
<tr>
<td>IDBACK</td>
<td>The setting of IDBACK is highly dependent on the overall environment size of the SAP system. IDBACK should be at least 30% higher than the sum of the following values: Number of SAP work processes on all SAP Web application servers that run on z/OS and that are connected to this DB2 subsystem. The number of rfcoscol * 2. IDBACK should have a value of at least 20.</td>
</tr>
<tr>
<td>MAXDBAT</td>
<td>The setting of MAXDBAT is highly dependent on the overall environment size of the SAP system. MAXDBAT should be at least 30% higher than the sum of the number of work processes and secondary connections on all SAP application servers that are connected to this DB2 subsystem via DRDA. Note that the work processes that serve BW open a secondary connection by default. This calculation excludes SAP application servers on z/OS. During installation, MAXDBAT should have a value of at least 64.</td>
</tr>
<tr>
<td>CONDBAT</td>
<td>Set CONDBAT either to the same or a larger value than MAXDBAT.</td>
</tr>
<tr>
<td>IDTHTOIN</td>
<td>Set IDTHTOIN (idle thread timeout) to 0 to disable time-out processing.</td>
</tr>
</tbody>
</table>
## 11.2 DB2 System Parameters with Highly Recommended Settings

The settings listed in the following table are highly recommended to ensure that SAP operates most efficiently. They should not be changed.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMLKUS</td>
<td>2097152</td>
<td>Sets a limit on the number of locks that any individual DB2 thread can hold. Once the limit is reached, the program that accumulated these locks will terminate with sqlcode –904. The maximum value for NUMLKUS is 2097152. This is recommended as an initial, first-cut value. Setting a lower value for NUMLKUS helps you to detect offending programs earlier and is especially recommended for test systems. In most production systems (except the Retail component), a lower value for NUMLKUS is acceptable, but it should not be lower than 500000. For some exceptions and more details, see Locking Considerations [Page 125] in Performance Tuning Considerations.</td>
</tr>
<tr>
<td>IRLMRWT</td>
<td>600</td>
<td>SAP applications are written so as to minimize database lock contentions. However, even when such a contention happens, SAP prefers a long wait rather than a quick conflict resolution that results in a potentially lengthy rollback.</td>
</tr>
<tr>
<td>DEADLOK</td>
<td>5,1</td>
<td>Belongs to the IRLM startup procedure. Lower values increase the likelihood of IBM latch connections.</td>
</tr>
<tr>
<td>EVALUNC</td>
<td>YES</td>
<td>Setting this parameter to YES reduces lock contention caused by locking of non-qualifying rows.</td>
</tr>
<tr>
<td>CONTSTOR</td>
<td>YES</td>
<td>Caching dynamically prepared statements places a significant demand on the virtual storage in the DB2 address spaces. By turning this parameter on, you ensure that the unused storage is contracted on a regular basis, thus improving storage utilization.</td>
</tr>
<tr>
<td>MONSIZE</td>
<td>500000</td>
<td>This parameter sets the OP buffer size (in bytes). The buffer is used by the statistics trace started for the RFCOSCOL alert router. By tuning this parameter, you ensure that all exception events can be held in the OP buffer until the RFCOSCOL alert router reads the buffer the next time. If the specified buffer size is not large enough, exception event loss occurs when the buffer fills before the RFCOSCOL alert router can obtain the exceptions.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NPGTHRSH</td>
<td>10</td>
<td>This parameter is inputted to the access path selection process. If NPAGES for a given table is less than the parameter value and not -1, it is better to have an index access path than a tablespace scan.</td>
</tr>
<tr>
<td>LOBVALA</td>
<td>$1000000k$</td>
<td>The size of the user storage for LOB values (in KB). The recommended value is 1 GB.</td>
</tr>
<tr>
<td>LOBVALS</td>
<td>$50000M$</td>
<td>The size of the system storage for LOB values (in MB). The recommended value is 50 GB.</td>
</tr>
<tr>
<td>STARJOIN</td>
<td>2</td>
<td>Enables a join type called star join for accessing data in a star schema.</td>
</tr>
<tr>
<td>SJTABLES</td>
<td>4</td>
<td>When star join is generally enabled by means of the STARJOIN parameter, it will be considered only if the number of tables is greater than or equal to the value specified in SJTABLES.</td>
</tr>
<tr>
<td>TABLES_JOINED_THRESHOLD</td>
<td>10</td>
<td>For SQL statements that access more than TABLES_JOINED_THRESHOLD tables, statement preparation resources are restricted as governed by the parameters MAX_OPT_CPU, MAX_OPT_ELAP and MAX_OPT_STOR. This is a hidden keyword ZPARAM.</td>
</tr>
<tr>
<td>MAX_OPT_CPU</td>
<td>2</td>
<td>Controls statement preparation resources in terms of CPU consumption (specified in seconds).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is a hidden keyword ZPARAM.</td>
</tr>
<tr>
<td>MAX_OPT_ELAP</td>
<td>4</td>
<td>Controls statement preparation resources in terms of elapsed time (specified in seconds).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is a hidden keyword ZPARAM.</td>
</tr>
<tr>
<td>SJMXPOOL</td>
<td>$128$ MB</td>
<td>Size of the star join pool. If available real storage allows it, increase the size of the star join pool up to 256 MB.</td>
</tr>
<tr>
<td>MGEXTSZ</td>
<td>YES</td>
<td>Allows DB2 to optimize extent sizing.</td>
</tr>
</tbody>
</table>
## 11.3 DB2 System Parameters with Recommended Settings

The DB2 system parameter settings listed in the following table are recommended as initial settings that should be adjusted based on the particular workload as a result of performance monitoring and tuning.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDMSTMTC</td>
<td>300000</td>
<td>Specifies the size (in KB) of the EDM statement cache.</td>
</tr>
<tr>
<td>MAXKEEPD</td>
<td>8000</td>
<td>For more information, see <a href="#">Virtual Storage Considerations</a> in Performance Tuning Considerations.</td>
</tr>
<tr>
<td>SRTPOOL</td>
<td>28000</td>
<td>Amount of storage (in KB) needed for the sort pool.</td>
</tr>
<tr>
<td>MAXRBLK</td>
<td>100000</td>
<td>The maximum size for RID List processing should be 100 MB. This value can be increased if there is enough real storage available.</td>
</tr>
<tr>
<td>CHKFREQ</td>
<td>10</td>
<td>A value of 10 denotes that DB2 takes a checkpoint every 10 minutes.</td>
</tr>
<tr>
<td>RETLWAIT</td>
<td>1</td>
<td>Applies to data sharing only. It is recommended that you wait for retained locks rather than receive a resource unavailable message immediately.</td>
</tr>
<tr>
<td>UTIMOUT</td>
<td>3</td>
<td>The value is reduced from its default (6) due to the relatively large timeout (IRLMRWT) value.</td>
</tr>
<tr>
<td>URCHKTH</td>
<td>1</td>
<td>Some SAP system processes commit very seldom, which can cause a number of problems. In some cases, there is not much that can be done about it because changing the application logic is too difficult. However, in the case of user-written programs, the appropriate changes are often feasible and the programs should be amended by inserting regular commits. This parameter enables you to identify such programs. As the frequency of messages identifying long running units of recovery is directly proportional to the CHKFREQ value, adjust URCHKTH to avoid too-frequent occurrences.</td>
</tr>
<tr>
<td>URLGWTH</td>
<td>100</td>
<td>Additional threshold for identifying long-running, non-committing transactions and reports. The value is in KB.</td>
</tr>
<tr>
<td>XLKUPDLT</td>
<td>YES</td>
<td>This parameter slightly reduces the overhead of acquiring locks for some statements.</td>
</tr>
<tr>
<td>SMFACCT</td>
<td>1, 2, 3</td>
<td>The Accounting Classes 2 and 3 provide valuable performance indicators. The overhead of Class 2 varies, it can be significant during major data imports (for example, SAP installation, migration or upgrade) and should therefore be deactivated at such times. However, after installation, especially during performance monitoring and tuning, it should be</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SYNCVAL</td>
<td>30</td>
<td>Activated to facilitate efficient monitoring.</td>
</tr>
<tr>
<td>SEQCACH</td>
<td>SEQ</td>
<td>Controls the synchronization of statistics recording across a data sharing group.</td>
</tr>
<tr>
<td>LBACKOUT</td>
<td>NO</td>
<td>Enables postponing backouts for long running units of recovery at restart.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>During the SAP application server startup, the application server accesses a large number of tables. Therefore, the overall DB2 subsystem should be in a consistent state once the application servers have been started. Setting LBACKOUT to NO ensures this, because it causes DB2 to completely process the backward log during DB2 restart. If the objects that need to be recovered are known and are not crucial for the functioning of the SAP system, LBACKOUT can also be set to AUTO or YES, which postpones some backward log processing. This reduces DB2 restart time and particularly makes those DB2 objects available earlier that do not need to be recovered.</td>
</tr>
<tr>
<td>PARAMDEG</td>
<td>see explanation</td>
<td>Sets the limit to the maximum degree of parallelism with which a query can be executed. SAP systems can use query parallelism in a very controlled manner (for some components and some selected statements only) and can turn it on explicitly (on a statement level). We recommend limiting the maximum degree of parallelism to the number of available CPUs.</td>
</tr>
<tr>
<td>PCLOSET</td>
<td>25</td>
<td>Indicates how many minutes will elapse after a page set or partition has been updated before DB2 converts the page set or partition from read-write to read-only state. This parameter is used in conjunction with PCLOSEN. If the condition for PCLOSEN or PCLOSET is met, the page set or partition is converted from read-write to read-only state. Having DB2 switch an infrequently updated page set from read-write to read-only state can result in performance benefits for recovery, logging, and data sharing processing.</td>
</tr>
<tr>
<td>PCLOSEN</td>
<td>15</td>
<td>Indicates how many checkpoints will be taken after a page set or partition has been updated before DB2 converts the page set or partition from read-write to read-only state. This parameter is used in conjunction with PCLOSET. If the condition for PCLOSEN or PCLOSET is met, the page set or partition is converted from read-write to read-only state. Having DB2 switch an infrequently updated page set from read-write to read-only state can result in performance benefits for recovery, logging, and data sharing processing.</td>
</tr>
<tr>
<td>MLMT</td>
<td>4</td>
<td>Belongs to the IRLM startup procedure. Specifies the maximum amount of private storage available that IRLM uses for its locks. The unit is GB. Ensure that the IRLM private address space is backed by real storage.</td>
</tr>
<tr>
<td>DDF</td>
<td>AUTO</td>
<td>Automatically starts the DDF address space</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>-------------------------------------------</td>
</tr>
</tbody>
</table>
| ACCUMACC | 5 | Accumulates accounting records with identical values for the following identifiers before externalizing the accumulated record:  
- Client user ID  
- Workstation name  
- Transaction name |
| LRDRTHLD | 10 | Threshold that controls the identification of long-running readers. |
11.4 DB2 Buffer Pool-Related Parameters with Recommended Settings

The following tables give recommended settings for DB2 buffer pool-related parameters. The first table contains recommendations for the initial buffer pool settings of DB2 subsystems that serve transactional SAP systems without BW functionality. The second table contains the recommendations for DB2 subsystems that are dedicated to SAP BW systems. If a DB2 subsystem serves both transactional workload and BW, the recommendation is to combine the recommended buffer pool settings from both tables.

Do not overcommit real storage. Allocate as large buffer pools as real storage allows it.

The buffer pool attribute PGFIX may only be set to YES if sufficient real storage is available.

DB2 Buffer Pool-Related Parameters with Recommended Settings for Transactional SAP Systems Without BW Functionality

<table>
<thead>
<tr>
<th>Buffer Pool</th>
<th>VPSIZE</th>
<th>VPSEQT</th>
<th>DWQT</th>
<th>VDWQT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP0 (catalog)</td>
<td>2000</td>
<td>50</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>BP1 (work files)</td>
<td>15000</td>
<td>100</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>BP2 (4KB tablespaces)</td>
<td>60000</td>
<td>50</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>BP3 (index spaces)</td>
<td>90000</td>
<td>40</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>BP4 (VB protocol)</td>
<td>1000</td>
<td>10</td>
<td>70</td>
<td>50</td>
</tr>
<tr>
<td>BP40 (LOB tablespaces)</td>
<td>4000</td>
<td>50</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>BP8KB0 (8KB tablespaces)</td>
<td>3000</td>
<td>50</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>BP16KB0</td>
<td>3000</td>
<td>50</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>BP32K (32KB tablespaces)</td>
<td>5500</td>
<td>50</td>
<td>30</td>
<td>5</td>
</tr>
</tbody>
</table>

DB2 Buffer Pool-Related Parameters with Recommended Settings for SAP BW Systems

<table>
<thead>
<tr>
<th>Buffer Pool</th>
<th>VPSIZE</th>
<th>VPSEQT</th>
<th>VPPSEQT</th>
<th>DWQT</th>
<th>VDWQT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP0 (catalog)</td>
<td>2000</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>BP1 (work files)</td>
<td>40000</td>
<td>100</td>
<td>50</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>BP2 (4K tablespaces)</td>
<td>15000</td>
<td>50</td>
<td>50</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>BP3 (index spaces)</td>
<td>20000</td>
<td>40</td>
<td>50</td>
<td>30</td>
<td>5</td>
</tr>
</tbody>
</table>
BP6 dim tablespaces
(DB2_DIM_BPOOL_DATA) 10000 30 50 30 5

BP7 dim indexspaces
(DB2_DIM_BPOOL_INDEX) 15000 30 50 30 5

BP8 Fact tablespaces
(DB2_FACT_BPOOL_DATA) 12000 50 50 30 5

BP9 Fact indexspaces
(DB2_FACT_BPOOL_INDEX) 12000 50 50 30 5

BP10 Aggr tablespaces
(DB2_AGGR_BPOOL_DATA) 20000 50 50 30 5

BP11 Aggr indexspaces
(DB2_AGGR_BPOOL_INDEX) 20000 50 50 30 5

BP40 (LOB tablespaces) 4000 50 50 50 10

BP8KB0 (8KB tablespaces) 3000 50 50 30 5

BP16KB0 (16KB tablespaces) 3000 50 50 30 5

BP32K (32KB tablespaces) 5500 50 50 30 5

The parameters DB2 DIM BPOOL DATA, DB2 DIM BPOOL INDEX, DB2 FACT BPOOL DATA and DB2 FACT BPOOL INDEX determine the buffer pools that are employed by the page sets of star schema fact and dimension tables. They are maintained in table RSADMIN (see SAP Note 536074).
11.5 DB2 Installation Panel DSNTIPD: Recommended Values

The values you supply on this panel are estimates used in calculating sizes for main storage and data sets.

**Installation Panel DSNTIPD**

<table>
<thead>
<tr>
<th>Field</th>
<th>Recommended Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATABASES</td>
<td>5000</td>
</tr>
<tr>
<td>TABLES</td>
<td>100</td>
</tr>
<tr>
<td>COLUMNS</td>
<td>20</td>
</tr>
<tr>
<td>VIEWS</td>
<td>1</td>
</tr>
<tr>
<td>TABLESPACES</td>
<td>100</td>
</tr>
<tr>
<td>PLANS</td>
<td>100</td>
</tr>
<tr>
<td>PLAN STATEMENTS</td>
<td>30</td>
</tr>
<tr>
<td>PACKAGES</td>
<td>200</td>
</tr>
<tr>
<td>PACKAGE STATEMENTS</td>
<td>30</td>
</tr>
<tr>
<td>PACKAGE LISTS</td>
<td>2</td>
</tr>
<tr>
<td>EXECUTED STATEMENTS</td>
<td>30</td>
</tr>
<tr>
<td>TABLES IN STATEMENTS</td>
<td>2</td>
</tr>
<tr>
<td>TEMP 4KB SPACES</td>
<td>100M (at least 10 times the size of the largest fact table)</td>
</tr>
<tr>
<td>TEMP 4 KB DATA SETS</td>
<td>4 (at least the maximum number of parallel tasks.)</td>
</tr>
<tr>
<td>TEMP 32 KB SPACES</td>
<td>40M</td>
</tr>
<tr>
<td>TEMP 32 KB DATA SETS</td>
<td>2</td>
</tr>
</tbody>
</table>

The storage-size-related values that are given in these tables should be considered as initial settings for systems with a significant workload, such as typical production systems. The settings are subject to regular monitoring and tuning. For more information, see Performance Tuning Considerations [Page 96].
Low-Workload Environments: 11.6 Recommended Settings

For systems for which a lower workload is expected, for example test and sandbox systems, you have the following options:

- You can combine multiple smaller SAP systems into a single DB2 subsystem and size it appropriately, for example by using the values as for one large SAP system.
- You can use significantly lower values for these parameters. For example, the values presented in the table below are appropriate for a system with ten to twenty light concurrent users, with up to ten work processes, and without significant batch activity; in such a system, these values reduce virtual and real storage consumption and still deliver acceptable performance.

Possible Settings for Low-Workload Environments

<table>
<thead>
<tr>
<th>Field</th>
<th>Possible Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTBUFF</td>
<td>400</td>
</tr>
<tr>
<td>EDMPOOL</td>
<td>5000</td>
</tr>
<tr>
<td>EDMSTMTC</td>
<td>30000</td>
</tr>
<tr>
<td>MAXKEEPD</td>
<td>6000</td>
</tr>
<tr>
<td>DSMAX</td>
<td>2000</td>
</tr>
<tr>
<td>SRTPOOL</td>
<td>10000</td>
</tr>
<tr>
<td>MAXRBLK</td>
<td>100000</td>
</tr>
<tr>
<td>VPSIZE BP0</td>
<td>200</td>
</tr>
<tr>
<td>VPSIZE BP1</td>
<td>200</td>
</tr>
<tr>
<td>VPSIZE BP2</td>
<td>2000</td>
</tr>
<tr>
<td>VPSIZE BP3</td>
<td>3000</td>
</tr>
<tr>
<td>VPSIZE BP40</td>
<td>2000</td>
</tr>
<tr>
<td>VPSIZE BP8K0</td>
<td>200</td>
</tr>
<tr>
<td>VPSIZE BP16K0</td>
<td>200</td>
</tr>
<tr>
<td>VPSIZE BP32K</td>
<td>200</td>
</tr>
</tbody>
</table>

The parameters listed above are only a small subset of all available DB2 system parameters. Their values have special significance for the SAP system environment. For a complete list, see the IBM documentation *DB2 UDB for z/OS Installation Guide*. We strongly recommend that you use the default values for any DB2 system parameter not explicitly mentioned in this section.

The following parameters are included here as customers often ask for their recommended values:

CDSSRDEF = 1 and RETVLCFK = NO and CMTSTAT = INACTIVE
12 Configuring the Default Sign-On Exit Routine

Since SAP systems use DB2 secondary authorization routines, the default sign-on exit routine must be replaced with the IBM-supplied sample routine. The installation job that can be used to perform this task is described in Installation Step 6: Define User Authorization Exit Routines: DSNTIJEX in the IBM documentation DB2 for z/OS Installation Guide.
13 Setting Up the Database Attach Name

This section is only relevant for the z/OS application server. For use with DB2 Connect, the SAP application server always attaches to a DDF location. For more information, see DB2 Connect Installation and Customization [Page 85].

The name that the SAP application server uses to attach to the DB2 subsystem is called the Database Attach Name. For non-data sharing, it is always the DB2 subsystem ID (SSID). In data sharing it is either the SSID of the DB2 members of the data sharing group or the group attach name. The SAP application server retrieves the Database Attach Name from the following places:

It is the value of the SAP environment variable dbs_db2_ssid, or the value of SSID of the main connection in the SAP connection profile file connect.ini which is described in the SAP installation guides for DB2 UDB on z/OS.

- During installation, migration and system copy of an SAP system, you will be prompted for the Database Attach Name. The SAP system will then create the appropriate environment variables and profile settings.

- For information on how to define the DB2 group attachment name, see “Installing the DB2 Subsystem”, installation step “Define DB2 to MVS” in the IBM documentation DB2 for z/OS Installation Guide.
14 Reducing the Number of Data Sets

The SAP system installation and upgrade processes create tablespaces and indexes with the option DEFINE=NO. This means that the underlying data sets are not created until the first row is inserted into the corresponding table. For most SAP systems, a large number of tables remain empty. This means that a very significant number of data sets will not be created. This is beneficial for many database administration tasks as well as for the DASD space utilization. The objects created with DEFINE=NO are fully supported by all of the functions within the SAP system. Potential problems may arise if you use third-party tools that do not provide support for the new objects. Please check with your tool provider to determine if the tools offer the necessary support.
15 Stored Procedures Enablement

Some CCMS functions, BW functions, the SAP upgrade tool and the SAP installation tool require stored procedures. The DBA Planning Calendar as part of CCMS, for example, utilizes the WLM-established stored procedure DSNUTILS. The following sections describe how to install and test stored procedures. For more information on defining classification rules with WLM, see SAP Planning Guide: z/OS Configuration for SAP on IBM DB2 on z/OS.

Stored procedure DSNUTILS with SAP BW and SAP APO:

The stored procedure DSNUTILS is a prerequisite for running SAP systems with BW functionality, because they run the DB2 Utilities RUNSTATS and REBUILD INDEX automatically.
15.1 Installing Stored Procedures Using WLM

To install stored procedures using WLM, proceed as follows:

1. Define a new application environment.
   a. Extract the definition from the Workload Manager couple data set.
   b. In the WLM Definition menu, select option Application Environment and create a new application environment. The following screen provides an example:

```
Create an Application Environment

Command ===>__________________________________________________________
Application Environment . . . D660UTIL____________________________   Required
Description    .  .  .  .  .  .  .  .  .  WLM ENV Stored Proc dsuutils
Subsystem Type    .  .  .  .  .  .  DB2
Procedure Name    .  .  .  .  .  .  D660UTIL
Start Parameters    .  .  .  .  .  .  DB2SSN=D660,NUMTCB=1,APPLENV=D660UTIL

Limit on starting server address spaces for a subsystem instance:
1  1.  No limit
2  2.  Single address space per system
3  3.  Single address space per sysplex
```

To ensure that these stored procedures DSNUTILS functions properly when it is executed in parallel, NUMTCB must be set to 1. This is described in more detail below.

In the above example, the parameter DB2SSN is populated with the value D660. In one of the next steps, a JCL must be created to run the WLM-established stored procedures address space for the DB2 subsystem D660. The JCL procedure will be started exclusively for the DB2 subsystem.

The variable &IWMSSNM for the DB2SSN parameter makes it possible to use one JCL for multiple DB2 subsystems. The variable &IWMSSNM represents the name of the DB2 subsystem for which the address space starts. The following screen provides an example:
Create an Application Environment

Command ===>

<table>
<thead>
<tr>
<th>Application Environment</th>
<th>D660UTIL</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>WLM ENV Stored Proc dsutils</td>
<td></td>
</tr>
<tr>
<td>Subsystem Type</td>
<td>DB2</td>
<td></td>
</tr>
<tr>
<td>Procedure Name</td>
<td>D660UTIL</td>
<td></td>
</tr>
<tr>
<td>Start Parameters</td>
<td>DB2SSN=&amp;IWMSSNM,NUMTCB=1,APPLENV=D660UTIL</td>
<td></td>
</tr>
</tbody>
</table>

Limit on starting server address spaces for a subsystem instance:
1. No limit
2. Single address space per system
3. Single address space per sysplex

2. You should now install the new definition on the WLM couple data set and activate the service policy. Display the status of the new application environment as follows (example):

/D WLM,APPLENV=D660UTIL

The following response appears (example):

RESPONSE=SAPC
IWMO291 16.48.57 WLM DISPLAY 543
APPLICATION ENVIRONMENT NAME STATE STATE DATA D660UTIL AVAILABLE
ATTRIBUTES: PROC=D660UTIL SUBSYSTEM TYPE: DB2

3. Create a JCL for running the WLM-established stored procedures address space (example):
4. Create stored procedures in the DB2 catalog of the subsystem and permit access to the new object. Adapt a copy of the member DSNTIJSG of the data set SDSNSAMP and create the procedure SYSPROC.DSNUTILS. The value of the parameter WLM ENVIRONMENT must match the entry in the WLM application environment (for example, D660UTIL). WLM ENVIRONMENT identifies the name of the WLM environment in which the stored procedure is to run when the DB2 stored procedure is WLM-established.
15.2 Testing Stored Procedures Using WLM

To test stored procedures using WLM, compile, link, and run a stored procedure requester program, as follows:

Adapt a copy of the member DSNTEJ6U of the data set SDSNSAMP to prepare and execute the sample program DSN8EPU, (which also resides in the data set SDSNSAMP).

The run of the sample program starts the additional DB2 address space, if not yet started, for running WLM-established stored procedures address space, for example, D660UTIL. In any case, the sample program executes a DIAGNOSE DISPLAY MEPL utility using a stored procedure to test whether it is available.
15.3 Installing Real Time Statistics (RTS) Using WLM

Real Time Statistics (RTS) are collected for easy and inexpensive detection of database objects needing some DBA intervention. The collected data includes a large number of statistics values such as the number of rows inserted, deleted, changed since the last `RUNSTATS`, `REORG`, or `COPY`. DB2 always generates in-memory statistics for each tablespace and indexspace. When the statistics are to be externalized, DB2 examines the in-memory data, calculates the new totals, updates the new real-time statistic tables with the new totals, and resets the in-memory data. This process is an asynchronous task. The real-time statistics are stored in tables `SYSIBM.TABLESPACESTATS` and `SYSIBM.INDEXSPACESTATS` in tablespace `DSNRTSDB.DSNRTSTS`.

The data collected in the RTS is examined by the stored procedure `DSNACCOR`, which recommends running a utility based on complex calculations performed with this data. While still keeping track of changes itself in SAP table MONI, beginning with R/3 Database Release 4.6C the SAP system utilizes `DSNACCOR` instead of using this data.

The SAP system data collection programs `RSDA_Collect_Daily` and `RSDBA_Collect_Hourly`, which run at system startup and daily or hourly, rely on the existence of RTS and DSNACCOR. Since the alerts that are generated by these programs are used as input for the CCMS Monitor Set (transaction RZ20), the extent Monitor (transaction DB02), the DBA Planning Calendar (transaction DB13), and the Backup Monitor (transaction DB12), these transactions will not function properly if RTS and DSNACCOR are not installed correctly.

The following instructions will only give you an overview of the tasks you have to complete to install RTS and DSNACCOR. For detailed information, refer to `DB2 UDB for z/OS V8 Utility Guide and Reference`.

In order to install Real Time Statistics and DSNACCOR, you must complete the following tasks:

1. Create the statistics database, tablespace, tables, and indexes. You can do this by adapting a copy of the member `DSNTESS` of the SDNSAMP data set to define the DB2 RTS objects. Calculate the space allocation requirements as requested, change `PRIQTY` and `SECQTY`, and run the adapted job in a SPUFI environment of the DB2 subsystem or use it as input for DSNTIAD.

2. Start the DSNRTSDB database in which the objects are created in read-write mode. Be aware that the database must be started explicitly although a display shows it as already started.

3. Create the procedure `DSNACCOR` and grant the `EXECUTE` privilege for the procedure to public. You can do this by adapting a copy of SDNSAMP’s data set member `DSNTIJSG` and running it on your DB2 subsystem. Ensure that the value of the parameter `WLM ENVIRONMENT` matches an entry in the WLM application environment as described in `Installing Stored Procedures using WLM` [Page 65].

4. Grant the RACF group `<SCHEMA>` SELECT privileges on the RTS tables `SYSIBM.TABLESPACESTATS` and `SYSIBM.INDEXSPACESTATS`. For example, if your SAP system is installed with the schema SAPR3, the following SQL statements must be issued in SPUFI or similar tool:
   ```sql
   GRANT SELECT, DELETE ON SYSIBM.TABLESPACESTATS TO SAPR3;
   GRANT SELECT, DELETE ON SYSIBM.INDEXSPACESTATS TO SAPR3;
   ```

5. Check your current `DSNRTIJUZ` job and set the interval for externalizing in-memory statistics to a value that meets your requirements by updating the parameter `STATSINT`. To activate the new value, `DSNZPARM` must be link-edited and the DB2 subsystem must be restarted.
DSNACCOR needs a TEMP database and tablespace in the TEMP database. If they do not exist, the SAP system will create them. They will not be dropped but reused each time DSNACCOR is invoked.

6. The SAP system will use the DSNACCOR exception table (see the SAP documentation Database Administration Guide: SAP Web Application Server). SAP’s recommendations for the installation of that exception table differ from the official IBM documentation:

7. Execute the following SQL commands, for example, in SPUFI:

   CREATE TABLE DSNACC.EXCEPT_TBL
   (DBNAME CHAR (8) NOT NULL,
   NAME CHAR (8) NOT NULL,
   QUERYTYPE CHAR(40),
   ASSOCDB CHAR (8)
   ASSOCTS CHAR (8)
   PRIMARY KEY (DBNAME,NAME))
   CCSID EBCDIC
   COMMIT;
   GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE DSNACC.EXCEPT_TBL
   TO PUBLIC; CREATE UNIQUE INDEX DSNACC.EXCEPT_TBL0 ON
   DSNACC.EXCEPT_TBL (DBNAME, NAME);
   COMMIT;

   If RTS or DSNACCOR is not installed correctly, messages will appear in the system log (transaction SM21) indicating that the data collection programs could not be executed.
15.4 Installing Control Center Procedures Using WLM

The FTP JES Interface has been substituted by utility and job submissions via Stored Procedures.

Use

Control Center for DB2 Universal Database for z/OS and OS/390 offers a set of Stored Procedures that can submit utilities and JCL jobs. You have to install at least the Stored Procedures listed below.

Stored Procedures for Database Administration

Parallel Utility Execution Stored Procedure (DSNACCMO)

The stored procedure DSNACCMO allows you to invoke utilities via the widely known stored procedure DSNUTILS on multiple objects in parallel. It is able to determine the optimal parallel degree of utility executions that performs the given task in the shortest overall utility execution time. The maximum parallel degree can be restricted by means of a stored procedure parameter.

A common task of an SAP installation is to process a large number of objects. This is one reason why DSNACCMO fits SAP very well: It facilitates the installation tremendously and automatically optimizes its execution.

DSNACCMO can be exploited by the DBA Planning Calendar (Transaction code DB13).

DB2 command submission Stored Procedure (DSNACCMD)

The stored procedure DSNACCMD executes DB2 commands on the DB2 subsystem to which it is connected. The DB Performance Monitor (Transaction code ST04) uses this stored procedure because it lessens the dependency on the external program rfcoscol. rfcoscol was previously used as the sole means to issue DB2 commands by means of IFI.

If either DSNACCMD or rfcoscol is operational, the SAP system can submit DB2 commands.

DB2 Subsystem ID Stored Procedure (DSNACCSS)

The DB Performance Monitor (Transaction code ST04) exploits the stored procedure DSNACCSS to query the SSID of the DB2 subsystem to which it is currently connected.

Stored Procedures for Data Set Manipulation

Stored Procedure to Create or Write to a Data Set (DSNACCDS)

The stored procedure DSNACCDS creates a data set member or a data set of type PS (physical sequential data set), PDS (partitioned data set) and PDSE (partitioned data set extended) and writes data to it. It either appends or replaces an existing data set or data set member. Moreover, it creates a new GDS (generation data set) for an existing GDG (generation data group). The amount of space that it allocates is minimal.

DSNACCDS was specifically designed for one purpose: Uploading JCL Jobs from the DBA Planning Calendar (Transaction code DB13). Due to the JCL limitation that each line can contain only 80 characters, DSNACCDS creates or writes to data sets with LRECL set to 80.

Stored Procedure to rename a data set (DSNACCDR)

The stored procedure DSNACCDR renames a data set of type PS, PDS or PDSE, or a data set member. It is currently not used by SAP, but that could change in the near future.
**Stored Procedure to delete a data set (DSNACCDD)**

The stored procedure DSNACCDD deletes a data set member, or a data set of type PS, PDS or PDSE, or a GDS. It is currently not used by SAP, but that could change in the near future.

**Stored Procedure to check if data set exists (DSNACCDE)**

The stored procedure DSNACCDE checks if a non-VSAM data set or a data set member exists. It accomplishes this by querying the ICF (Integrated Catalog Facility) catalog of z/OS.

It is used by SAP to check if members exist during Upload of JCL jobs from the DBA Planning Calendar (Transaction code DB13).

**Stored Procedure to query data set properties (DSNACCDL)**

The stored procedure DSNACCDL lists data sets, data set members, VSAM clusters, generation data sets or GDGs. It provides attributes on these objects, such as the primary extent size.

It is currently not used by SAP, but in the near future this stored procedure will allow SAP to stop using IDCAMS LISTCAT jobs to find out data set sizes and properties for the SAP tables and indexes space monitor (Transaction code DB02). SAP systems today rely on submitting IDCAMS LISTCAT jobs through FTP. They parse the output of these jobs to determine the relevant information, which is a cumbersome approach that is also complex to set up.

DSNACCDL yields the following data set attributes:

- Creation date
- Type of data set
- Volume name
- Primary extent size
- Secondary extent size
- Unit (track, block, cylinder)
- Extents in use
- Actual disk usage in bytes

For VSAM data sets, it additionally provides the frequently-allocated RBA and the frequently-used RBA.

**Stored Procedures for JCL and UNIX commands**

**Stored Procedure to submit a JCL job (DSNACCJS)**

The stored procedure DSNACCJS submits a JCL job for batch execution. It returns the ID that JES assigns to the job. The text of the job is passed to DSNACCJS through a global temporary table.

**Stored Procedure to retrieve JCL job status (DSNACCJQ)**

The stored procedure DSNACCJQ allows you to inquire the status of a JCL job. Depending on the requested mode, it returns related console messages or a status value. The returned status values indicate if a job is currently in the input queue, is active or is in the output queue.

**Stored Procedure to fetch output of a JCL job (DSNACCJF)**

The stored procedure DSNACCJF fetches the spool output files of a specified JCL job. It inserts the lines of the output listings into a result set table.
**Stored Procedure to cancel a JCL job (DSNACCJP)**

The stored procedure DSNACCJP purges or cancels a specified job. It returns related console messages.

DSNACCJS, DSNACCJQ, DSNACCJF, and DSNACCJP are used for asynchronous z/OS JCL Job submission from JES Interface (Transaction code DB2J) and for synchronous JCL Job submission in the context of rfcoscol administration (BIND, GRANT, and START rfcoscol in Transaction code DB2 -> Checks/Settings -> SAP Collector Settings).

**Stored Procedure to issue a UNIX command (DSNACCUC)**

Finally, the stored procedure DSNACCUC issues UNIX commands in UNIX System Services. The commands are executed in the UNIX environment of the user under which DSNACCUC runs. It returns the output in a result set table. It's currently not used by SAP, but that is going to change in the near future.

**Installing the Stored Procedures**

The Stored Procedures mentioned above can be installed as described step by step in the following checklist. This description does not intend to replace official documentation for instance in the PTFs and APARs. A very good source of information in this context is chapter "25. DB2-supplied stored procedures" in IBM redbook SG24-7083-00 "DB2 for z/OS Stored Procedures: Through the CALL and Beyond" that is available for free as a download at www.redbooks.ibm.com.

8. The following WLM application environments are needed:

<table>
<thead>
<tr>
<th>Stored Procedures</th>
<th>NUMTCB</th>
<th>Suggested Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUTILS, DSNUTILU</td>
<td>1</td>
<td>[SSID]UTIL</td>
</tr>
<tr>
<td>DSNACCMO</td>
<td>100</td>
<td>[SSID]CCMO</td>
</tr>
<tr>
<td>DSNACCJS, ..JQ, ..JF, ..JP, ..UC</td>
<td>N</td>
<td>[SSID]CCJ</td>
</tr>
<tr>
<td>Others</td>
<td>N</td>
<td>[SSID]CCA</td>
</tr>
</tbody>
</table>

**Table Key**

N

N for NUMTCB in the table above means any natural number from 1 to 100. To minimize the overhead required to start new and manage WLM application environment address spaces, we recommend that you create a WLM application environment with NUMTCB=25 to 40. However, in a very CPU-constrained environment, the recommendation is to use a lower number such as 6 through 10.

**DSNUTILS and DSNUTILU**

DSNUTILS and DSNUTILU use data sets that are allocated in the JCL procedure for the WLM application environment. If more than one instance of DSNUTILS were allowed to run in the same address space, the instances would overwrite each others data sets, leading to unpredictable behavior. The same requirement exists for LANGUAGE REXX stored procedures, which must also run in a NUMTCB=1 environment.

**DSNACCMO**

DSNACCMO is a complex stored procedure that creates up to 99 parallel threads to execute DB2 online utilities. For every parallel thread, a separate Task Control Block is required. Hence, a WLM application environment for DSNACCMO with NUMTCB=100
has to be created. No other stored procedure should be allowed to run in that application environment.

**JCL and USS Commands**

The stored procedures for submitting JCL and USS commands need their own WLM application environment because they must be program-controlled. If a stored procedure that is not program-controlled is executed in this WLM application environment, a subsequent run of DSNACCJ*/DSNACCUC fails and results in the following messages:

ICH420I PROGRAM ... FROM LIBRARY ... CAUSED THE ENVIRONMENT TO BECOME UNCONTROLLED.

BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR DAEMON (BFX.DAEMON) PROCESSING.

No special considerations exist for the remaining stored procedures.

For the description of WLM setup and the WLM application environment definition please refer to “Performance Tuning Considerations”, section “WLM/SRM Considerations” in the SAP Planning Guide “DB2 Setup”, and Installing Stored Procedures Using WLM [Page 65].

9. The stored procedures for submitting JCL and USS commands must be program-controlled, because they use the `__login()` function for switching user identity. If you have defined BPX.DAEMON profile in the facility class in your environment (which is recommended by the way), you either need to define programs from traditional libraries to program control or to define BPX.DAEMON.HFSCTL profile in the facility class. The latter is not recommended, because then programs that are loaded from MVS libraries are not checked for program control. In other words, any MVS data set is considered to be secure and can execute nearly anything. In our opinion, this poses a severe security risk. Therefore we describe the first option here, which is more work but far more safe:

To define programs from traditional libraries to program control, you need to:

a. Activate the RACF program control (both access control to load modules and program access to data sets).

   SETROPTS WHEN (PROGRAM)

b. The following members of SDSNLOAD require to be program controlled:

   DSNX9WLM
   DSNX9SPA
   DSNARRS
   DSN3ID00
   DSNX9WLS
   DSNACCJS
   DSNACCJP
   DSNACCJQ
   DSNACCJF
   DSNACCUC
   DSNACCSS
   DSNACCIS
DSNUTILS

This can be done by defining discrete RACF PROGRAM class profiles as follows:

RDEFINE PROGRAM member name ADDMEM ('datasetname'//NOPADCHK)
UACC (READ)

for instance:

RDEFINE PROGRAM DSNX9WLM ADDMEM
('SGF1.SDSNLOAD'//NOPADCHK)
UACC (READ)

c. Refresh the in-storage copy of the PROGRAM profile:

SETROPTS WHEN(PROGRAM) REFRESH

For more information on BPX.DAEMON and setting up program control, you can refer to z/OS UNIX System Services Planning, GA22-7800-03.

10. The stored procedures DSNACCJP and DSNACCJQ use the Extended MCSconsole to issue JES commands to the console. These two stored procedures use the TSO/E user ID (which must be the same like your SAP user, and you have to maintain your password in transaction DB2Jas previously) in the same way as an authorized TSO/E user can do during a console session.

This means, that the security administrator has to define a RACF user profile to control the console attributes of the EMCS console for the administrators TSO/E resp. SAP user ID. For example:

ADDUSER USER001 OPERPARM(AUTH(SYS))

This example defines the user ID USER001 as an EMCS console with console attributes defined by the OPERPARM keyword. Note that the example includes only the information about console attributes for USER001. For complete information on the RACF ADDUSER command, refer to z/OS Security Server RACF Command Language Reference, SA22-7687-03.

Ensure that the user of DSNACCJP and JQ has READ access to a profile in the RACF OPERCMDS class named MVS.MCSOPER.console-name:

- Issue the SETROPTS command to activate the OPERCMDS class:
  
  SETROPTS CLASSACT(OPERCMDS)

- Issue the SETROPTS command to activate generic profiles for the class:
  
  SETROPTS GENERIC(OPERCMDS)

- Issue RDEFINE to establish a profile for MVS.MCSOPER.*:
  
  RDEFINE OPERCMDS MVS.MCSOPER.* UACC(NONE)

- Give the TSO/E user ID access to the class:
  
  PERMIT MVS.MCSOPER.* CLASS(OPERCMDS) ID(USER001) ACCESS(READ)

- Issue the SETROPTS RACLST command to refresh the OPERCMDS reserve class:
  
  SETROPTS RACLST(OPERCMDS) REFRESH

The stored procedures DSNACCJP and DSNACCJQ issue the JES commands to cancel, purge or display a job. You will probably want to protect these JES commands:

- Define the resource profile:
  
  RDEFINE OPERCMDS jesname.CANCEL.* UACC(NONE)

  RDEFINE OPERCMDS jesname.STOP.* UACC(NONE)
RDEFINE OPERCMDS jesname.DISPLAY.* UACC(NONE)

g. Give UPDATE access to users:

PERMIT jesname.CANCEL.* ID(USER001) ACCESS(UPDATE)

PERMIT jesname.STOP.* ID(USER001) ACCESS(UPDATE)

PERMIT jesname.DISPLAY.* ID(USER001) ACCESS(READ)

SETRROPTS RACLIST(OPERCMDS) REFRESH

Last but not least, to make sure that the related messages are always received by the EMCS console, provide the command:

ALTUSER USER001 OPERPARM(ROUTCODE(ALL) AUTH(INFO))

for the user employing DSNACCJQ and JP.

For more information on RACF commands, refer to z/OS Security ServerRACF Command Language Reference, SA22-7687-03.

For more information on the EMCS console, refer to z/OS MVS Planning: Operations, SA22-7601-03.

11. Create the stored procedures by:

a. adapting a copy of members DSNTIJSG and DSNTIJCC of the SDSNSAMP data set according to the instructions in the headers. Ensure that the value of parameter WLM_ENVIRONMENT for each stored procedure matches the correct WLM application environment.

b. You have to grant SELECT and DELETE privileges on two RTS tables named SYSIMB, TABLESPACESTATS and SYSIMB.INDEXSPACESTATS to user <SCHEMA>. For example, if your SAP system is installed with schema SAPR3, the following SQL statements have to be added in member DSNTIJCC:

GRANT SELECT, DELETE ON SYSIMB.TABLESPACESTATS TO SAPR3;

GRANT SELECT, DELETE ON SYSIMB.INDEXSPACESTATS TO SAPR3;

c. You have to GRANT ALL privileges to PUBLIC on the global TEMP tables that are used by the Stored Procedures. This is not a security issue, because global TEMP tables are instantiated in the session, so their content is only visible from within the same session. Please add the following SQL statements to member DSNTIJCC:

GRANT ALL ON TABLE DSNACC.BP TBL to PUBLIC;

GRANT ALL ON TABLE DSNACC.CMDMSG TBL to PUBLIC;

GRANT ALL ON TABLE DSNACC.DBSTATUS TBL to PUBLIC;

GRANT ALL ON TABLE DSNACC.DSLIST to PUBLIC;

GRANT ALL ON TABLE DSNACC.DSNRECORDS to PUBLIC;

GRANT ALL ON TABLE DSNACC.IXTEMP TBL to PUBLIC;

GRANT ALL ON TABLE DSNACC.JFRECORDS to PUBLIC;

GRANT ALL ON TABLE DSNACC.JSRECORDS to PUBLIC;

GRANT ALL ON TABLE DSNACC.MO_SYSPRINT to PUBLIC;

GRANT ALL ON TABLE DSNACC.MO_TBL to PUBLIC;

GRANT ALL ON TABLE DSNACC.MO_TBL2 to PUBLIC;

GRANT ALL ON TABLE DSNACC.ST_TBL_IN to PUBLIC;

GRANT ALL ON TABLE DSNACC.ST_TBL_OUT to PUBLIC;

GRANT ALL ON TABLE DSNACC.THREAD_TBL to PUBLIC;
GRANT ALL ON TABLE DSNACC.TSTEMP_TBL to PUBLIC;
GRANT ALL ON TABLE DSNACC.UCRECORDS to PUBLIC;
GRANT ALL ON TABLE DSNACC.UTILITY_TBL to PUBLIC;

d. running DSNTIJSG and DSNTIJCC on your DB2 subsystem

Many of the stored procedures require a TEMP database including tablespace. For ease of use, SAP is going to create them if there are none existing yet. They will not be dropped and are reused.

DSNACCMO is used by SAP Installation, so you need to install at least DSNACCMO.

If the installation of the stored procedures did not work, you will be notified by corresponding messages in system log (Transaction SM21) due to unsuccessful stored procedure executions within transactions DB13, ST04, DB2 and DB2J.

12. Stored procedures do not recognize secondary authorization IDs by default. SAP always uses a secondary authorization ID because it issues SET CURRENT SQLID=<SCHEMA>. You need to activate a certain DB2 exit that is delivered with the example library in order to use a secondary authorization ID with Stored Procedures:

a. Adapt a copy of member DSN3SATH of the SDSNSAMP data set by uncommenting the code marked with PQ51163 and PQ57756.

b. Run member DSNTIJEX of the SDSNSAMP dataset on your DB2 subsystem.

Recommendations

13. SAP produces for all of the stored procedures small trace files in SAP directory DIR_HOME (see transaction AL11), that can come in very handy in the case of an error. They are reset at each invocation of a stored procedure, so they show only the last execution of the stored procedure. If you encountered an error, that you want to analyze at a later time, take care to save this file in a different place to prevent it from getting overwritten.

14. You can check if stored procedures work properly by using:

/ D WLM,APPLENV=* 

This will show you in z/OS system log output like this:

D WLM,APPLENV=* 
IWMO29I  17.20.03  WLM DISPLAY 528

APPLICATION ENVIRONMENT NAME     STATE     STATE DATA
D8C0CCJ                         AVAILABLE
D8C0CCMO                        STOPPED
D8C0UTIL                        QUIESCED
D8D0CCJ                         AVAILABLE
D8D0CCMO                        AVAILABLE
D8D0UTIL                        AVAILABLE

If any of the WLM environments of a system, that you are working with, has another state than AVAILABLE, restart the WLM environment in question with a command like the following:

/V WLM,APPLENV=D8C0CCMO,RESUME
/V WLM,APPLENV=D8C0UTIL,RESUME
The stored procedure itself can also be stopped due to an error. You can check this with DB2 command:

/-SGF1 DISPLAY PROC(*)

This results in the following output:

- SGF1 DIS PROC(*)

DSNX940I  -SGF1 DSNX9DIS DISPLAY PROCEDURE REPORT FOLLOWS - 664

-------- SCHEMA=SYSPROC

<table>
<thead>
<tr>
<th>PROCEDURE</th>
<th>STATUS</th>
<th>ACTIVE</th>
<th>QUEUED</th>
<th>MAXQUE</th>
<th>TIMEOUT</th>
<th>WLM_ENV</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNACCOR</td>
<td>STOPPED</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>SGFUTIL</td>
</tr>
<tr>
<td>DSNUTILS</td>
<td>STARTED</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>SGFUTIL</td>
</tr>
</tbody>
</table>

DSNX9DIS DISPLAY PROCEDURE REPORT COMPLETE

DSN9022I  -SGF1 DSNX9COM '-DISPLAY PROC' NORMAL COMPLETION

In this example, stored procedure DSNACCOR was stopped due to a previous error. Restart the stored procedure to continue:

/-SGF1 START PROC(DSNACCOR)

15. If you get the following RACF messages in z/OS system log:

ICH420I PROGRAM ... FROM LIBRARY ... CAUSED THE ENVIRONMENT TO BECOME UNCONTROLLED.

BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR DAEMON (BPX.DAEMON) PROCESSING.

then you either did not define a separate WLM environment for the stored procedures DSNACCJS, JQ, JF, JP, and DSNACCUC, or you missed one of the members of SDSNLOAD, that need to be program controlled.

To solve the first problem, define a separate WLM environment as described above, and redefine the stored procedures with the correct specification of the new WLM environment as WLM_ENVIRONMENT.

To solve the second problem, let your system programmer execute the following RACF commands:

RDEFINE PROGRAM member  ADDMEM('library'//NOPADCHK) UACC(READ)

SETROPTS WHEN(PROGRAM) REFRESH

"member" and "library" can be taken directly out of the RACF message.

1. You find the following message in SAP system log (transaction SM21):

Error executing stored procedure DSNACCJ? - MES_TXT: EMCS activation failed. Macro MCSOPER: RC=04,RSN=00

where ? can be Q or P.

As already detailed above, DSNACCJQ and JP use EMCS console. SDSF uses the same console, and if the user submitting JCL job is at the same time in SDSF, DSNACCJQ and JP can't execute. Please leave SDSF while submitting JCL jobs from SAP.

Another possible source for this error can be, if your system programmer did not execute the following RACF command for the user ID submitting JCL jobs from SAP:

ALTUSER userID OPERPARM(ROUTCODE(ALL) AUTH(INFO))
15.5 Install the DB2 Metadata Stored Procedures

Before you can use certain functions in the DB2 Universal JDBC Driver, you need to install the following DB2 stored procedures:

- SQLCOL_PRIVILEGES
- SQL_COLUMN
- SQL_FOREIGNKEYS
- SQL_GET_TYPEINFO
- SQL_PRIMARYKEYS
- SQL_PROCEDURE_COLS
- SQL_PROCEDURES
- SQL_SPECIAL_COLUMNS
- SQL_STATISTICS
- SQL_TABLE_PRIVILEGES
- SQL_TABLES
- SQL_UDTS
- SQL_CAMEO_MESSAGE

These procedures are provided in DB2 for z/OS Version 8. They are created by installation job DSNTIJSG when created as part of a new installation or migration, or by job DSNTIJMS, for installations that were installed or migrated before the procedures were introduced into Version 8. These jobs must be customized before execution, as described in the job prologs.

Prior to running these jobs, you should set the subsystem parameter named DESCSTAT to YES. DESCSTAT corresponds to installation field DESCRIBE FOR STATIC on panel DSNTIPF. See Part 2 of the IBM V8 DB2 Installation Guide for more information.
16 Checking the Installation of the DB2 Subsystem

To finally check the DB2 subsystem installation, perform the following steps:

1. Simple check:
   
   Verify that DB2 is installed and a subsystem is accessible by executing from TSO:
   
   ```
   dsn system(<DB_ATTACH_NAME>)
   ```
   
   where `DB_ATTACH_NAME` is the Database Attach Name, which is either the DB2 subsystem name or the DB2 group attachment name, depending on which name you want to use.
   
   If the subsystem is accessible, this command brings up the DB2 command processor; otherwise an error message is displayed.
   
   To leave the DB2 command processor, use the `END` command.

2. Refer to the installation verification procedure of DB2 described in "Verifying with the Sample Applications" in *DB2 for z/OS Installation Guide*. 
17 Ensuring Optimal DB2 Settings

Using the Correct DB2 System Parameters

Make sure that all of the DB2 system parameters are set as listed in this documentation under DB2 Setup.

Switching off Unnecessary DB2 Traces

The only trace that should be active during the SAP system installation are the DB2 default statistics trace classes. All other traces, particularly global and performance traces, should be switched off. At SAP system installation time, the Accounting and Monitor trace classes 2 and 3 can have a significant overhead, so make sure you switch them off too. However, during regular operations, Accounting trace classes 2 and 3 should be switched on. You can check which traces are active by checking the output of the DISPLAY TRACE(*) command.

Optimizing DB2 Log Processing

A good performance of the DB2 logging is very important at the time of heavy DDL (INSERT, UPDATE and DELETE) activity. At SAP system installation time, a large number of DB2 tables are created and a large number of rows are inserted into them.

Make sure you create multiple (at least three) and large DB2 active log data sets, ideally one per 3390 volume. In any case, place your active logs on the DASD devices with the best write characteristics available in your installation. Do not share volumes on which they reside with other highly active data sets. Turn on the DASD Fast Write feature, and add as much Non-Volatile Storage (NVS) as you can afford.

For more information on DB2 Log Processing, see the SAP documentation SAP on IBM DB2 UDB for z/OS: Database Administration Guide.
18 DB2 DDF Setup

The setup of DB2 DDF (Distributed Data Facility) consists of the following steps:

1. Create a distributed data facility address space startup procedure (xxxxDIST) in the appropriate PROCLIB, for example, SYS1.PROCLIB, where xxxx is the name of the DB2 subsystem. The name of the procedure must not be changed.

   See the following example:

   ```
   //*************************************************
   //*    JCL PROCEDURE FOR THE STARTUP OF THE
   //*    DISTRIBUTED DATA FACILITY ADDRESS SPACE
   //*    
   //*************************************************
   //D620DIST PROC RGN=17M,
   //             LIB='D640.V810.SDSNEXIT'
   //IEFPROC EXEC PGM=DSNYASCP,REGION=&RGN
   //STEPLIB  DD  DISP=SHR,DSN=&LIB
   //         DD  DISP=SHR,DSN=SYS1.SCEERUN
   //         DD  DISP=SHR,DSN=D640.V810.SDSNLOAD
   ```

2. Define the DDF startup mode in the DB2 initialization parameters (DSNTIJUZ).

   You can define DDF=NO, AUTO or COMMAND. We recommend to define DDF=AUTO, so this facility is automatically initialized and started when the DB2 subsystem is started. The DDF address space is started as part of DDF initialization.

3. Define the DDF parameters in the BSDS of the DB2 subsystem.

4. Specify the location of the DB2 subsystem. The location is a unique name which requesters use to connect to this DB2 subsystem. The name must begin with a letter and must not contain special characters. Acceptable characters are A-Z, 0-9, and underscore.

5. Specify the logical unit name (LU name) for this DB2 subsystem.

   This name uniquely identifies this DB2 subsystem to VTAM. It is also used to uniquely identify logical units of work within DB2 trace records. The name must begin with a letter and must not contain special characters.

6. Specify the TCP/IP port number used for accepting TCP/IP connection requests from remote DRDA clients.

7. Specify the TCP/IP port number used to process requests for 2-phase commit resynchronization. This value must be different to the value specified for DRDA PORT.

   See the following example:

   ```
   //*MODLOG DDF DEFINITIONS / DST
   /*JOBPARM SYSAFF=SAPE
   //MAKELU EXEC PGM=DSNJU003
   //STEPLIB  DD  DISP=SHR,DSN=D640.V810.SDSNLOAD
   ```
//SYSUT1 DD DISP=OLD,DSN=D640.BSDS01
//SYSUT2 DD DISP=OLD,DSN=D640.BSDS02
//SYSPRINT DD SYSOUT=* 
//SYSUDUMP DD SYSOUT=* 
//SYSIN DD *

DDF LOCATION=D640,LUNAME=SAPD0013, 
NOPASSWD,RESPORT=7626,PORT=7625 

//*
DB2 Connect Installation and Customization

All of the customizing but not the installation for DB2 Connect takes place automatically during the SAP Web Application Server 6.40 installation process.

This information serves as a guide in case you need to make these changes manually (for example: customizing for failover, multiconnect setup or a FixPak upgrade)

If you are using a 6.40 downwards compatible kernel with a previously installed 6.20 based SAP system and now plan to migrate from DB2 V7 using ICLI to DB2 V8 using DB2 Connect refer to SAP Note 731937.

The following topics are covered in this section:

- Customization of DB2 Connect installation to fit the requirements of an SAP system
- Enablement of the DB2 Connect gateway mode
- Disablement of DB2 Connect connection pooling
- SAP environment variables for use with DB2 Connect
- Set-up of user IDs and passwords
- db2radm tool for DB2 Connect configuration
- Remote database cataloging
- DB2 Connect binding
19 DB2 Connect Installation Prerequisites

Before using DB2 Connect with SAP on an application server, you need to make sure that you have successfully installed DB2 Connect Unlimited Edition or another version of DB2 Connect which satisfies the DB2 licensing conditions. For more information, see *Quick Beginnings for DB2 Connect Enterprise Edition* (IBM Document-No. GC09-4833-00) and *IBM DB2 Connect User’s Guide* (IBM Document-No. SC09-4835-00).

You need to install DB2 Connect V8.15 FixPak Version 6 or higher. Consult the SAP Note 81737 for the current FixPak version.

In addition, you must have successfully created a 64 Bit DB2 instance during or after the DB2 Connect installation process. When you are logged on as the DB2 instance user, you can check to see if you have done this by entering the `db2level` command at the prompt. The following is an example of the output you would receive (using FixPak 6):

```
DB21085I Instance "db2inst1" uses "64" bits and DB2 code release "SQL08015" with level identifier "02060106". Informational tokens are "DB2 v8.1.1.48", "s040212", "U496793", and FixPak "6". Product is installed at "/usr/opt/db2_08_01".
```

There are two ways to install DB2 Connect:

1. via the GUI-based DB2 Connect installation process that guides you through the installation;
2. by using a response file.

In case you decide to install DB2 Connect with a response file, you may base your response file on the sample response file that DB2 Connect provides. This file is contained in the following directory of the DB2 Connect installation CD:

```
db2/platform/samples
```

While most of the parameters of the response file can be set individually, the following parameters need to be set to specific values for SAP systems, *(<instance_name> stands for the DB2 instance name):*

```
<instance_name>.TYPE = wse
```

This setting installs DB2 Connect in server mode.

```
<instance_name>. WORDWIDTH = 64
```

This creates a 64-bit instance.

To install DB2 Connect with minimal footprint, specify the following parameter setting:

```
INSTALL_TYPE = COMPACT
```
Setting Up DB2 Connect User ID and Password

DDF checks the user authorization at connect time using an RACF user ID (the DB2 Connect user ID). If you are using an application server on z/OS, you can use user <sapsid>adm that is defined on z/OS as DB2 Connect server. See the SAP Planning Guide for z/OS for more information on <sapsid>adm.

The z/OS group ID <SCHEMA> must be defined as a DB2 secondary authorization ID of the DB2 Connect user. The <SCHEMA> defines the schema of the DB2 objects of the SAP system as specified by the environment variable dbs_db2_schema.

On the client side, the DB2 Connect user needs to be specified by the following:

- environment variable: dbs_db2_user
- profile parameter: dbs/db2/user

After setting dbs_db2_user in the environment, call the command

```
dbdb2pwd -create <connect_user_password>
```

as <sapsid>adm to create the encrypted password file dbdb2pwd in the respective global directory:

- UNIX: /usr/sap/<SID>/SYS/global
- Windows: \\<SAPGLOBALHOST>\sapmnt\<SID>\global

Since the global directory is shared by all application servers of an SAP system, the password file is as well. Therefore, you only need to control the password file on one instance.

⚠️ We strongly recommend that you create a DB2 Connect user in RACF without TSO segment and set the password without an expiration date. If the password expires, new processes cannot connect to the database, and the reconnect mechanism for all processes will no longer function until the password file is recreated with the new valid password.

We recommend that every SAP system uses a DB2 Connect user of its own. This ensures that if the password is supplied incorrectly for one system and the user is revoked by RACF, the effect does not propagate to other SAP systems which are running with the same DB2 Connect User. For more information, see SAP Note 731937, which is step-by-step documentation on how to customize DB2 Connect for Web AS 6.20 and DB2 V8.
21 DB2 Connect Configuration Using db2radm

The tool db2radm is used to catalog the remote database and bind the Collection ID for the SAP system. It will check/correct the DB2 Connect registry variables and the Database Manager Configuration. We strongly recommend that you use this tool instead of carrying out these tasks manually.

Format


Description

db2radm does the following:

- checks the Connect registry variables and database manager configuration
- updates these settings if the check is not successful
- starts the DB2 Connect instance if the instance is stopped
- catalogs the remote database with DDF port and location as specified by options -P and -L. The connection profile will be used if one of the options is not specified. The remote database will be cataloged at the remote host as specified by the SAP environment variable SAPDBHOST (if it has not been not overwritten by option -H.) The remote database will be cataloged using Database Alias, DCS name and node name as specified by the SAP environment variable dbs_db2_ssid (if it has not been overwritten by option -S.) If only option -B is specified, the catalog operation will be omitted.
- binds the CLI packages. The SAP naming convention for the Collection ID is used to bind the packages if not overwritten by option -C. If option -B no is specified, the bind will be omitted.

Details

Cataloging with Connection Profile

We recommend that you catalog a connection.

The use of a connection profile is not limited to failover.

To catalog connections when your connections are already specified in the connect.ini file, call:

db2radm –m db2i

db2radm will catalog the connections specified in the connect.ini file and bind the main connection. You must have specified the SSID, HOST, PORT and LOCATION for every connection.

If you have connections that are already cataloged with this name, they will be overwritten.

PORT and LOCATION can be queried on z/OS with the DB2 command DISPLAY DDF.
This following is an example of the *connect.ini* file with one application server and one member: (For more information on the *connect.ini* file, see SYSPLEX Failover Support and Connection Profile [Page 248], section “Connection Profile”.)

```
[DEFAULT_GROUP]
CON1=D8A1_on_ihsap1

[D8A1_on_ihsap1]
SSID=D8A1
HOST=ihsap1
PORT=7801
LOCATION=D8A0
```

If you do not need to bind, use the option `-B no`, for example call:

```
db2radm -m db2i -B no
```

If you only need to bind, (for example, if you install a new FixPak) call:

```
db2radm -m db2i -B only
```

**When to Bind**

- If you are using a FixPak for the first time with the DB2 subsystem.
- If you are using your own Collection ID (this is only necessary in special cases: multi connect, for example).
- The section below explains how you specify your own Collection ID.
- If you need a new Collection ID for a new Unicode flavor; note that non-Unicode or Unicode must be different because the bind options are different.

**When Not to Bind**

- If you add application servers on a different platform.
- For example, if you are running a Linux, AIX and a Windows application server, it is sufficient to bind the collection only once. You can use any platform to do this.
- If there is a connected thread in DB2 which currently uses the Collection ID.
- In this case, your bind will be blocked.

**Messages**

Messages are written to `db2radm.log` in the local directory.

**Cataloging Without Connection Profile**

To use the *db2radm* tool without connection profile, call:

```
db2radm -m db2i -L <LOCATION> -P <PORT>
```

where

- `<LOCATION>` is the DB2 location name
- `<PORT>` is the DDF listening port

to create the catalog entries for your SAP system and bind.

`db2radm` uses the SSID as specified by the environment variable `dbs_db2_ssid` as location alias, as node name and as dcs database name.

If you do not need to bind the SAP collection, use the option `-B no`. For example, call:
db2radm -m db2i -L <LOCATION> -P <PORT> -B no

**SAP Naming Conventions for a Collection ID**

- By default `db2radm` uses the hostname as specified by the environment variable `SAPDBHOST`. If you need to specify the remote hostname, use the option `--host <hostname>`.

- By default `db2radm` uses a generated Collection ID. In case you need to specify the Collection ID use the option `--collection <collection>`.

  The naming convention for the generated Collection ID is as follows:

  `SAPrr.vv.ffffU`

  where:

  - `rr` = DB2 Connect major version (2 digits)
  - `vv` = DB2 Connect minor version (2 digits)
  - `ffff` = DB2 Connect FixPack (4 digits)
  - `U` = Unicode label (This is left blank for non-Unicode.)

  For example: the Collection ID for FixPack 5 for DB2 Connect V8.1 non-Unicode is `SAP08.01.0005`

**Setting Up Multiconnect**

If you are using DB2 Connect as SAP database connectivity, the secondary connections to DB2 V8 or DB2 V7 are also handled by DB2 Connect.

Assuming the secondary database with subsystem ID `DSNX` is located on host `ihsapx`, with DDF location name `DSNXLOC` and DDF port `8888`, the schema is `SAPX` and the Collection ID `SAP08501`.

  9. Create a new DBCON entry where you specify connect user and password and connection information for the secondary connection.

  Set the connection information in the DBCON entry to:

  `SSID=DSNX;SCHEMA=SAPX;PS=SAP08501`

  10. Call the `db2radm` tool to create the catalog entries and bind:

      `db2radm -m db2i -C SAP08501 -H ihsapx -P 8888 -L DSNXLOC -S DSNX`
### 22 Manual Setting of DB2 Connect Parameters

The following section contains information about enabling the DB2 Connect gateway mode and disabling the default DB2 Connect connection pooling.

We strongly recommend that you use the `db2radm` tool to set the DB2 Connect parameters instead of doing it manually.

#### DB2 Registry Variables

Under Windows, you must open a DB2 Connect Command Window to run the DB2 commands listed below:

**Programs → IBM DB2 → Command Line Tools → Command Window**

In order to check which DB2 registry variables are currently set, issue the following command:

```
db2set -all
```

**Enabling the DB2 Connect Gateway Mode**

Gateway mode must be enabled so you can monitor the network of the database connection.

To enable the gateway mode in DB2 Connect you need to set the DB2 registry variable `DB2CONNECT_IN_APP_PROCESS` with the following command:

```
db2set DB2CONNECT_IN_APP_PROCESS=NO
```

The SAP application server and the DB2 Connect gateway should be on the same physical hardware. We do **not** recommend running any connections to the remote DB2 database through a real gateway which is located on a separate machine. This would only unnecessarily complicate the network, slowing down each request and thus having a negative impact on the performance of SQL requests.

#### Enabling SQL Break

To enable DB2 Connect to break running SQL statements, you must set the DB2 registry variable `DB2CONNECT_DISCONNECT_ON_INTERRUPT` with the following command:

```
db2set DB2CONNECT_DISCONNECT_ON_INTERRUPT=ON
```

**DB2CODEPAGE**

This variable must be set to code page 819 in a non-Unicode environment. There is no need to set this variable in a Unicode environment.

You find a list of all SAP environment variables, including the DB2 Connect-specific variables in [Environment Variables](Page 344).

**Disable DB2 Connect Connection Pooling**

By default, DB2 Connect uses connection pooling. This means that if you start and stop an SAP application, the DB2 threads that were created in DB2 on the z/OS host stay active although the remote application has been terminated. Connection pooling keeps the connection to these threads open so that DB2 can quickly re-use them for new DB2 Connect applications.

Connection pooling is beneficial for applications which connect and disconnect frequently. SAP applications usually do not do this. Therefore, you should disable connection pooling by setting the DB2 Agent pool size `NUM_POOLAGENTS` like this:
db2 update dbm cfg using NUM_POOLAGENTS 0

You can use the following command to find out the current value of the NUM_POOLAGENTS parameter for your DB2 instance:

`db2 get dbm cfg`
We strongly recommend that you use the `db2radm` tool for cataloging instead of doing it manually.

Each SAP database that you want to access needs to be cataloged in DB2 Connect using the set of commands listed below.

For a newly installed SAP database, cataloging is done automatically during the installation procedure.

Here we list the commands which are necessary if you need to do this manually.

You need to catalog a TCP/IP node, a database and a remote (DCS) database. The naming convention that you should follow is:

- `database alias = subsystem ID as specified in connect.ini, or the environment variable db_ssid`
- `node name = database alias`
- `dcs name = database alias`

**Example**

Under Windows, you must open a DB2 Connect Command Window to run the DB2 commands listed below:

```
Programs → IBM DB2 → Command Line Tool → Command Window
```

In the example below it is assumed that:

- the DB2 subsystem ID is `SG61`;
- the remote z/OS host is `saphost1`;
- DB2 on z/OS DDF is listening to port `5111` and that the remote DB2 location name is `SAPDSN1`;

```
catalog tcpip node SG61 remote saphost1 server 5111
catalog db SG61 at node SG61 authentication server
catalog dcs database SG61 as SAPDSN1
```
24 Manual Execution of DB2 Connect Bind Command

Description
This section describes how to perform a bind under DB2 Connect manually.

⚠️ We strongly recommend you use `db2radm` to issue the Bind command.

Prerequisites
To ensure optimal performance and for the functional reasons, SAP requires specialized bind options. These options are used when binding DB2 Connect V8.1 DBRMs to DB2 on z/OS.

⚠️ V8.1 CLI packages are bound by default to the NULLID collection in DB2 on z/OS. When using SAP applications, you should **not** use the NULLID collection, since some bind options (for example, `KEEPDYNAMIC YES`) needed for SAP applications must have certain defaults that are blocked by the NULLID collection. In order to override this, you must specify a different collection which will create a new set of packages with the options the SAP application requires.

Procedure
For this process, a specialized DB2 Collection ID needs to be created using a bind command similar to the one below. This Collection ID should then be used by the SAP Database Shared Interface when accessing DB2 on z/OS.

Under Windows, you must open a DB2 Connect Command Window to run the DB2 commands listed below:

Programs → IBM DB2 → Command Line Tools → Command Window

1. To execute the bind, you need to be connected to the DB2 subsystem on the z/OS host on which the SAP database is located:

   ```
   db2 connect to SG61 user <SAPUSER> using <PASSWORD>
   ```

2. Then you need to run the following bind command and create a new collection called as shown in these examples:

   UNIX:
   ```
   db2 bind <Instance directory>/sqllib/bnd/@ddcsmvs.lst ACTION REPLACE KEEPDYNAMIC YES GENERIC "DEFER PREPARE" REOPT ONCE COLLECTION SAP<rr.vv.ffff><U> ISOLATION UR BLOCKING UNAMBIG RELEASE COMMIT GRANT PUBLIC SQLERROR CONTINUE
   ```

   Windows:
   ```
   db2 bind %DB2TEMPDIR%\bnd\@ddcsmvs.lst ACTION REPLACE KEEPDYNAMIC YES GENERIC "DEFER PREPARE" REOPT ONCE
   ```
COLLECTION SAP<rr.vv.ffff><U> ISOLATION UR BLOCKING UNAMBIG
RELEASE COMMIT GRANT PUBLIC SQLERROR CONTINUE

⚠️

You must use the parameters exactly as listed above, otherwise you may experience unexpected behavior in your SAP system.

For more information on the collection naming convention, see DB2 Connect Configuration Using db2radm [Page 88].

3. Disconnecting from the DB2 subsystem:

```
db2 disconnect SG61
```
Performance Tuning Considerations

Performance monitoring and tuning in the SAP system environment is a complex and challenging task. The following sections are intended to be a collection of tuning steps that have shown to be notably relevant and beneficial with respect to SAP systems. Depending on your requirements, additional tuning steps will be necessary.

To evaluate the effects of tuning and to detect the development of new bottlenecks and performance deficiencies, you need to establish a basis for performance evaluation. This can be done by collecting and storing the performance data over a longer period of time, but most importantly before and after any tuning activities.

After installation, you should also observe most of the aspects documented in the “Performance Tuning Considerations Before Installing an SAP System” in the Planning Guide: z/OS Configuration for SAP on IBM DB2 UBD for z/OS.

The following is a short summary of the information contained in the section “Performance Tuning Considerations Before Installing an SAP System” of that document and in DB2 Setup [Page 48]:

- Apply the recommended service to the z/OS and DB2 code levels.
- To prioritize work according to your objectives, use WLM.
- Monitor and tune the ICF catalog performance.
- Observe the required and highly recommended DB2 system parameters in DB2 Setup [Page 48].
  
  The system parameters that are categorized as recommended initial values should be adjusted based on the site-specific workload.
- Switch off unnecessary traces.
  
  The only traces that should be active in addition to those recommended in section DB2 Setup in the SAP system installation documentation are:
  - DB2 Accounting trace classes 2 and 3. These classes should be active most of the time. Only during major imports of data, for example, Client Copy or at times of high overall CPU utilization of the system can you consider switching off the accounting class 2 trace.
  - DB2 performance trace IFCID 318. This trace provides valuable data for the statement scope statistics.
  - DB2 statistics trace IFCID 199. This trace provides data set statistics.
  
  The SAP system aims to ensure that the DB2 Accounting trace classes 2 and 3 and the performance trace IFCID 318 and statistics trace IFCID 199 are always on. For more information, see Automatic Start of DB2 Traces [Page 171].
- Maintain the recommended SAP profile parameters settings.

The following sections describe some of the considerations that are relevant after the installation of your SAP system and for daily usage of the component.

- Virtual Storage Considerations [Page 98]
- Setting Optimal SAP Profile Values [Page 104]
- Customizing the SAP Objects Topology [Page 106]
- Ensuring Optimal Access Paths [Page 107]
- Clustering Index [Page 121]
- Partitioning Key [Page 122]
• Locking Considerations [Page 125]
• Buffer Pool Tuning Considerations [Page 130]
• Dynamic Statement Caching Considerations [Page 131]
• DB2 Accounting and Workload Management (WLM) [Page 132]
25 Virtual Storage Considerations

Dynamic statement caching and long running DB2 threads are characteristics of the SAP system environment that puts a strain on the virtual storage in one of the DB2 address spaces, DBM1. While many storage structures reside above the 2 GB bar of DBM1 and exploit 64-bit virtual storage, there are a few structures that jointly use the space below this bar. The following sections provide estimates on storage consumption by most of the resources typically used in the SAP system, as well as recommendations for virtual storage planning, monitoring, and tuning.

Recommendations

To avoid problems related to the excessive usage of virtual storage below the 2 GB bar, you should follow these recommendations:

1. Make sure your system has the current DB2 service level, particularly regarding fixes and enhancements in the area of storage usage. This involves the following:
   o Apply cumulative preventive maintenance, usually two to four times a year.
   o On a monthly basis, apply HIPERs marked as pervasive and HIPERs relevant to your workload.

   For information about DB2 PTFs that are critical for SAP system environments, see SAP Note 81737.

2. Do not overcommit virtual storage by specifying inappropriate parameters that influence its usage. The following is a list of the most important consumers of the DBM1 storage below the 2 GB bar:
   o Thread storage
   o Local dynamic statements cache storage
   o Open data sets
   o System storage
   o EDM pool
   o RID map blocks of RID pool
   o Storage cushion

   The storage consumption for each of these resources is influenced by their corresponding parameters settings. For more information, see Virtual Storage Usage Estimates [Page 100]. In any case, make sure you leave at least 250 to 300 MB virtual storage free below the 2 GB bar in DBM1 as “headroom” for performance tuning, workload growth, and service ability.

3. Set the DB2 system parameter CONTSTOR to YES.

4. Activate the following SAP profile parameter rdisp/wp_auto_restart to periodically recycle SAP work processes. That will cause the deallocation of the corresponding DB2 threads and, consequently, the release of the associated storage. Initially set the value to 86400. For more information about the proper use of this parameter, see Setting Optimal SAP Profile Values [Page 104].

5. Regularly monitor virtual storage usage.

RMF provides historical and snapshot reporting on the virtual storage. You can use it to detect trends and determine if there is enough storage left for increasing DB2 pools (RID pool, local dynamic statement cache, and so on) to increase overall system performance.
25.1 Estimates on Virtual Storage Usage Below the 2 GB Bar

In the calculations that follow, many of the constant and parameter values are averages that are likely to match most SAP system installations. Be aware, however, that the exact values depend on your particular workload characteristics, and hence can vary. You should therefore only regard the information presented here as estimates of the virtual storage usage.

Thread Storage

Let T be the number of allocated threads. The number of allocated threads will be equal to the sum of the number of SAP work processes that connect to the DB2 subsystem and the secondary connections that are created by SAP work processes at the DB2 subsystem.

T is not necessarily equal to the sum of the system parameters MAXDBAT and IDBACK which is typically somewhat larger to allow for a possible increase in the number of SAP work processes, secondary connections and other background connections.

The total amount of storage that needs to be reserved for the allocated threads is calculated as follows:

thread storage = T * 2.0 MB

The system parameter CONTSTOR must be set to YES.

Storage for Local Dynamic Statement Cache

Let M be the larger value of these two:

- \( \text{MAXKEEPD} \times 1.1 \)
- Maximum number of statements whose skeletons (SKDS) can be cached in the EDM Statement Cache. As a rule, assume that the average SKDS size is 30 KB.

The total amount that needs to be reserved for this storage is calculated as follows:

Local dynamic statements local cache storage = M * 3.5KB + \( \text{MAXKEEPD} \times 1.1 \times 40 \) KB

We recommend that you start with \( \text{MAXKEEPD} = 8000 \) and adjust it downwards based on the monitoring of virtual storage usage and local cache hit ratio. For the EDM Statement Cache, we recommend that you initially set the DB2 system parameter EDMSTMTC to a value of 300000. Adjust it upwards based on the monitoring of the global cache hit ratio. To avoid multiple EDM Statement Cache entries for the identical statement, make sure that you connect from all application server to DB2 using the same authorization ID.

Storage for Open Data Sets

The total amount that needs to be reserved for this storage is calculated as follows:

storage for open data sets = \( \text{DSMAX} \times 1.5 \text{ KB} \)

\( \text{DSMAX} \) is the system parameter that controls the number of concurrently opened data sets. We recommend that you leave \( \text{DSMAX} \) at its default value, which is 10000.
**System Storage**

There is a fixed amount of storage that DB2 needs for its operations. It varies depending on whether data sharing is used or not:

- No data sharing: 168 MB
- Data sharing: 354 MB

**EDM Pool**

Plans and packages reside in the EDM pool. The size of the EDM pool is defined by the user using system parameter `EDMPOOL`.

**RID Pool**

This pool is used for list prefetch and multiple index access processing. Its total size is specified using the system parameter `MAXRBLK`. It is split between a RIDMAP pool, which resides below the 2 GB bar, and a pool above the 2 GB bar that contains the RIDLISTS. The size of the RIDMAP pool is calculated as follows:

`MAXRBLK * 0.25`.

**Storage Cushion**

In extreme situations, when virtual storage use is critically high, DB2 triggers a process that releases lots of storage. This process has a considerable CPU overhead, so you should take steps to prevent it from being used.

The process is triggered when a certain combination of the maximum number of open data sets (`DSMAX`), the maximum number of allocated threads (`CTHREAD`) and maximum number of DDF threads (`MAXDAT`) reached. For a rough estimate, use the following calculation:

\[
\frac{(40000 + (DSMAX * 500) + (20^*(CTHREAD + MAXDBAT) * 2000))}{(1024*1024)}
\]

If free storage in DBM1 is equal to or lower than this value, the storage cushion process will be triggered. Obviously, you should take care to avoid overallocating any of the constituent parameters as it unnecessarily increases exposure of frequently hitting the threshold.

**Example**

This example demonstrates estimating the storage. Assume that there are 300 SAP work processes that connect to DB2 via DRDA and the relevant system parameters are set as follows:

- `EDMPOOL = 32M`
- `MAXKEEPD = 8000`
- `DSMAX = 10000`
- `MAXRBLK = 100M`
- `IDBACK = 20`
- `CTHREAD = 30`
- `MAXDBAT = 390`
- The DB2 system is non-data sharing.

The following amount of storage should be reserved:

- **Thread Storage**: 300 * 2.0 MB = 600 MB
- **Local Dynamic Statements Cache Storage**: 10000 * 1.1 * 3.5KB + 8000 * 1.1 * 40KB = 381 MB
- **Open Data Sets**: 10000 * 1.5KB = 15 MB
- **System Storage**: 168 MB
- **EDM Pool**: 32 MB
- **RID Pool**: $100 \times 0.25 = 25$ MB
- **Storage Cushion**: 17 MB

This makes a total of around 1238 MB. You should never exhaust all of the available storage as some of it is used for smaller, but still important storage pools. Also, the sizes given for the threads storage and dynamic statements cache storage can occasionally be exceeded, in which case you need some reserve.
25.2 Virtual Storage Monitoring

Regular monitoring of the virtual storage usage can help you to avoid situations where severe storage shortages are encountered. The RMF MONITOR I VIRTUAL STORAGE: PRIVATE AREA report provides a historical view of virtual storage usage and identifies potential storage shortages. Use the following calculation to find out the amount of storage available above the 16 MB line:

Available storage = X - Y - Z where

X = REGION ASSIGNED
Y = MAX LSQA/SWA/229/230 PAGES ALLOCATED
Z = MAX USER REGION PAGES ALLOCATED

The following graphic is an example of the relevant RMF report. The values used in the above calculation are in italics. In this particular example, the amount of available storage is:

1862 – 1273 – 567.3 MB
26 Setting Optimal SAP Profile Values

There are numerous SAP profile parameters that are extremely important for a well-performing SAP system. The sizes of application server storage areas for buffering SAP objects and the number and type of work processes are only a couple of them that clearly indicate the importance of setting them correctly. The SAP online documentation provides lots of details on these parameters; SAP Basis consultants and EarlyWatch service are likely to set or recommend the values that are optimal for the circumstances characteristics of your installation.

In any case, make sure that these selected SAP profile parameters have the following values:

- `rsdb/prefer_fix_blocking = 1`
- `rsdb/prefer_in_itab_opt = 1`
- `rsdb/prefer_union_all = 1`
- `rsdb/max_blocking_factor = 10`
- `rsdb/min_blocking_factor = 3`
- `rsdb/max_in_blocking_factor = 35`
- `rsdb/min_in_blocking_factor = 6`

The SAP profile parameters are contained in `/usr/sap/<SID>/SYS/profile`. 
27 Periodically Recycling SAP Work Processes

To reduce the amount of virtual storage that is accumulated by the DB2 threads that serve SAP work processes, it is recommended to periodically recycle SAP work processes. This deallocates the DB2 threads, which frees the accumulated storage, and allocates new threads.

The SAP profile parameter that controls the duration of DB2 threads is `rdisp/wp_auto_restart`.

- The value of `rdisp/wp_auto_restart` is given in seconds. The timer gets initialized for the first time at application server startup. Whenever a work process finishes a dialog step and its context is rolled out, the timer is checked. If expired, the work process is restarted (terminated and created again) and the timer reset. The corresponding DB2 thread is deallocated (freeing up all the accumulated storage) and then a new thread is allocated.

  This technique ensures that neither a dialog steps nor batch jobs are cancelled. This means that you can use it without affecting the application flow.

- The value for the parameter depends on the virtual storage usage that is specific to a particular customer’s installation. A value that is too low has a negative impact on system performance. We recommend the following value as the initial settings:

  `rdisp/wp_auto_restart = 86400`

- The value should be adjusted accordingly based on regular virtual storage monitoring, which is described in Virtual Storage Monitoring [Page 103].
28 Customizing the SAP Objects Topology

After you install an SAP system component, there are a large number of data sets backing thousands of related DB2 table spaces and index spaces. For a detailed description of how the SAP tables and indexes map into the DB2 stogroups, databases, and table spaces, see Storage Management [Page 24].

It is very likely that the tables/indexes-to-volumes mapping will not be optimal for your DASD. For example, two or more heavily used tables or indexes may reside on the same volume, causing DASD contention and long I/O times when accessing the tables and indexes.

Unless your DASD subsystem supports contention-reducing features such as ESS Parallel Access Volume and Multiple Allegiance, you should find out which tables and indexes are most frequently accessed in your environment and isolate them on separate volumes. Call transaction ST10 to determine the access frequency and pattern on a per-table basis.

In general, SAP data compresses very well. For more details on how to decide whether to compress a table space and how to do it, see Data Compression [Page 322].
29 Optimal Access Paths Assurance with RUNSTATS

The DB2 Optimizer is cost-based. The access path for a given statement is usually influenced by the following:

- Statistics for tables referenced in the statement and associated objects, such as tablespaces, indexes and columns
- Size of the buffer pool
- Central processor model

The statistics are stored in a number of DB2 catalog tables. DB2 provides the RUNSTATS utility that collects the necessary statistics and updates the catalog. The statistics collected by RUNSTATS can also be used as an indicator whether a tablespace or index should to be reorganized, and can also be used for calculating the space requirements for a table.

The most important questions about using RUNSTATS are:

- When is RUNSTATS due?
- Which RUNSTATS options should be used?
- Will updating catalog statistics with RUNSTATS ensure optimal access paths?

The following sections address these questions and give practical advice on maintaining the catalog statistics in SAP system environments.

For full details on RUNSTATS and other DB2 utilities referenced here, see DB2 for z/OS Utility Guide and Reference.
29.1 When RUNSTATS Is Due

Outdated statistics are one of the most common reasons for DB2 not selecting the optimal access path for a given statement. For example, the table's size and cardinalities of its columns can significantly change as a result of heavy insert activity. If RUNSTATS has not been run after such an activity, the DB2 Optimizer bases its selection of the access path on outdated input, resulting in a less than optimal access path.

RUNSTATS and CPU Performance

RUNSTATS uses a considerable amount of CPU, which might not always be available. If you run it indiscriminately on all tables too frequently, this does not necessarily result in better input for the selection of the access path.

For example, the statistics for a large, static table might not change enough to warrant spending CPU on refreshing it, in other words, setting the values to what they were before.

In addition, if RUNSTATS is run on all the tables, it opens all the underlying data sets and a number of them (close to the system parameter DSMAX) remain open. This has an impact on the amount of available storage, the restart after an abnormal termination and shutdown times.


DB2 real-time statistics (RTS) provide an easy way to detect tables on which RUNSTATS needs to be run. When a row is inserted, deleted and changed in a table, DB2 keeps track of this change in its RTS, which reside in memory and which DB2 periodically externalizes to the catalog tables SYSTABLESPACESTATS and SYSTABLESPACESTATS. The DB2 stored procedure DSNACCOR analyzes these tables and – based on different criteria - deduces a list of tables that should be equipped with new statistics. The SAP system exploits DSNACCOR hourly and provides the list of tables for which RUNSTATS is recommended in the CCMS Monitor Set (transaction RZ20). Also, the job Update statistics of recommended objects in the DBA Planning Calendar (transaction DB13) works on the tables that are recommended by DSNACCOR. As DB2 itself keeps track of the database changes and formulates the recommendations, DSNACCOR and the CCMS Monitor Set provides for a very precise and efficient way to determine the tables that need new statistics.

Determining Tables Needing New Statistics: ST10 and the COPY Utility

Alternative ways of deciding if RUNSTATS is due are transaction ST10 and the COPY utility with the CHANGELIMIT REPORTONLY option.

Transaction ST10

Transaction ST10 is a very useful tool for monitoring workload on a per-table basis. Its output is a list of tables and the operations statistics for the tables, such as the number of INSERTS, UPDATES and DELETES for various time intervals: since startup, today, previous day, this week, previous week, this month, previous month. You can use these statistics to assess which tables need RUNSTATS: in general, those with a considerable number of changes (relative to the table's size) since the last time RUNSTATS was run.

Transaction ST10 is not as accurate as DSNACCOR. For example, the ABAP SQL clause 'SELECT SINGLE' is counted as a random access. However, the actual access path that the DB2 optimizer chooses may rather be a sequential read.
COPY Utility
Running the COPY utility with the CHANGETLIMIT REPORTONLY option on a particular object produces a report (without actual copying) that includes the percentage of pages changed since the last time an image copy was taken.

⚠️
This method assumes that RUNSTATS is run every time an image copy is taken.

When to Schedule RUNSTATS
RUNSTATS should be scheduled:

- As soon as convenient for the tables with a considerable number of changes.
- After the initial load, migration and upgrade.
  
  There is a separate step in the SAP installation procedure where RUNSTATS is performed for all the tables in the system. RUNSTATS should be run at this time for catalog tables as well, because there is a large number of new database objects.
- It is important to run RUNSTATS not only after, but also at least once during the process (in the first quarter, not too soon after the start) if batch input is used to import data into the system. Batch input includes queries as well as inserts, and the queries need current statistics in order to use optimal access paths.
- For the tablespaces and indexes that have just been reorganized.
  
  The most efficient way to accomplish this is an inline RUNSTATS execution (the REORG’s STATISTICS option).
- For the tablespaces and indexes that have just been recovered.
- For the newly created indexes.
  
  If the index is created using the REBUILD utility consider inline RUNSTATS invocation.
- For newly created and populated tables.
  
  You can identify tables for which RUNSTATS was never executed by checking the STATSTIME catalog column. The value is ‘0001-01-01.00.00.00.000000’ for tables with no RUNSTATS.
- To invalidate cached statements.
  
  Cached statements are implicitly invalidated every time a RUNSTATS TABLESPACE is run on a tablespace that contains a table that is referenced by these cached statements. A statement is invalidated to let the DB2 optimizer reprepare it and take updated statistics into account that were not available the first time the statement was prepared. If RUNSTATS TABLESPACE is only run to invalidate cached statements, it should be executed with the options REPORT NO UPDATE NONE. Specifying both options prevents DB2 from actually scanning the tablespace, saving CPU resources. The only effect of RUNSTATS TABLESPACE with these options is that cached statements that reference at least one of the tables from the tablespace are invalidated.

⚠️
Creating a new index implicitly invalidates the caches statements associated with the base table of the index.

Note that the SAP BW system automatically schedules RUNSTATS with appropriate options for star schema tables when necessary. For example, after new data has been loaded.
## 29.2 RUNSTATS Options To Be Used

In general, the following **RUNSTATS** specifications are recommended for the SAP system environment:

- **To collect and update catalog statistics for all the tables in a tablespace:**
  ```
  RUNSTATS
  TABLESPACE <tablespace name>
  TABLE (ALL)
  SAMPLE
  INDEX (ALL)
  KEYCARD
  SHRLEVEL (CHANGE)
  ```

- **To collect and update catalog statistics for a single table in a given tablespace:**
  ```
  RUNSTATS
  TABLESPACE <tablespace name>
  TABLE (table name)
  SAMPLE
  INDEX (<index1 name> KEYCARD, <index2 name> KEYCARD, ...)
  SHRLEVEL (CHANGE)
  ```

- **To collect and update catalog statistics for a given index:**
  ```
  RUNSTATS
  INDEX (<index name> KEYCARD)
  SHRLEVEL (CHANGE)
  ```

- **To collect and update catalog statistics for a LOB tablespace:**
  ```
  RUNSTATS
  TABLESPACE <name of auxiliary tablespace>
  SHRLEVEL (CHANGE)
  INDEX (ALL)
  ```

- **To invalidate a cached statement that references a table in <tablespace name>:**
  ```
  RUNSTATS
  TABLESPACE <tablespace name>
  SHRLEVEL (CHANGE)
  REPORT NO UPDATE NONE
  ```

### **RUNSTATS FREQVAL Option**

In most cases, the **RUNSTATS** options specified above provide the catalog statistics necessary for selecting the optimal access path. However, sometimes additional catalog statistics can be beneficial for example it the values of columns that are not at the first position of an index are skewed or for correlated columns. This is the frequency distribution for individual columns or combinations of concatenated key columns and it is collected if the **RUNSTATS FREQVAL** option is specified. By default, **RUNSTATS** collects frequency distributions for the 10 most frequently occurring values of the first column of an index.
REOPT(ONCE) Bind Option

The DB2 optimizer can only exploit these frequency distributions if no parameter markers are present during optimization. As SAP BW uses literals in its OLAP queries, frequency distribution can always be used for these queries. However, SAP normally uses parameter markers.

The DB2 optimizer can take advantage of frequency distributions for statements with parameter markers under the assumption that the bind option REOPT(ONCE) is used, which is highly recommended in SAP environments. This causes the DB2 optimizer to defer query optimization until the first set of host variable values for parameter markers is provided. Without REOPT(ONCE), the DB2 optimizer does not know the values that the application will provide for parameter markers and thus assumes default values. They often differ considerably from the values that SAP will provide leading to non-optimal access paths. Prepared statements are executed using the same access path, even when the application provides different sets of values for parameter markers.

ABAP hints that influence the DB2 optimization process (%_HINTS DB2 'USE VALUES FOR OPTIMIZATION', 'SUBSTITUTE VALUES', 'SUBSTITUTE LITERALS') are no longer necessary if a single access path satisfies the performance requirements for all values that are assigned to the parameter markers of a statement. In this case, you might consider removing the ABAP hints that were declared prior to DB2 V8, which introduced the bind option REOPT(ONCE), to decrease the pressure on the dynamic statement cache.

Single-column frequency distributions particularly benefit statements with Boolean term predicates that involve IN or range predicates.

STATISTICS Keyword

DB2 allows you to collect catalog statistics inline within the REORG and REBUILD INDEX utilities. This feature is requested by the STATISTICS keyword followed by usual RUNSTATS options on the REORG and REBUILD specifications.

RUNSTATS Jobs in Parallel

If you need to run RUNSTATS for a large number of objects, consider running multiple RUNSTATS jobs in parallel.

Running RUNSTATS jobs in parallel can reveal an error in the z/OS setup. The Scheduler Work Area (SWA) must be defined above the 16 MB line. See the SAP documentation Planning Guide: z/OS Configuration for SAP on IBM DB2 UDB for z/OS.

More Information About RUNSTATS

For a full description of the RUNSTATS options and job specifications, see the DB2 for z/OS Utility Guide and Reference.
29.3 Access Path Considerations for Volatile Tables

In the majority of cases, updating catalog statistics with RUNSTATS ensures optimal access paths. However, there are some important exceptions that must be addressed differently. These cases include some special purpose tables that have to be accessed in a particular way regardless of their catalog statistics. The access path considerations for these cases are described in Access Path Considerations for Special SAP Tables [Page 113].

Special Considerations for Volatile Tables

Special considerations also apply to volatile tables. If RUNSTATS is run on a table that is empty or very small (occupying only a few pages), the statistics collected at that time can be very misleading if there are subsequently a large number of inserts in the table. This can often happen on tables with transient data, such as update log tables (VBDATA, VBMOD, VBHDR), or any kind of queue tables in general. In that case, the wrong access path can also cause heavy lock contention, including deadlocks.

Examples for Other Volatile Tables

Other examples are tables that get archived and significantly reduced in size, but only temporarily. Batch input is also prone to such problems. Typically the tables are empty to start with and then grow very rapidly in size. Queries are also operating on them during this process. A wrong access path can also cause a heavy lock contention, including deadlocks.

DB2 System Parameter NPGTHRSH

Fortunately, we can avoid most of these problems by telling the Optimizer to use a heuristic approach instead of cost based optimization if, and only if, the tables are small (for example, less than 10 pages) or empty. For such a table, the following access path should be selected:

If there is an index that has at least one matching column for given statements’ predicates, a tablespace scan should be avoided. If there are more than one such an index, choose the one with the largest number of matching columns. If there are still more than one qualifying, then choose the one that provides ordering (if applicable). Neither list prefetch nor sequential prefetch should be selected.

The way to achieve this is simple: You only need to set the DB2 system parameter NPGTHRSH. Its value is taken into account during access path selection. For a given table, if NPAGES is less than the NPGTHRSH value and not -1, an index access for the table will be preferred over a tablespace scan. The recommended value is 10.

Note that volatile tables are addressed by the system parameter NPGTHRSH and that SAP special tables are addressed by the DB2 table attribute VOLATILE (see next section for details on SAP special tables).
29.4 Access Path Considerations for Special SAP Tables

Use

The following special SAP tables need to be accessed in a special way:

- **Asynchronous Update Protocol Tables:**
  - VBHDR
  - VBMOD
  - VBDATA

- **TRFC and QRFC Tables:**
  - ARFCSDATA
  - ARFCSSTATE
  - ARFCRDATA
  - ARFCRSTATE
  - TRFCQDATA
  - TRFCQSTATE
  - TRFCQOUT
  - TRFCQIN
  - TRFCQINS

- **SAP Cluster Tables:**
  A complete list of these tables is returned by the query:

  ```sql
  SELECT * FROM <SCHEMA>.DDNTT
  WHERE TABFORM = 'T' AND TABTYPE = 'C' WITH UR
  ```

- **ABAP Export/Import Tables:**
  These tables are also called ABAP clusters. A complete list of these tables is returned by the query:

  ```sql
  SELECT TBNAME FROM SYSIBM.SYSCOLUMNS X
  WHERE TBCREATOR='<SCHEMA>'
  AND NAME = 'CLUSTD'
  AND COLNO = (SELECT MAX(COLNO) FROM SYSIBM.SYSCOLUMNS
  WHERE TBCREATOR = '<SCHEMA>'
  AND TBNAME = X.TBNAME)
  ```

Purpose

All of these special tables must be accessed in a particular way in order to minimize deadlock occurrences and optimize their performance. Their optimal access path should not be cost-based (like for a large majority of other tables) but a matching index scan with neither sort nor list prefetch. If the tables are not accessed as described above, there is a possibility of increased lock contention including deadlocks. It is also likely that a less than optimal access path will be selected.
Solution

To make sure that these tables are accessed properly, the table attribute VOLATILE is set when they are created. This ensures that the DB2 Optimizer chooses the described access path for statements accessing these tables. If some of these tables were created prior to DB2 V8, you may run report RSDB2VOLATILE in transaction SE38 to set the VOLATILE attribute on all special tables.

Transport M7EK000181, which is located in directory general/R3server/abap/note.0417920/relindep on sapserv, makes this report available for older SAP releases. This transport can be imported into any SAP release.
29.5 Access Optimization to VBHDR, VBMOD and VBDATA

The asynchronous update protocol tables (VBHDR, VBMOD, VBDATA) have a special purpose in the SAP system and are generally very frequently accessed. Consequently, it is very important to ensure optimal performance for the statements that refer to the tables.

There are three major areas in tuning the VB protocol tables. They are described in:

- Optimal Access Paths Assurance for Table Access Statements [Page 116]
- Assigning the VB Protocol Tables to Dedicated Buffer Pools [Page 117]
- Partitioning VB Protocol Tables [Page 119]
29.5.1 Optimal Access Paths Assurance for Table Access Statements

Among other requirements listed in Access Path Considerations for Special SAP Tables [Page 113], it is important that you ensure that only one index is defined for each of the VB protocol tables (VBHDR, VBMOD, and VBDATA). This is the primary index created when an SAP system is installed. Determine if any additional indexes are defined on the tables by checking the:

- **ABAP Dictionary**
  
  Call transaction SE11 and enter the name of a VB protocol table. On the next panel, choose *Indexes*. If any index other than the primary one exists, delete it.

- **DB2 Catalog**
  
  Use the query:
  
  ```sql
  SELECT      TBNAME, NAME, UNIQUEROLE
  FROM        SYSIBM.SYSINDEXES
  WHERE       TBCREATOR='<SCHEMA>'
  AND         TBNAME IN ('VBMOD', 'VBHDR', 'VBDATA')
  ORDER BY    TBNAME, NAME;
  ```

  There should be only one index for each of the tables. It can be identified by the index name suffix (it is 0 for primary index) and by the `UNIQUERULE` value equal to “P”.
29.5.2 Assigning the VB Protocol Tables to Dedicated Buffer Pools

Use

When you install the SAP system, all SAP tables, (including the VB protocol tables VBDATA, VBHDR, VBMOD), are assigned to common buffer pools and VBDATA is defined with a page size of 32 KB.

We highly recommend that you assign the VB protocol tables to a separate buffer pool whose attributes match the tables' access pattern.

The following describes how to accomplish this objective.

Some of the steps are also required for partitioning the VB protocol tables; see Partitioning VB Protocol Tables [Page 119].

Procedure

1. Make sure the SAP system is unavailable to others during the procedure.
2. Call transaction SE16 to check whether the update tables VBHDR, VBMOD, and VBDATA are empty. If they are not, process all of the outstanding updates with SAP transaction SM13. You cannot continue with the subsequent steps until all of the outstanding updates have been processed.
3. Import the transport KDOK000668 into your system.

   It is located in the directory:
   
   sapservX:~tp/general/R3server/abap/note.0122599,
   
   ... The file names are:
   
   KDOK000668
   KDOR000668

   As a result, the length of the domain VBDATA is shortened to 3800 characters so that it fits into a tablespace with a 4 KB page size and each of the tables is isolated in a 4 KB tablespace of its own.

4. Check whether each of the tables VBHDR, VBMOD, and VBDATA has been placed in a separate 4 KB tablespace. If this is not the case, use SAP transaction SE14 to isolate them manually. For more information, see Storage Management → Storage Parameters → Partitioning Tables [Page 310].
5. Stop the VBHDR, VBMOD, and VBDATA tablespaces by stopping their associated databases.
6. Assign the tablespaces and indexes to an unused buffer pool, for example, BP4:

   ALTER TABLESPACE database-name.tablespace-name BUFFERPOOL BP4

7. Start the databases again.
8. Set the buffer pool parameters:

   ALTER BUFFERPOOL(BP4) VPSIZE(1000) VPSEQT(10) DWQT(70) VDWQT(50)
Post-Assignment Activities

After you have made the changes, monitor the buffer pool (BP4). If any of the critical thresholds are reached, apply the buffer pool tuning techniques described in Buffer Pool Tuning Considerations [Page 130].
29.5.3 Partitioning VB Protocol Tables

Use

This step is considered to be an advanced tuning procedure that is especially beneficial in data sharing environments. In non-data sharing environments, it should be applied when the performance related to the VB protocol tables is not satisfactory even after the tasks described in Optimal Access Paths Assurance for Table Access Statements [Page 116] and Assigning the VB Protocol Tables to Dedicated Buffer Pools [Page 117] have been performed.

Procedure

Make sure the SAP system is unavailable to others during the procedure.

1. Call transaction SE16

   With this transaction, you check whether the update tables VBHDR, VBMOD, and VBDATA are empty. If they are not, process all of the outstanding updates with SAP transaction SM13. You cannot continue with the subsequent steps until all of the outstanding updates have been processed.

2. Import the transport KDOK000668 into your system if you have not done so yet. Check whether the VB protocol tables have been isolated in their own tablespaces.

   For more information, see Assigning the VB Protocol Tables to Dedicated Buffer Pools [Page 117].

3. Partition the VB protocol tables

   Partitioning is actively supported by transaction SE14 but since VBHDR, VBMOD, and VBDATA are multiplex tables, they cannot be converted. Consequently, you have to proceed as follows:


   b. Choose Delete database table.

   c. Choose Storage parameters.

   d. Specify and save the storage parameters for the partitioning of the table. As a partitioning index, use VBDATA's primary key (index ID =’0’). This primary key encompasses the IP address of the application server.

      Note that the IP address (in hexadecimal code) of the application server (and not its name) is written to the primary key field VBKEY.

      There are four application servers for the dialog processes.

      Their IP addresses are:

      155.56.94.121 (hex 9B.38.5E.79)
      155.56.94.121 (hex 9B.38.5E.7A)
      155.56.94.123 (hex 9B.38.5E.7B)
      155.56.94.124 (hex 9B.38.5E.7C)

      The system number is 11.
This results in the following mapping between the application server and the update key:

155.56.94.121 -> ‘9B385E7911...
155.56.94.121 -> ‘9B385E7A11...
155.56.94.123 -> ‘9B385E7B11...
155.56.94.124 -> ‘9B385E7C11...

Accordingly, you should use the following attributes for the partitioning:

PART 1 VALUES (‘9B385E7911’)
PART 2 VALUES (‘9B385E7A11’)
PART 3 VALUES (‘9B385E7B11’)
PART 4 VALUES (X’FF’)

Also, specify buffer pool BP4 for each VB tablespace and index.

On the initial screen of transaction SE14, select Create database table.

For more information, see Partitioning VB Protocol Tables [Page 119].

4. Set the buffer pool parameters
   ALTER BUFFERPOOL(BP4) VPSIZE(1000) VPSEQT(10) DWQT(70) VDWQT (50)

5. Change the SAP profile
   dynp/trans_id_format=2

6. The following recommendations apply to data sharing environments only:

7. Establish proper affinity between VB* table partitions and application servers. For a set of application processes that are connected to a DB2 data sharing member, define the corresponding update processes on the application servers that are also connected to the very same data sharing member. The easiest way to achieve this is to disable update log on dispatch balancing and define the update processes at the same application server where the corresponding dialog processes are defined.

The related profile parameters should be set as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdisp/vb_dispatching</td>
<td>0</td>
</tr>
<tr>
<td>rdisp/vbIncluded_server</td>
<td>no entry (for example, blank)</td>
</tr>
<tr>
<td>rdisp/vbname</td>
<td>name of the local application server; also possible replacement variable: $(rdisp/mynname)</td>
</tr>
</tbody>
</table>

⚠️ This method could create a bottleneck if the update work processes defined at an application server cannot service the load generated at that application server or, in the worst case, if that application server goes down. If you need to address this, use the method (called multiplexing) described in SAP Note 109515.
30 Clustering Index

Definition

Bad transaction response times can be caused by excessive I/O for specific statements. When a row is inserted in a table, DB2 tries to place it near the rows that have similar key value for the index known as the clustering index. Placing the rows in this manner greatly improves subsequent retrievals of ranges of rows that are accessed using the clustering index. Namely, the rows can be read with fewer I/O operations: the pages read are likely to contain more rows that need to be retrieved. In addition, DB2 can maximize effects of its sequential prefetch feature.

Problem

Obviously, there can be only one clustering index. When the SAP system defines tables and indexes, it is not known which of them should be defined as clustering because the usage and the table access are mostly customer specific. Therefore, the SAP system makes an arbitrary choice and specifies that the primary index (index 0) is clustering. In most cases, this proves to be optimal choice.

Solution

It is possible that a table is accessed index-sequentially and mostly via an index that is not clustering. That results in less than optimal response times for these statements. The index-sequential access occurs in most cases for the range predicates (BETWEEN, >, <), predicates that include only a prefix of an index or for accesses through a non-unique index. You can identify these statements in the ST04’s Cached Statements Statistics Panel output. If they have relatively large average response times and these times are caused by I/O suspensions, you should consider the following recommendations.

- If there are statements that access a table index-sequentially through an index that is not defined as clustering and this is a predominant way that table is accessed, the performance can be improved by specifying that particular index as clustering. To change the clustering index, issue ALTER INDEX with option CLUSTER on the index that you want to specify as clustering. This implicitly drops the clustering attribute of the existing clustering index. New rows are immediately inserted according to the new clustering index. Existing data remains clustered by the previous clustering index until you reorganize the corresponding tablespace.
- This process is not done using the ABAP Dictionary but natively within DB2.
- Be aware that a transport or SAP release upgrade that would change the structure of the table in such a way that the table needs to be recreated would reinstate the primary index as the clustering index. In this case, you would need to repeat the process of changing the clustering index. This does not apply to partitioned tablespaces for which all the attributes including clustering are preserved.

The bottom line is that you should consider this tuning step only if you understand the considerations listed above and there are statements that would be significantly improved through tablespace reclustering.
31 Partitioning Key

Definition

With range partitioning, partitioning keys define how table rows are distributed among the partitions of a partitioned table space. As of DB2 V8, the partitioning key does not have to be tied to any index.

Since the use of tables in SAP applications can be very different from company to company, the sizes of tables and optimal partitioning keys are not generally known at installation time.

Alternatives to Partitioning

Consider the following points if you want to partition a table because of its size. They may allow you to reduce the size of the table so that partitioning is no longer necessary.

- The table may become unnecessarily large as a result of improper customization. SAP Note 48009 describes some of these cases. Check to see if they may apply to you.
- Archiving old table data allows you to limit table growth. Archiving also improves overall performance due to the smaller table size, utilities are accelerated and migrations and upgrades take less time.
- Check the compression ratio. SAP data generally compresses well. You can find additional information in Data Compression [Page 322].

Use

Below we describe a procedure for partitioning if you only want to partition because your tables are reaching the size restrictions.

Two criteria should govern the way you partition a table:

- The partitions should be of similar size
- When using DB2 data sharing: Ideally, there should be an affinity between the partitions and the DB2 members.

   The partitioning key of a table does not have to coincide with the clustering index of the table.

Partition Size

The partitioning key should be selected so that the partitions are approximately the same size. This usually optimizes performance and availability. Try to prevent some partitions increasing in size much faster than others as this may bring you close to the maximum partition size. The DSSIZE parameter in the CREATE TABLESPACE statement defines the maximum partition size. As DSSIZE and the page size of a table space affect the maximum number of partitions, you should aim to choose a value for DSSIZE that supports the number of required partitions. For example, table spaces with a page size of 4 KB and DSSIZE = 4 GB support up to 4096 partitions. Table space with a page size of 4 KB and DSSIZE = 64 GB support only 256 partitions. For more information, see the IBM documentation DB2 UDB for z/OS SQL Reference.

If, at a later point in time, you notice that the selected key ranges are not optimal, an ALTER TABLE statement and subsequent REORG can be used to change the key ranges. The ALTER TABLE statement also allows you to dynamically add partitions.
Affinity Between Partition and DB2 Member

With DB2 data sharing, all DB2 subsystems that together form a data sharing group can have read and write access to the same tables. To synchronize parallel access to data, DB2 uses P-Locks (physical locks). The overhead for these locks is usually low.

Partitioning tables provide additional options for fine-tuning. If a partition is mainly processed by a single DB2 member, the locking volume of P-Locks is further reduced.

If the application allows for it, it is therefore desirable to establish an affinity between the partitions of a table and the members of the data sharing group.

Batch Scheduling Group

To also ensure the affinity of DB2 members and partitions with SAP background programs and anonymized application servers (logon groups), you can define so-called batch scheduling groups. Numerous application servers of an SAP system can be grouped in transaction SM61. When you schedule batch jobs in transaction SM36, these groups can then be used as target servers. As a result, the batch job is executed on an application server of the specified batch scheduling group.

If all application servers that are connected to the same DB2 member are now grouped to form a batch scheduling group, and if a batch job only accesses certain partitions, the affinity between the DB2 member and the partition is assured.

Partition Due to Size Problems

The following method describes an approach for the partitioning of tables if the only motivation to partition a table is to avoid hitting the maximum size of unpartitioned tables.

The number of partitions must be determined in accordance with the anticipated table growth. The more a table increases in size, the more partitions it should define. You should select 20 partitions as the default with a value of 64 GB for DSSIZE.

To distribute the records of the table evenly on the partitions, appropriate key ranges are required. You can determine these in transaction SE14 with the limit key function. The limit key function returns those key values that distribute the current table entries evenly onto the partitions for a table, an index of this table and the desired number of partitions. Partitioning Tables [Page 310] describes this function in more detail.

If a proposed partition increases disproportionately, divide them into two or several partitions.

The size of table TBSP is 61 GB. It continues to grow and cannot be reduced in size by any of the methods mentioned at the beginning. It has the following columns:

- **DATUM** (key field)
- **UZEIT** (key field)
- **AHOST** (key field)
- **FLIST**

The partitioning key is defined on column AHOST, because it enables to distribute the table contents evenly.

The number of partitions is 20. Each partition would then initially be about 3 GB large, which gives enough room for further growth. The limit key data that define the different partitions needs to be determined then. Be aware that there may be hotspot areas in the table, in other words, areas that grow faster than other value ranges.
Partitioning Financial Tables

In specific financial tables rows of different years are physically sorted by BELNR in financial tables like BKPF or BSIS. The common trait of these tables is that their BELNR columns are followed by GJAHR in the primary key. Since BELNR is only unique within each year, rows of different years with the same BELNR value are physically located next to each other in the table. This is not ideal because these financial tables are predominantly accessed by year.

As of DB2 V8, table clustering and table partitioning are separate processes.

Financial applications, for example, can take advantage of this by partitioning financial tables by column GJAHR. The primary index would remain the clustering index. This would mean that each partition would contain the postings of one year. Within each partition, the rows are clustered according to the primary index. This perfectly suits queries that access the postings of a given year only, which should be the usual case.
32 Locking Considerations

The following points need to be observed to avoid locking problems such as long suspensions, timeouts, and deadlocks.

Parameter Settings

Ensure that the following recommendations are observed for the locking-related parameters and options:

- **System parameters**
  - **NUMLKUS** = at least 500000
  - **DEADLOK=5,1**
  - **IRLMRWT=600**
  - **XLKUPDLT=YES**
  - **RELCURHL=YES**
  - **EVALUNC=YES**

- **Bind options**
  - **ISOLATION=UR**
  - **Application server on z/OS:** CURRENTDATA=YES
  - Otherwise: BLOCKING UNAMBIG
  - **RELEASE(COMMIT)**

- **DDL options**
  - **LOCKSIZE=ROW**
  - **LOCKMAX=1000000** (for NUMLKUS > 1000000, otherwise adjust it accordingly)

- **IRLM private address space**
  - The IRLM startup procedure parameter MLMT on DB2 installation panel DSNTIPJ, which specifies the maximum amount of private storage that the IRLM uses for its locks should be at least 4 GB. If you find out that this value is too low, increase it adequately. In any case, ensure that the size does not exceed the amount of real storage available to IRLM private address space. Be aware that the maximum number of locks is 100 mio. in DB2 V8.

Lock Escalations

Applications that perform massive updates and deletes and do not commit frequently can potentially accumulate a large number of locks. This regularly has a negative effect on the overall system performance and throughput. The reasons are multifold:

- Increased lock waits due to contentions with other concurrently running transactions and programs, which can result in time-outs and deadlocks.
- Increased paging.
- Increased CPU consumption
- Potentially severe availability exposure in case of an extremely high number of locks requested and held:
  - The number of locks that IRLM (the DB2 locks manager) can manage concurrently is limited by 90% of the total storage given to the IRLM private address space. Each lock requires about 540 bytes. For example, if the size of the IRLM private address space is 4 GB, IRLM can hold up to approximately
7900000 locks. Once this limit is reached, the report or transaction that requests additional locks will be abnormally terminated with the ‘resource unavailable’ symptom (SQL return code –904). In many cases this is the very application process that accumulated all these locks and performed a large number of updates that now need to be backed out, for example, the application must do a rollback. Consequently, the back out process takes very long (much longer than the time the application spent until it abended). During this long time, the affected resources (including the IRLM itself!) cannot be accessed by other transactions and reports.

- In data sharing environments the number of locks held concurrently is additionally limited by the size of the Coupling Facility (CF) lock structure which must be resident in the central storage. Before CF level 12, this is limited to 2 GB. Therefore, it is highly beneficial to apply CF level 12, which eliminates this limitation and introduces 64-bit support. Furthermore, the storage allocated to the lock structure is shared by the Lock Hash Table and the Record List Entries (RLE) sections. The initial split in terms of relative storage allocation is either explicitly specified by means of the parameter IRLM HASH (introduced by RLM 2.1 APAR PQ44114) or controlled by an internal IRLM algorithm. The maximum number of modify locks related to database update processing that can be concurrently held and propagated to the CF is limited by the size of the RLE section. The actual number of modify locks that can be held in a given RLE section size is dependent on the CF Level and the level of z/OS (for more details, see: www.ibm.com/servers/eserver/zseries/ps0/cftable.html). For example, if the CF lock structure size is 256 MB, the number of propagated locks that can be concurrently held might be around 1650000. Once such limit is reached (or, more precisely, 90 % of the limit values), new modify lock requests will be rejected and application processes will abnormally terminate with –904. Like an IRLM storage exhaustion, long and very disruptive backouts would follow.

- If the DB2 subsystem abnormally terminates for any reason, the restart would take very long time if a large number of locks were held at the time of the abend. Furthermore, in data sharing environments, where a single subsystem failure does not result in an outage, the large number of modified locks held, (for example, propagated to the CF), would affect the entire sysplex. Namely, it is possible that the surviving members do not have the storage capacity to hold retained locks for the failed member.

The exposure of encountering these problems is likely to increase in time because the growth of the database will result in mass updating, non-committing transactions and reports requesting and holding ever more locks.

The problems can be prevented by 'lock limit aware' application coding and by using DB2 means to reduce the number of locks held concurrently. Most SAP delivered programs are written in such a way. The problems are more likely to be encountered with user-written programs, which need to be reviewed to ensure an appropriate commit frequency.

In addition, DB2 also provides mechanisms for limiting the number of locks concurrently held: the system parameters NUMLKUS and NUMLKTTS, and the tablespace attribute LOCKMAX. The system parameter NUMLKUS sets a limit on the number of locks any individual DB2 thread can hold. Once this limit is reached, the program that accumulated these locks will terminate with SQL code –904. The maximum value for NUMLKUS is 100 million and it is recommended to use 2097152 as an initial, first-cut value. Setting a lower value for NUMLKUS helps you to detect offending programs earlier and is especially recommended for test systems. In most production systems (except the Retail component), a lower value for NUMLKUS is acceptable, but it should not be lower than 500000.

The system parameter NUMLKTTS sets the default for the tablespace attribute LOCKMAX. If the number of locks on a particular tablespace exceeds the LOCKMAX value, these locks will be replaced by a single tablespace-scope lock.
**LOCKMAX** is enforced on a per thread, per tablespace basis.

This process is called lock escalation and it can eliminate the occurrence of IRLM and CF lock structure exhaustion.

On the other hand, lock escalations do not necessarily address all lock contention problems. They can lead to deadlocks, but it is very likely that the deadlock victim (the process that needs to backout) is a process for which the backout is least expensive.

This addresses the issue of long backouts caused by IRLM or CF lock structure exhaustion. Another challenging aspect of lock escalations is coming up with an optimal value for **LOCKMAX**. This is very customer-specific and depends on the particular workload and the available central storage resources. If set too high, it allows the transactions and reports to hold too many locks, which leads to the problems described earlier. If set too low, it causes lock escalations too frequently, with the resulting negative consequences.

For example, if **LOCKMAX** is set too low, it might happen that an application process that has acquired more than **LOCKMAX** locks triggers a lock escalation that in turn cannot be successfully executed because other applications are holding non-compatible locks.

After the timeout period, the escalation triggering application process would receive a –913 (timeout) and would need to rollback. Had the **LOCKMAX** value been higher, the application might have completed without lock escalation.

With **NUMLKUS** set to 2097152, we recommend using an initial, first-cut value of 1000000 for **LOCKMAX**. This value can be adjusted by the customers depending on the above described site-specific considerations and the **NUMLKUS** setting. As said earlier, holding a large number of locks is not recommended. The lock escalation mechanism helps avoid serious consequences, but it is still the best approach to identify the transactions and reports that requested and held so many locks and try to change these applications, for example, by inserting commits if the application logic allows it) to reduce the number of locks held. For SAP-written programs, open an SAP problem message, for user-written programs, talk to the application developers.

There are a number of ways to identify these critical transactions and tablespaces:

- Monitor the maximum number of locks that are held by individual DB2 threads. Transaction ST04 provides the **Thread Activity: Thread List** panel where a snapshot of the currently active DB2 threads is reported. One of the columns is 'Max Locks Held', which reports the maximum number of locks held by the thread at any time during the current transaction. Sort the thread list by that column in a descending order and investigate the corresponding work processes for which transactions, reports and tables have been involved in acquiring and holding large number of locks. Transaction SM66 displays which tables are being accessed by which work process.

- Monitor lock escalations. Every time an escalation occurs, DB2 issues IFI record 337 and message DSNI031I is written to the z/OS console. The database alert router part of RFCOSC0L catches IFI record 337 and provides this alert in transaction ST04. The alert identifies the SQL statement, the originating ABAP report and transaction code and the tablespace for which escalation occurred.

- Monitor fields Executions, Getpages, Rows Processed and Rows Examined by the individual update, delete and insert statements reported in the ST04 Cached Statements Statistics panels. This could help you determine which tablespaces need special attention.

- Monitor the number of changes per table in transaction ST10. Again it serves as a hint on where to go next.

- The long-running, non-committing units of recovery do not necessarily hold a large number of locks, but are definitely worth further investigation. They can be detected by using the system parameters **UR CHECK FREQ** on panel **DSNTIPB** (**ZPARM URCHKTH**) and **UR LOG WRITE CHECK** on panel **DSNTIPB** (**ZPARM URLGWTH**). If any of these
thresholds is reached, DB2 issues a warning message to the console and IFI record 313.

The SAP CCMS Monitor Set (transaction RZ20) and transaction ST04 exploit IFI record 313, which allows you to identify long-running transactions. For more information, see section 5.2.9 DB Alert Router.

Apart from enabling lock escalation you should ensure that the IRLM and CF are sized to maximize the number of locks that can be concurrently held.

In data sharing environments, you additionally need to consider the coupling facility lock structure size. The CF lock structure is split between the Lock Hash Table and Record List Entries sections where only the latter is relevant in terms of the number of modify locks that can be concurrently held. The split can be explicitly specified by means of the IRLM parameter HASH and INITSIZE value for the LOCK1 structure in the CFRM policy.

The difference between INITSIZE and HASH is the storage allocated for the modified locks, for example, the RLE section. Once the RLE section size is determined, you should adjust the lock escalation trigger parameter (LOCKMAX) for your tablespaces. You can calculate the maximum LOCKMAX value that makes sense for given INITSIZE and HASH values based on the fact that a single lock entry in the RLE takes approximately 160 bytes, but it will vary based on CF Level and level z/OS. For example, if you set INITSIZE to 384 MB and HASH to 128, the RLE section size will be 256 MB, and it can hold approximately 1650000 lock entries (rounded down for safety). Therefore, setting the LOCKMAX value higher than that would not shield you from exhausting the structure even by a single thread accessing a single tablespace.

The actual INITSIZE and HASH values you will use depend on the amount of available central storage on the CF allowing for structure failover from the alternate CF.

Note that the entire lock structure must be backed by the central storage. Of course, keeping the number of false contentions small (directly affected by the size of the Hash Table) should be another objective, but that is beyond the scope of this text.

Identify Long-Running Read-Only Reports

As we have seen in the previous section, not committing regularly in the update transactions and reports has a negative impact on the overall system. However, application programmers often do not consider that even read-only transactions and reports need to be committed regularly in order to improve the overall concurrency in the system. This is because some locks are acquired within read-only processes as well and these locks are released only at commits.

For example, pageset intent locks and so-called claims are acquired at different phases of statements prepare and execute time. In general, they do not affect concurrent DML statements. However, the concurrent DDL statements, some DB2 commands, and utilities can be negatively affected. Also, some tables (cluster tables and pool tables) are read using the isolation levels that acquire shared locks. The shared locks are not compatible with concurrent updaters and can result in lock contentions including deadlocks and timeouts.

Typical problems that can be expected in an environment with long-running read-only transactions and reports that do not commit regularly are:

- Lock suspensions during the switch phase of the online REORG.
  This is because the REORG waits for read-only transactions/reports to commit in order to move them over the reorganized data. During the wait (that can end up in a timeout), other transactions are queuing up behind the REORG, which results in a system performance degradation.
  The options RETRY, RETRY_DELAY, DRAIN_WAIT, and FASTWITCH significantly improve REORGs concurrency with other transactions.
- STOP DATABASE failures.
These commands are sometimes used to quiesce access to a particular database or
tablespace, for example, while re-assigning a tablespace to a different buffer pool.

- Timeouts and deadlocks.

Regular commits (recommended frequency is approximately one per minute) need to be
included in both the long-running read-only and update transactions and reports. If a
particular cursor needs to be open for a long time, consider using the WITH HOLD option in
order to preserve the cursor position across commits. This will not release the claim on the
tablespace referenced in the cursor, for example, online REORG on that particular tablespace
still needs to wait in the SWITCH phase, but it will make all other objects available in that unit
of recovery.

To identify long-running read-only reports, use the Long-running transactions alert monitor in
SAP CCMS Monitor Set (transaction RZ20) or transaction ST04. The DB2 system parameter
LONG-RUNNING READER on panel DSNTYPE (ZPARM LRDRTHLD) controls the threshold on
the duration of read-only transactions that makes DB2 consider them long-running readers. If
read-only transactions run longer than this threshold, DB2 issues IFI record 313. The
information provided includes the name of the offending ABAP report. Such reports are most
often found in user-written transactions and batch reports. The best practice is to change
these reports by inserting commits if the application logic allows it. For SAP written programs,
open an SAP problem message, for user-written programs, talk to the application developers.

**Various Considerations**

Avoid lock contention caused by inappropriate access paths. See “Ensuring optimal access
paths” and “Optimizing access to VBHDR, VBMOD, and VBDATA” for recommendations on
how to achieve the objective. Avoid application server buffering for the tables with frequent
updates. That not only adds to the cost of buffer resynchronizations, but can cause heavy
lock contentions.

Call SAP transaction ST04 for monitoring locking events including snapshots of current
locking conflicts across the system. See the SAP CCMS Monitor Set (transaction RZ20) or
transaction ST04 to identify timeouts and deadlocks. For more information, see Monitoring
and Performance [Page 139].
33 Buffer Pool Tuning Considerations

Buffer pools are among the most important objects in DB2 performance monitoring and tuning. After the installation of an SAP system, all of the tablespaces are backed by some predefined buffer pools. The recommended initial buffer pool parameter settings are sufficient for the installation and functional verification of the product, but they are not optimal in terms of performance.

This guide contains detailed steps you should perform to tune DB2 buffer pools for optimal performance. For more information, see Buffer Pool Tuning [Page 182] under Monitoring and Performance -> Database Tools.
34 Dynamic Statement Caching Considerations

Caching of the dynamic SQL statements is the key ingredient of a well-performing SAP system component. The following parameters need to be set to optimize performance of this feature:

- **System parameter CACHEDYN=YES** (this is the default value).

- **Sufficiently large EDM Statement Cache**
  
  The DB2 system parameter EDMSTMTC controls the size of the EDM Statement Cache. We recommend that you set it to 300 MB. You need to monitor the global cache hit ratio (reported in DB2 Subsystem Activity monitor, transaction ST04). If it is consistently lower than 95%, increase EDMSTMTC in increments of 20 MB. Do not oversize it because it results in overcommitting of the real storage and increased system paging. If you connect to DB2 via DRDA and locally from application servers on z/OS, make sure that all of them use the same user ID. Namely, the identical SQL statement that is prepared by two different user IDs results in two cache entries instead of one.

- **System parameter MAXKEEPD**
  
  This parameter specifies how many prepared statements are kept across all of the threads. We recommend that you start with 8000.

- **System parameters**
  
  Make sure that the delivered packages and plans that are used by the SAP system are bound with KEEP_DYNAMIC (YES).
35 Transaction-Based DB2 Accounting and Workload Management

The SAP application server takes advantage of DB2’s capability to set client identifiers (for example, workstation name and transaction name) to associate units of work that are processed within DB2 with the corresponding units of work within the application server. It provides very detailed information on both static and dynamic properties of the SAP system.

**Static properties** do not change during the lifetime of SAP workprocess and DB2 threads. They include the SAP system ID and the work process ID.

**Dynamic properties** of work processes can change at any time. An example of a dynamic property is the ABAP report name.

Every DB2 transaction that is used to execute a specific SAP transaction is tagged with DB2 client identifiers that are filled with information about that transaction. These identifiers contain details of the SAP system that executes the report and information on the report itself, including the SAP end user ID. As the DB2 client identifiers are the base for the accounting information that DB2 generates and for managing the DB2 workload by means of WLM, accounting data can be analyzed and workload management can be performed at a very fine level.

To let the SAP application server pass the transaction level properties to DB2, set the SAP profile parameter `dbs/db2/use_accounting = 1`. For the SAP application server on z/OS set the SAP profile parameter `dbs/db2/use_wlm = 1` to enable transaction level workload management.

This section describes the attributes that the SAP application server passes to DB2, which allows you to do DB2 accounting and workload management at the level of individual SAP transactions, ABAP reports, batch jobs and end users. This enables IT departments to charge back the costs that a specific department generated in DB2 to this department. Also, by means of WLM, a higher priority can be assigned to a crucial SAP transaction or batch job for example.
35.1 DB2 Client Identifiers and WLM Qualifiers

The term DB2 client identifier refers to the IFI fields within DB2 that can be set by clients. These fields are tagged to DB2 threads and transactions. Each client identifier is contained in an IFI field.

The following table summarizes the client identifiers that are relevant to SAP and the corresponding IFI field names and WLM qualifiers. To carry out workload management based on a specific SAP attribute, you must know the DB2 client identifier that is employed by this SAP attribute. See the sections Static Properties Passed to DB2 [Page 134] and Dynamic SAP Properties Passed to DB2 [Page 135] for more information.

When you have this information, you can define WLM classification rules for the WLM qualifier that is used by this DB2 client identifier.

<table>
<thead>
<tr>
<th>DB2 Client Identifier</th>
<th>IFI Field</th>
<th>WLM Qualifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation ID</td>
<td>QWHCCV</td>
<td>CI</td>
</tr>
<tr>
<td>Accounting string suffix</td>
<td>QMDASUFX</td>
<td>AI (positions 56 – 143)</td>
</tr>
<tr>
<td>Primary authorization ID</td>
<td>QWHCAID</td>
<td>UI</td>
</tr>
<tr>
<td>Plan name</td>
<td>QWHCPLAN</td>
<td>PN</td>
</tr>
<tr>
<td>Transaction name</td>
<td>QWHCEUTX</td>
<td>PC</td>
</tr>
<tr>
<td>Workstation name</td>
<td>QWHCEUWN</td>
<td>SPM (positions 17 – 34)</td>
</tr>
<tr>
<td>Client user ID</td>
<td>QWHCEUID</td>
<td>SPM (positions 1 - 16)</td>
</tr>
</tbody>
</table>
35.2 Static Properties Passed to DB2

The following table show the DB2 client identifiers to which the static SAP attributes are assigned. The work processes of the SAP application server pass these attributes to DB2 when they establish connections to DB2.

Note that some of the DB2 client identifiers are shared by multiple SAP ABAP attributes.

The last column shows the positions occupied by an SAP attribute within a specific client identifier.

<table>
<thead>
<tr>
<th>SAP Attribute</th>
<th>DB2 Client Identifier</th>
<th>Positions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP system ID (SAPSID)</td>
<td>Correlation ID</td>
<td>1 – 3</td>
</tr>
<tr>
<td>Work process type</td>
<td>Correlation ID</td>
<td>4 – 6</td>
</tr>
<tr>
<td>Work process number</td>
<td>Correlation ID</td>
<td>7 – 9</td>
</tr>
<tr>
<td>Host name of application server</td>
<td>Accounting string suffix</td>
<td>1 – 32</td>
</tr>
<tr>
<td>SAP system number (SAPSYSTEM)</td>
<td>Accounting string suffix</td>
<td>33 – 34</td>
</tr>
<tr>
<td>Work process ID</td>
<td>Accounting string suffix</td>
<td>35 – 44</td>
</tr>
<tr>
<td>Database connection name</td>
<td>Accounting string suffix</td>
<td>45 – 74</td>
</tr>
<tr>
<td>User ID used to connect to DB2</td>
<td>Primary authorization ID</td>
<td>All (1 – 8)</td>
</tr>
<tr>
<td>Plan name of SAP</td>
<td>Plan name</td>
<td>All (1 – 8)</td>
</tr>
</tbody>
</table>

The following attributes are passed by SAP Java applications:

<table>
<thead>
<tr>
<th>SAP Java Attribute</th>
<th>DB2 Client Identifier</th>
<th>Positions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP system ID (SAPSID)</td>
<td>Primary authorization ID</td>
<td>4 - 6</td>
</tr>
<tr>
<td>Host name of application server</td>
<td>Workstation name</td>
<td>All (1 - 18)</td>
</tr>
<tr>
<td>Indicator for Java workload (constant db2jcc)</td>
<td>Correlation ID</td>
<td>1 - 6</td>
</tr>
</tbody>
</table>
35.3 Dynamic SAP Properties Passed to DB2

Dynamic SAP properties are generally passed to DB2 at DB2 transaction boundaries. When a new DB2 transaction commences, or in other words, when the first SQL statement after COMMIT WORK or ROLLBACK WORK is submitted, the SAP application server passes the corresponding SAP properties to DB2. Since the dynamic SAP properties may change during a single DB2 transaction, changed properties are also passed to DB2 at the end of database transactions. The rationale is that the dynamic SAP properties that are valid at the end of transactions are generally more representative and meaningful. DB2 takes the changed properties for accounting into consideration. Due to its nature, workload management is always based on the properties that are set at transaction begin. For SAP batch jobs, the dynamic SAP properties are set at job begin and do not change during their execution. The dynamic SAP properties are listed in the table below.

<table>
<thead>
<tr>
<th>SAP Attribute</th>
<th>DB2 Client Identifier</th>
<th>Positions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABAP report name</td>
<td>Transaction name</td>
<td>All (1 – 32)</td>
</tr>
<tr>
<td>BW infoprovider ‘’ BW report name</td>
<td>Transaction name</td>
<td>All (1 – 32)</td>
</tr>
<tr>
<td>SAP transaction code</td>
<td>Workstation name</td>
<td>All (1 – 18)</td>
</tr>
<tr>
<td>SAP batch job name</td>
<td>Workstation name</td>
<td>All (1 – 18)</td>
</tr>
<tr>
<td>SAP end user</td>
<td>Client user ID</td>
<td>1 – 12</td>
</tr>
</tbody>
</table>

Note that the SAP transaction code and the SAP batch job name have the same client identifier. However, SAP passes the batch job name only during batch work processes. For all other work process types, SAP passes the transaction code onto DB2. The ABAP report name and the concatenation of BW infoprovider and BW report name also share the same client identifier (Transaction name). The BW infoprovider is passed instead of the ABAP report name during the execution of OLAP queries only.

In installations with SAP Web AS Version 6.10 that utilize the downward compatible kernel of release 6.40 to connect to DB2 V8, the batch job name is not passed to DB2. Moreover, in this scenario, the first database transaction of batch jobs always starts in the batch infrastructure report RSBRCTRE. For installations of SAP Web AS Version 6.20, you can avoid these problems by applying SAP basis support package SAPKB62048.

DB2 generates the transaction-level accounting data and changes WLM enclaves at transaction boundaries only if the DB2 system parameter CMTSTAT is set to INACTIVE, which is the default value. It is recommended that you keep the default value. As the SAP application server utilizes the Bind option KEEPDYNAMIC (YES), the DB2 threads that serve SAP nevertheless remain active all the time. This behavior is desired as it optimizes performance.

Dynamic properties are only available for ABAP workload.

ABAP Report Name Passed to DB2

During the execution of an ABAP report, the report might call function modules, submit other reports or call transactions that submit other reports as well. Therefore, it is important to know which ABAP report name is being passed to DB2.
For all work processes (except for batch work processes), SAP passes the ‘main’ report name to DB2, which only changes by issuing the ABAP statements submit <report> or call transaction or call dialog. Local function calls do not affect the main report name, which results in more predictable names that are passed to DB2.

There are exceptions to this. If the main report is a generic ABAP infrastructure report that starts with SAPMSSY, then the application server inspects the call stack and passes the first ABAP report name to DB2 that does not have this prefix. Since the maximum length of ABAP report names is 40 and the DB2 client identifier contains only 32 bytes, the name that is passed to DB2 may be truncated.

The SAP BW component accesses infoprovider data always by the same ABAP report. Hence, the ABAP report name does not contain appropriate information to classify workload. It is therefore not passed to DB2. Instead, the name of the infoprovider and, if available, the BW report name is passed to the client identifier transaction name. Infoprovider and BW report name are concatenated by a slash (/).

For function modules that are called via RFC, the application server uses the same algorithm to determine the ABAP report name to be propagated to DB2 as for any other dialog work. In other words, it passes the report name of that function group to DB2 to which the function that is called via RFC belongs.

The report names of function groups always start with SAPL. For example, if the function SAMPLE_FUNC is part of the function group SAMPLE_GROUP, then the application server would pass SAPLSAMPLE_GROUP as ABAP report name to DB2.

During batch job execution, SAP propagates the name of the ABAP report to DB2 that is specified in the definition of the batch job. This makes analyzing accounting records and workload management for batch jobs much easier.

Note that all work in update work processes is performed by the ABAP report RSM13000.

**SAP Transaction Code Passed to DB2**

The SAP application server passes the transaction code to DB2 that is valid when submitting the first SQL statement of a DB2 transaction. If the transaction code changes during the DB2 transaction, the changed transaction code is passed to DB2 just before commit or rollback. The DB2 accounting information then contains the updated transaction code.

As function modules that are invoked via RFC do not have an associated SAP transaction code associated with them on the local system, the SAP application server does not pass a SAP transaction code to DB2 for them.

For update work processes, the transaction code of the dialog transaction that originated an update task is not known at the start of the update task. Therefore, workload management based on the SAP transaction code is not possible for update work processes.

For OLAP queries from SAP BW that are executed in dialog processes, the string BI_REPORT is passed to the DB2 instead of the SAP transaction code. This allows you to identify workload that is generated by online reporting.

**SAP End User Passed to DB2**

The SAP application server passes the SAP end user ID to DB2 that is valid when submitting the first SQL statement of a DB2 transaction. If the end user ID changes during the DB2 transaction, the changed end user ID is passed to DB2 just before commit or rollback. The DB2 accounting information then contains the updated end user ID.

For batch jobs, the behavior is slightly different. Analog to the ABAP report name, the end user ID that is passed for batch jobs is fixed for the duration of the job. SAP propagates the user ID name to DB2 that is specified in the definition of the batch job.
**Aspects of Accounting and Workload Management at Transaction Boundaries**

Some conditions may prevent accounting data and workload management at transaction boundaries. In DB2 V8 held cursors, held LOB locators and active declared global temporary tables prevent accounting intervals and WLM enclaves to complete at the end of transactions. You should take this into consideration in user-written ABAP reports.

To automatically drop declared global temporary tables at commit, you may consider using the clause `ON COMMIT DROP TABLE` during their declaration. For more information, see the IBM documentation *DB2 for z/OS V8: Administration Guide*.

For RFC work, make sure that RFC destinations are defined completely, in other words, include user ID and password. Otherwise, the first transaction in a function that is called via RFC is attributed to generic identifiers.
35.4 Analysis of Accounting Records

To analyze the accounting records that DB2 generates, you can use a tool like DB2 Performance Expert (PE) for z/OS. Such a tool allows you to filter out and view individual records and to aggregate accounting records with respect to DB2 client identifiers. With this information, you can charge back costs to individual users or departments.

The structure that DB2 PE offers to define groups of users or values is the set name. Another feature of DB2 PE concerns the correlation ID field. DB2 PE allows you to divide this identifier (which contains many different SAP attributes) into the subfields correlation name and correlation number. DB2 PE treats these fields like it would any other identifiers. The correlation ID translation can be modified as well.
Monitoring and Performance
36 Setup of Database Performance Monitoring with rfcoscol

DB2 collects a broad range of performance indicators on a wide variety of aspects. It makes them available via its Instrumentation Facility Interface. The performance data that is most relevant to SAP systems is provided by the SAP integrated Database Performance Monitor, which is transaction ST04. This monitor uses rfcoscol to collect DB2 performance data via the Instrumentation Facility Interface. With regard to performance monitoring, rfcoscol serves the following purposes:

- Provides current snapshot information of the overall system
- Allows you to submit DB2 commands
- Catches database alerts that are raised by DB2 and propagates them to the SAP system

To ensure that these features function properly, it is important that there is always an instance of rfcoscol running. This is essential for proper performance monitoring.

For details on DB2 trace classes and the IFI interface, see the SAP documentation SAP on IBM DB2 UDB for z/OS: Database Administration Guide.

To be able to monitor database performance, the rfcoscol needs to be started. You can do this from within the SAP system. Prior to starting rfcoscol, you can perform the following configuration steps if the default settings do not match your requirements:

- Specify the configuration parameters in DB2
- Specify the necessary parameters to run rfcoscol
- Create the appropriate RFC connection in SM59
36.1 Installation of rfcoscol

You can install rfcoscol and the corresponding database request module for z/OS using SAPinst. For the application server on z/OS, the rfcoscol is supplied as part of the kernel, and is started automatically when you start the SAP system.

If you have chosen not to install rfcoscol and the database request module for z/OS in the recommended way using SAPinst, see SAP Note 426863 and Manual Installation of rfcoscol on z/OS [Page 142].

It is recommended that you use the rfcoscol that is provided with the installation of saposcol. In this case, you share one rfcoscol installation between saposcol and database performance monitoring.

Depending on your requirements or due to maintenance reasons, you might want to use one rfcoscol level for different SAP system releases. In this case, you can install rfcoscol in a separate directory.

When you install rfcoscol, ensure that the appropriate permission bits are set:

- EXECUTE permission for the rfcoscol for the user ID that starts it
- READ permission for the database request module for the user ID that binds it.
36.2 Manual Installation of rfcoscol on z/OS

This section describes how to install rfcoscol manually on z/OS.

We recommend that you allow SAPinst to install rfcoscol instead of carrying out the installation manually.

Installing rfcoscol executables

All the files needed for rfcoscol operations are contained in the file IBMOS390.PAX in packed form. The pax archive is included on the SAP upgrade or installation master DVD in directory

<DVD_Drive>:/OS390/DBTOOLS

To install the executables:

1. Create a directory <rfcoscol_dir> in z/OS UNIX System Services HFS.
2. Transfer file IBMOS390.PAX to directory <rfcoscol_dir>. To do this, use the FTP command and execute the transfer in binary mode.

   If an FTP transfer directly to HFS is not possible, you must reserve a data set <hlq>.IBMOS390 first (record format FB, size at least 10 MB).
   - Transfer file IBMOS390.PAX with FTP to this data set.
   - Copy the file into HFS using the TSO command:

     oput '<hlq>.IBMOS390' '<rfcoscol_dir>/IBMOS390.PAX' binary

3. Unpack file IBMOS390.PAX in HFS directory <rfcoscol_dir> with the following command:

   pax –rzvf IBMOS390.PAX

Creating Installation-Specific Files

In the HFS directory <rfcoscol_dir> create the text file saprfc.ini with the following content:

DEST=<destination>
TYPE=R
PROGID=<program_ID>
GWHOST=<gateway_host>
GWSERV=<gateway_service>
RFC_TRACE={0|1}

Replace the variables with a suitable entry.

Description of Variables

<table>
<thead>
<tr>
<th>Variable or Value</th>
<th>Description</th>
</tr>
</thead>
</table>

SAP Online Help 20.04.2005
<table>
<thead>
<tr>
<th><strong>&lt;destination&gt;</strong></th>
<th>Name of your choice (case-sensitive)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Recommended value:</strong></td>
<td><strong>&lt;SAPSID&gt;</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>TYPE=R</strong></th>
<th>RFC connection type: TCP/IP</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>&lt;program_ID&gt;</strong></th>
<th>Content of the SAP profile parameter <code>dbs/db2/hosttcp</code> which you choose yourself. The value is case-sensitive.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Recommended value:</strong></td>
<td><code>&lt;db-hostname&gt;.rfcoscol</code></td>
</tr>
<tr>
<td></td>
<td>The value needs to match the program ID of the corresponding RFC destination specified in transaction SM59.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>&lt;gateway_host&gt;</strong></th>
<th>IP address or host on which the SAP gateway is running</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>&lt;gateway_service&gt;</strong></th>
<th>Service or port number of the SAP gateway</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>RFC_TRACE=1</strong></th>
<th>RFC_TRACE=1 or RFC_TRACE=0 determines whether an RFC log (dev&lt;PID&gt;_&lt;TID&gt;.trc file) is written, where &lt;PID&gt; is the process ID and &lt;TID&gt; is the thread ID.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RFC_TRACE=0</strong></td>
<td>This optional parameter should only be switched on when tracing is necessary. To switch tracing on, specify 1. To switch it off, specify 0.</td>
</tr>
<tr>
<td></td>
<td>Be aware that when you switch tracing on during normal work, this can have an impact on system performance and slow down the retrieval of performance data. Choose RFC_TRACE=0 for normal operations.</td>
</tr>
</tbody>
</table>

If you specify a service entry, verify that the corresponding entry is made in `/etc/services`.

One `saprfc.ini` file can contain configurations for several RFC destinations, which means that it can be used as an .ini file by different `rfcoscol` instances.

The individual RFC destination specification blocks are initiated and separated by the `DEST` parameter.
36.3 Configuration of rfcoscol

Generally, there should be a dedicated rfcoscol for each DB2 subsystem and for each SAP system. This type of architecture ensures that there are no dependencies or side-effects between different SAP systems.

However, multiple SAP systems can also share one rfcoscol. This option might be appealing in MCOD landscapes. Moreover, if several DB2 subsystems reside on the same LPAR, they can be monitored by a common rfcoscol. Conversely, multiple rfcoscols can also serve a single SAP system. The advantage of a latter topology is that multiple monitoring requests can be processed in parallel.

To support SYSPLEX failover, you can specify an RFC destination to an rfcoscol for each logical database connection as defined in the connection profile. (For more information, see the SAP installation guides.) This enables the SAP application server to automatically use the rfcoscol that is in charge of the DB2 subsystem to which the application server is currently connected. If no RFC destination is specified for a database connection, the destination SAPOSCOL_<DB_HOSTNAME> is employed, where <DB_HOSTNAME> is the name of the database host. These destinations always exist.

If you plan to use a separate rfcoscol that is not shared with saposcol, you must configure this rfcoscol manually and must also create the RFC destination in SM59.
36.4 Starting rfcoscol Manually

On start, the rfcoscol either reads the saprfc.ini file, which configures the RFC connection in detail, or accepts the required arguments as command line parameters. The rfcoscol start job, which can be submitted from transaction DB2, uses the latter approach.

If you opt for the saprfc.ini file, we recommend that the file reside in the same directory as the rfcoscol. Either create a saprfc.ini or, if one already exists, you can modify that one.

For more information about how to create a saprfc.ini, see Manual Installation of rfcoscol on z/OS [Page 142], "Creating Installation-Specific Files".

Make sure there are no spaces between the parameter identifier and the current value for both the saprfc.ini file and the command line parameters. A sample start command is as follows:

```
./rfcoscol -gissp2n99 -arfcoscol -xsapgw04 &
```

To start rfcoscol properly, the following environment variable must be set:

- **STEPLIB**
  The STEPLIB environment variable must contain a reference to the DB2 load library `<sdsnload_library>`.

The following environment variables are recommended:

- **RFCOSCOL_RETRY**
  If this variable is set, rfcoscol will retry to connect to the gateway if the connection cannot be established.

- **RFC_INI**
  Path to the saprfc.ini configuration file. Default is saprfc.ini in the local directory, which means in the same directory as rfcoscol. Be aware that this environment variable is not required when saprfc.ini is in the local directory or when you start the rfcoscol with the command line parameters.

- **RFC_TRACE_DIR**
  Path of the directory to which rfcoscol will write trace files. Default is the local directory.

The environment variable `<sdsnload_library>` (which is the name of the DB2 load library) can be set either in a start script of the rfcoscol or in the user environment.

If the rfcoscol is started manually or by a system automation tool, you have two options. The first one is to use the following command:

```
nohup rfcoscol -D<destination> &
```

The `-D` option refers to the DEST parameter of the saprfc.ini file. When the rfcoscol is invoked, it searches the saprfc.ini file for a matching DEST parameter. If the parameter is found, it reads all subsequent parameters up to the next DEST parameter, or the EOF marker.
For more details about the rfcoscol parameters, the description of the variables, and the environment variables, see the section “Configuration of rfcoscol”.

Both the necessary environment variables and the rfcoscol start command can be put in a script file, which is used for start up.

Create the file containing the following commands:

```
#!/bin/sh

_EDC_PUTENV_COPY=YES; export _EDC_PUTENV_COPY
_BPX_SHAREAS=NO; export _BPX_SHAREAS
_BPX_SPAWN_SCRIPT=NO; export _BPX_SPAWN_SCRIPT
_BPXK_AUTOCVT=NO; export _BPXK_AUTOCVT
_TAG_REDIR_IN=TXT; export _TAG_REDIR_IN
_TAG_REDIR_OUT=TXT; export _TAG_REDIR_OUT
_TAG_REDIR_ERR=TXT; export _TAG_REDIR_ERR

export LIBPATH=$LIBPATH:<rfcoscol_dir>:/usr/lib
export STEPLIB=<sdsnload_library>
export RFCOSCOL_RETRY=1

cd <rfcoscol_dir>

nohup rfcoscol –D<destination> &
```

Before you execute the start script:

- Change the start script to an executable by using the command:
  chmod a+rx <start_script>

- Ensure that the user ID used to invoke the start script has the EXECUTE permission

If you do not want to use a start script file or the saprfc.ini file, the rfcoscol can be invoked directly as described at the beginning of this section.
36.5 Stopping rfcoscol Manually

Regardless of how you have started the rfcoscol, either by calling the DB2 transaction or by starting it manually, you have to stop it by the appropriate command.

To stop the rfcoscol, issue the kill command in the UNIX System Services shell:

```
kill <pid>
```

If the connection to rfcoscol works, you can also stop it from within transaction ST04.

⚠️ When you issue the `kill` command, not only rfcoscol is closed down, but the associated statistics trace is also stopped. A `kill -9` is not recommended since it forces a termination of the process and does not allow any cleanup work. Only if the normal kill does not work should you use `kill -9` to stop the rfcoscol.
36.6 SAP System Configuration of the RFC Connection

If not otherwise specified in transaction DB2, the DB performance monitor uses the RFC destination `SAPOSCOL_<DB_HOSTNAME>` to communicate with the `rfcoscol`, where `<DB_HOSTNAME>` is the name of the connected database host. Such an RFC destination always exists for every database host.

However, depending on your requirements, you might need to use a different RFC destination.

**Process Flow**

To configure the RFC destination:

1. Log on to the SAP System with a user authorized to call transaction SM59.
2. Call transaction SM59.
3. Choose `Create`.
   
   Create the RFC destination with the following parameters:
   
   - **RFC destination name**
     Enter a freely chosen name in the RFC destination field.
   - **Connection type**
     Enter the connection type `T`, which indicates a TCP/IP connection.
   - **Enter a description.**
   - **Make sure that the Trace option is not set.**
4. Save the configuration.
5. Choose `Registration`.
   
   - **Program ID**
     The program ID specified in SM59 must match the parameter `PROGID` in the `saprfc.ini` file or the `-a` command line parameter respectively.
   - **Activation type**
     The activation type must be `Registered Server Program`.
   - **Gateway options**
     Specify the Gateway host and Gateway service.

   This step is optional. It is only necessary if multiple application servers share an `rfcoscol`.
6. Save your entries.
7. Choose `Test connection` to connect to `rfcoscol`.
To test your newly created RFC destination in transaction SM59, you must start the rfcoscol. For details on how to do this, see the section on Starting rfcoscol Manually [Page 145]. As soon as the rfcoscol is started, the RFC destination should work properly.

SAP system KDF with issp2n99 application server and system number 04. The relevant DB2 subsystem SGF2 is running on the z/OS, ihsapfg.

RFCOSCOL is started with the KDF destination. The relevant entry in the saprfc.ini file is:

DEST=KDF
TYPE=R
PROGID= ihsapfg.rfcoscol
GWHOST= issp2n99
GWSERV= sapgw04

In the /etc/services file the corresponding entry for sapgw04 must be made.

In transaction SM59 the RFC destination IFICOL_IHSAPFG has automatically been created as the TCP/IP connection with the program ID ihsapfg.rfcoscol.

In transaction DB2 under Checks/Settings, the destination IFICOL_IHSAPFG is specified for the subsystem SGF2.

The database request module (DBRM.db2clddb) must be correspondingly bound to the subsystem SGF2 with the plan name CKDF620 and the collection CKDF620.

**Additional Configuration Steps When Using a Gateway**

When you want to use several application servers connected to one and the same DB2 subsystem for monitoring and you want to use only one rfcoscol, call transaction SM59 to direct your RFC connection to the application server that has an RFC connection open to the rfcoscol.

In this case, you have to specify the gateway options (gateway host and gateway service) in transaction SM59 for those application servers that have no RFC connection to an rfcoscol. It is not possible to connect directly to the rfcoscol that is already connected to a different application server, since an rfcoscol only accepts requests from one application server.

Be aware that in this case you are becoming dependent on a single, specific application server, which is used as the gateway. As soon as you stop either the
gateway application server or the corresponding rfcoscol, you will lose your connection and you are no longer able to monitor database performance.

**Additional Configuration Steps in an MCOD Environment**

If you run multiple SAP systems in one database (MCOD) and want to monitor them with a single RFCOSCOL, a package needs to be bound per SAP system and all the collections (=package lists) need to share a common plan. Be aware that this requires changes to the predefined BIND and GRANT jobs.

Assume that in an MCOD landscape with two SAP systems, you have SAP system 1 and SAP system 2. The SAP system 2 is used as a gateway to the rfcoscol. The BIND statement of the plan would be as follows:

```
BIND PLAN (<ifi_plan> OWNER(<sapsidadm_system>) –
PKLIST (<sap_system1>.*, <sap_system2>.* ) –
ACTION(REPLACE) RETAIN ISOLATION(UR) DYNAMICRULES(RUN) –
ACQUIRE(USE) RELEASES(COMMIT) –
CURRENTDATA(NO) KEEPDYNAMIC(YE S) –
SQLERROR(CONTINUE)
```

The variables are described in the following table:

<table>
<thead>
<tr>
<th>Variable or Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ifi_plan&gt;</td>
<td>Name of plan used by rfcoscol</td>
</tr>
<tr>
<td>&lt;sap_system1&gt;</td>
<td>Name of collection of SAP system 1</td>
</tr>
<tr>
<td></td>
<td>Recommended value: C&lt;SAPSID&gt;&lt;REL&gt;, for example, CKDF620.</td>
</tr>
<tr>
<td>&lt;sap_system2&gt;</td>
<td>Name of collection of SAP system 2</td>
</tr>
<tr>
<td>&lt;sapsidadm_system&gt;</td>
<td>User ID of the &lt;sapid&gt;adm user of SAP system</td>
</tr>
</tbody>
</table>

Also, additional privileges are necessary. They can be granted by issuing the following commands:

```
GRANT EXECUTIVE ON PACKAGE <sap_system1>.* TO <sapsidadm_system>;
GRANT PACKADM ON COLLECTION <sap_system1> TO <sapsidadm_system>;
```

As a last step, the RACF definitions need to be enhanced. The ID of the user who started the rfcoscol (here: <sapsidadm_system 2>) has to be authorized for all RACF secondary groups of an MCOD landscape. For more information, see the SAP documentation Planning Guide: z/OS Configuration for SAP on IBM DB2 UDB for z/OS.

The RFC destination in SM59 for SAP system 1 must be configured in the way that the SAP system 2 is used as gateway. This means when you create the RFC destination, you must specify the gateway host and the gateway service in addition to the parameters described in Configuration of the RFC Destination [Page 148]. As soon as the rfcoscol for SAP system 2 is up and the application server is running, you can test the RFC destination.
Be aware that in this case the application server - which is used as gateway - must be running. As soon as you stop either the gateway application server or the corresponding zfcoscol, you will lose your connection and no longer be able to monitor database performance.
36.7 Maintaining rfcoscol in Transaction DB2

The rfcoscol destination is configured automatically during system installation. You only need the following procedure to configure the RFC destination manually.

You should adjust the gateway options (gateway host and gateway service) manually.

You call transaction DB2 and choose Checks/Settings → SAP Collector Settings. On this screen, you can specify configuration parameters and administer rfcoscol.
The required configuration parameters are as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System</strong></td>
<td>For each database name, the RFC destination name as specified in transaction SM59 for the RFC destination of the RFCOSCOL. Default: SAPOSCOL_&lt;DB_HOST&gt;</td>
</tr>
<tr>
<td></td>
<td>This RFC destination always exists and is automatically created by the SAP system.</td>
</tr>
<tr>
<td><strong>Collection ID</strong></td>
<td>Collection ID that the rfcoscol is going to use for this SAP system Default: C&lt;SAPSID&gt;&lt;SAP release&gt;, for example, CKDF640.</td>
</tr>
<tr>
<td><strong>Number of users</strong></td>
<td>Maximum number of concurrent users Recommended value: 5</td>
</tr>
<tr>
<td><strong>HFS path of collector</strong></td>
<td>Directory in which the rfcoscol resides. For the application server on z/OS, this is /usr/sap/&lt;SID&gt;/SYS/exe/run.</td>
</tr>
<tr>
<td><strong>HFS path of trace directory</strong></td>
<td>Directory which contains the trace files.</td>
</tr>
<tr>
<td><strong>SAP Collector user ID</strong></td>
<td>User ID of the user that starts rfcoscol</td>
</tr>
</tbody>
</table>

When the parameters are correct and have been saved, you must bind the database request module against the DB2 subsystem by choosing **Bind SAP Collector**. This bind is necessary, since for DB performance monitoring the rfcoscol issues SQL statements against the connected DB2 subsystem. The database request module, which is released together with the rfcoscol, must have been put in the same directory as the rfcoscol, otherwise the execution of the generated **BIND** job will fail. As soon as you change the directory in which the rfcoscol and the corresponding database request module is located, or as soon as you want to use a different level of the rfcoscol residing in a different directory, you have to change the name of the directory accordingly. For the bind, the information about rfcoscol and the corresponding database request module are taken from the specified directory.

Prerequisites for the submission of jobs are that the z/OS password of the SAP user and the general profile parameters are maintained in the JES Interface (transaction DB2J). This applies to the following user IDs:

- User ID submitting the **GRANT** job
- User ID binding the database request module
- User ID starting the rfcoscol

In addition, the user ID submitting the **BIND** job for the first time must have DB2 **SYSADM** authority. For rebinds with the same job, you can use the user ID that starts the rfcoscol, since all necessary DB2 authorization rights are already granted to this user if you have once submitted the **GRANT** job. If you choose to use a different user ID to submit the **BIND** job, neither a user ID having the DB2 **SYSADM** authority nor the user ID starting the rfcoscol, make sure that this user has the appropriate DB2 authorization rights to bind a plan and a
package. Be sure that all rfcoscols using the plan are stopped. You can only rebind the plan if it currently is not in use.

You have to resubmit the BIND job, when the database request module is changed. This is always the case when a new rfcoscol level is installed.

As soon as you change the plan name, or the collection ID, or the user ID that starts the rfcoscol, you have to resubmit the BIND job using the new setting.

When the plan is bound, you should grant the appropriate DB2 authorization rights, for example, for a specified plan and collection ID to the user ID that starts the rfcoscol by making use of the appropriate button. Also, the user ID that starts the rfcoscol requires the MONITOR1 and MONITOR2 privilege to collect trace data. The TRACE privilege to start the monitor trace needs to be granted to the user ID as well.

In addition, the user ID submitting the GRANT job must have DB2_SYSADM authority. You do not need to resubmit the GRANT job if only the rfcoscol is replaced.

As soon as you change the plan name, or the collection ID, or the user ID that starts the rfcoscol, you have to resubmit the GRANT job using the new settings. If you do not resubmit the GRANT job, rfcoscol will fail with DB2 authority problems.

If you ran the GRANT job and permission problems were detected with the user ID submitting the grants, ensure that the READ permission is set for the database request module.

You can start the rfcoscol including its alert router thread by choosing Start SAP Collector. If you want to start multiple rfcoscols, you can choose Start SAP Collector as often as required to start a rfcoscol. Be aware that a separate rfcoscol and alert router thread is started each time.

If you run the START job and permission problems are detected with the user ID starting the rfcoscol, ensure that the EXECUTE permission is set for this particular user ID.

To check whether the connection to rfcoscol works, you can press the button SAP Collector Ping.

The button Stop SAP Collector stops rfcoscol. However, if a connection to rfcoscol has not been established, it cannot be stopped using this button. You must then stop it manually. This procedure is described in Stopping rfcoscol Manually [Page 147].

If the automatically created jobs do not exactly match your requirements, you may change the job skeletons in the JES Interface. The names of these jobs are as follows:

- SAPCLGRANT
- SAPCLBIND
• **SAPCLSTART**

If you want to execute these jobs directly on z/OS, you can download them. Call transaction DB2J and choose *Change* → *Save* as a local file from the menu.

The configuration parameters plan name and collection ID are not only used for bind or grant purposes, but also used during the collection of performance data. They are passed to the `rfcoscol`, which uses them for the DB2 connection. Therefore they must be changed accordingly if the user wants to switch between different `rfcoscol` setups.

Since some requests to collect performance data are very resource intensive, these requests can be limited. The maximum number of concurrent users can be specified and adjusted to your needs.
36.8 Troubleshooting for rfcoscol

Analyze the error message in the SYSLOG (transaction SM21) and in the db2cl.<pid>*.trc trace file. The trace files are stored in the start directory of rfcoscol. If the database connect was successful, you can also display the trace files from transaction DB2. Choose Traces/Logs → IFI Data Collector → Show Trace. At this point, the trace level can also be dynamically set for the IFI Data Collector of rfcoscol. If you call rfcoscol using ST04, rfcoscol displays its own version number in the db2cl.<pid>*.trc trace file.

If you experience problems submitting the rfcoscol start job, the origin of the problem might be a limitation of the BPXBATCH job. For details on BPXBATCH, see the IBM documentation z/OS –UNIX System Services Command Reference. Problems might also be encountered if the STEPLIB environment variable is set in the UNIX System Services shell of the user. In this case, the STEPLIB environment variable of the jobs (SAPCLBIND, SAPCLGRANT, and SAPCLSTART) might be ignored.

**Error Messages**

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Meaning</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTIAR could not be loaded</td>
<td>Environment variable not set</td>
<td>Check environment variable STEPLIB</td>
</tr>
<tr>
<td>SQLCODE -923 not found</td>
<td>Plan name</td>
<td>Check plan name in DB2</td>
</tr>
<tr>
<td>SQLCODE -805: call set failed</td>
<td>Current DBRM or package name not found in plan</td>
<td>Rebind the plan with the correct DBRM</td>
</tr>
</tbody>
</table>

**Environment Variables**

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CL_ASYN_NO</td>
<td>0(Def),1</td>
<td>Alert not started if value set to 1</td>
</tr>
<tr>
<td>CL_TRACE</td>
<td>0(Def) to 5</td>
<td>Trace level. Setting a trace can impair the performance of the data collector.</td>
</tr>
<tr>
<td>CL_FFLUSH</td>
<td>0(Def),1</td>
<td>The trace output will be flushed to the trace file immediately if the value is set to 1.</td>
</tr>
</tbody>
</table>
37 Database Performance Monitor

The DB2 performance indicators are made available at different levels of reporting by RFCOSCOL:

- Subsystem-wide scope: This includes performance data for the DB2 resources that are shared by all of DB2 users, for example, buffer pools and logging.
- Thread-wide scope: The thread-wide monitoring is very useful if you are looking at a specific SAP work process, for example, a long-running batch job, and its consumption of DB2 resources. This is due to the static relation between an SAP work process and its associated DB2 thread.
- Statement-wide scope: For each SQL statement cached, DB2 maintains a number of performance indicators. This data presents a prime source of performance monitoring and tuning indicators.
- Data set scope: Data set monitoring allows to detect I/O performance problems of individual data sets.

Moreover, the DB Performance Monitor includes a set of panels that display the current DB2 system parameters. The correct system parameters are crucial for the efficient operation and performance of SAP Systems.

If DB2 data sharing is employed, all the information is provided for every DB2 member of the data sharing group.

While the integrated DB2 monitor is a very good vehicle for monitoring DB2 in SAP system environments, it lacks some of the features available in stand-alone DB2 monitors. For example, reporting the subsystem-wide statistics data for any given period (DB2 PE Statistics report set) or post-processing of various DB2 performance traces (DB2 PE Record Trace report set) should be obtained by other means.

In the DB performance monitor area, both the ICLI Performance Services and the ICLI Alert Router are no longer available.

The RFCOSCOL connects to the DB2 subsystem for which the performance data should be gathered. Some of the collected data is only available if DB2 monitor trace class 1 is active. For that reason, RFCOSCOL starts that trace class, if necessary. Moreover, the SAP system aims to ensure that the DB2 accounting classes 2 and 3, the performance trace IFCID 318, and the statistics trace IFCID 199 are always on. For more information, see Automatic Start of DB Traces [Page 171]. As soon as a request for performance data comes in, the RFCOSCOL calls DB2 via the IFI interface. The collected data is examined and written into regular tables which are owned by the SAP system. Then the calling transaction reads this data from the corresponding tables.

To keep the SAP monitor tables from growing excessively, the content of the tables is automatically deleted.

For details on DB2 trace classes and the IFI interface, see SAP on IBM DB2 UDB for z/OS: Database Administration Guide.

Access

To access the performance monitor, choose:

Tools → Administration → Monitor → Performance → Database → Activity

or call transaction ST04.

The Database Performance Analysis:DB2 UDB for z/OS screen appears.
All ST04 functions can be accessed by calling transaction DB2. For more information, see the Central Navigation Tool [Page 20].
37.1 DB2 Subsystem Activity

For information about the overall DB2 database performance, choose DB2 subsystem activity on the initial screen of the performance monitor. The DB2 Subsystem Activity - Overview screen displays important DB2 statistics, counters, percentages and ratios of the connected DB2 subsystem. If the DB2 subsystem is a member of a data sharing group, you can access that information for any member of the group.

Most of the DB2 statistics field values increase while the DB2 subsystem is running. As soon as DB2 is started, the accumulation process begins and it continues until the DB2 subsystem is stopped.

The data is displayed in one of three forms:

- **Accumulated values**
  - Values summed up over the measurement period

- **Current values**
  - Values at the current time

- **High water mark values**
  - Maximum value the counter has reached since the time the DB2 subsystem was started the last time

Values can be displayed for three measurement periods:

- Since DB2 start
- Since reset

When you choose DB2 Subsystem activity in the Database Performance Analysis: DB2 UDB for z/OS screen, the system displays the data since the DB2 subsystem start by default.

If DB2 data sharing is employed, you can access the information for any DB2 member by selecting that member from the dropdown list box at the top of the screen.
The type of period measured you select remains the same for all subscreens of the *DB2 Subsystem Activity* screen.

The following standard functions are available:

**Standard Functions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset</td>
<td>Sets the reset time to the current time. All accumulated values are set to zero.</td>
</tr>
<tr>
<td>Since reset</td>
<td>The period measured is the duration between the previously determined reset time and the current time.</td>
</tr>
<tr>
<td>Since DB start</td>
<td>The period measured is the duration between the time the subsystem was started and the current time.</td>
</tr>
<tr>
<td>Summary</td>
<td>Generates a list with some data for the DB2 subsystem. This list can be downloaded to your PC or printed.</td>
</tr>
</tbody>
</table>

**Detail Screens for DB2 Subsystem Activity**

For information on specific areas you can choose, for example, *Buffer pool activity* or *SQL activity*. The detail screens have the same functions (*Reset*, *Since reset*, and so on) as the screen *DB2 Subsystem Activity: Overview*.

From the *DB2 Subsystem Activity: Buffer Pool List* screen, you can access detailed information on a specific buffer pool by selecting the buffer pool and choosing *Details*. 
37.2 Lock Waits

A lock wait is a situation where a thread waits to be able to use a resource locked by another thread.

You can view all lock waits within your DB2 subsystem by choosing Lock waits in the DB2 Subsystem Activity: Overview screen.

The Lock Waits screen displays a list of all waits. The list includes the following:

- the locked resource;
- the process currently using this resource;
- the process waiting to use this resource.
37.3 DBM1 Virtual Storage

To monitor DBM1 virtual storage usage, choose **DBM1 virtual storage monitoring**. The information provided allows you to determine the main consumers of the DBM1 address space for each DB2 subsystem or member. This facilitates the timely identification of bottlenecks in this important address space for DB2.

These statistics are periodically, not synchronously, determined in the DB2 statistics interval, which is controlled by the DB2 system parameter **STATIME**. Transaction ST04 or DB2 provides the history of the past week. The daily CCMS job (**RSDB2_COLLECT_DAILY**) automatically deletes older DBM1 virtual storage statistics. DB2 provides these statistics in IFCID 225, which is contained in the Statistics Trace class 6.
### 37.4 Global Times

This function displays the times that should represent the total system activity as a percentage for the activity time and the time spent in DB2. Additionally, you get the average times of the different kinds of suspension in milliseconds.

The global times can be presented in two ways: current snapshot and long-term monitoring. The current snapshot view shows the times as the averages of all active threads, in other words, threads that are currently performing a transaction. The long-term monitoring view presents times that are calculated as averages of the snapshots of active threads that are collected in the DB2 statistics interval. The timers remain in that table for about 3 days before they are automatically deleted.

To be able to display global times, the accounting trace classes 1, 2, and 3 must be active. They are automatically started during SAP system startup. You can start these trace classes manually by issuing the following command:

```sql
START TRACE (ACCTG) CLASS(1,2,3)
```

This command is among the predefined commands of the [DB2 Commands](#) section of transaction ST04.

All the values are global times that have been calculated as the averages of all threads that are active at the time of a snapshot. Extreme values of a single thread may influence the averages displayed for all threads considerably if there is a low number of threads. Also, the global times are not necessarily representative since the thread activity may be different at snapshot time than at other times.

The times displayed can help guide your investigation into application performance and tuning.

As with the DB2 subsystem activity, you can also access information on global times for other DB2 members of the data sharing group. To do this, select the required member from the dropdown list box at the top of the screen.
37.5 Thread Activity

To display the list of active threads connected to the DB2 subsystem you are monitoring, choose Thread activity in the Database Performance Analysis: DB2 UDB for z/OS. The list of active threads (threads that are currently performing a transaction) is refreshed when you call this screen.

If there is no transaction running within a thread at the time of the snapshot and the thread holds no locks, then the thread will not be displayed.

The measurement period is between the start of the currently running transaction of a thread and the time the thread list has been refreshed.

As with the global times the thread-level time information also relies on the accounting traces 1, 2 and 3 being on.

DB2 Threads and SAP Work Processes

Each DB2 thread is associated with exactly one SAP work process. As work processes can create secondary connections, multiple DB2 threads can exist for a single work process. The correlation between a DB2 thread and its corresponding SAP work process is established by SAP system ID, SAP system number, host name of the application server, the work process number and the database connection name.

Among the information that is displayed in the thread overview are the SAP end user, the SAP transaction code and the ABAP report name that the currently active database transactions serve. The RFCOSCOL identifies itself in the thread list under transaction name “RFCOSCOL” and report name “Snapshot”. The alert router thread of RFCOSCOL uses the transaction name “RFCOSCOL” and report name “Alert router”.

You can select any thread listed and use one of the following functions:

**Thread Activity Functions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Buffer pool act.</strong></td>
<td>Displays all buffer pools used by the selected thread in the measurement period. For information on read and write activity, both to and from a specific buffer pool, select the buffer pool and choose Details.</td>
</tr>
<tr>
<td><strong>Locking activity</strong></td>
<td>Displays locking information on the selected thread. Locking problems can be further investigated by choosing Locked resources or lock waits, to examine resources locked by or waited for by the thread monitored.</td>
</tr>
<tr>
<td><strong>Lock waits</strong></td>
<td>Displays a list of resources that the selected thread is waiting for.</td>
</tr>
<tr>
<td><strong>Locked resources</strong></td>
<td>Displays a list of the resources locked by the selected thread.</td>
</tr>
<tr>
<td><strong>DS Locking</strong></td>
<td>Data sharing only: The system displays data sharing locking.</td>
</tr>
<tr>
<td><strong>SQL activity</strong></td>
<td>Displays number of times SQL statements are executed when processing a DB2 application. The sections Total DML (Data Manipulation Language) and Total DCL (Data Control Language) display information for each thread. Total DDL (Data Definition Language) shows a table containing the number of executions performed by various DDL SQL statements for each of the applicable object types.</td>
</tr>
<tr>
<td><strong>SQL statement</strong></td>
<td>Displays information about the current SQL statement being processed.</td>
</tr>
</tbody>
</table>
executed by the monitored thread. If you need to see all the
details of the access path for that statement, choose Explain
statement.

<table>
<thead>
<tr>
<th><strong>Group BP</strong></th>
<th><strong>Data sharing only.</strong> The system displays all active group buffer pools.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Times</strong></td>
<td>Displays various times for processing in DB2 and out of DB2 and suspension times. Displays the response times of various actions performed by the currently executing transaction of the selected thread. The times displayed can help guide your investigation into application performance and tuning. Accounting trace class 2 must be started for this information. If you use accounting trace class 3, the values are more detailed.</td>
</tr>
</tbody>
</table>

In general, the thread field values are accumulated from the beginning of the currently executing database transaction.

At any time, data from the same point in time is shown on all thread detail screens. This means you can examine a specific thread in detail with data referring to the same point in time.

If DB2 data sharing is employed, you can access the thread information for any DB2 member by selecting that member from the dropdown list box at the top of the screen. For a consolidated view of the DB2 threads of all members in the data sharing group, you can choose ALL from the dropdown list box.
37.6 Cached Statement Statistics

This function provides monitoring capability for the prepared SQL statements that reside in the dynamic statement cache. The information about the cached statements will be gathered during their preparation and execution.

Trace IFCID 0318

Statistics information for cached statements is invaluable for performance tuning on the level of individual SQL statements. This information is only collected if the DB2 performance trace IFCID 318 is turned on.

Therefore, at SAP system startup this trace is automatically turned on. By subsequently stopping and restarting IFCID 318, any statistics that were previously collected for existing statements in the cache will be cleared. If a DB2 system executes dynamic SQL statements from the dynamic SQL statement cache while the IFCID 318 is off, DB2 will return a record for any qualifying statements in the cache, but most of the statistics fields for those records will show all zeros.

Filtering

You can select any of the following statistics fields for filtering. DB2 will then return information about the statements that have the highest values in the selected field.

Statistics Fields for Selection: Filtering

<table>
<thead>
<tr>
<th>Statistics Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of executions</td>
</tr>
<tr>
<td>Number of buffer reads</td>
</tr>
<tr>
<td>Number of getpages</td>
</tr>
<tr>
<td>Number of rows examined</td>
</tr>
<tr>
<td>Number of rows processed</td>
</tr>
<tr>
<td>Number of sorts</td>
</tr>
<tr>
<td>Number of index scans</td>
</tr>
<tr>
<td>Number of tablespace scans</td>
</tr>
<tr>
<td>Number of parallel groups</td>
</tr>
<tr>
<td>Number of buffer writes</td>
</tr>
</tbody>
</table>

You can additionally provide a threshold value for the statistics fields. DB2 will then return information about all statements that exceed the given value for the selected statistics field.

Data Sharing Enabled View

If DB2 data sharing is employed, you are able to access cached statements statistics for any DB2 member of the data sharing group by selecting that member from the dropdown list box at the top of the screen. For a consolidated view of the SQL statements in the dynamic statements cache or all members in the data sharing group, select ALL from the dropdown list box.

SQL Statement Text and Explain Function

The records displayed provide identifying information, statistics, and part of the SQL statement text. Due to the potential size of the SQL statement text, only the first 60 bytes are shown. If you want to see the entire statement text for a particular statement, click this.
statement text. If you need to see all the details of the access path for the statement, choose *Explain statement*.

For more information, see *EXPLAIN* under Database Tools [Page 175].

**Correlation of SQL Statement and ABAP program code**

Every SQL statement that is submitted by an ABAP program is tagged with information on this program (IFC field QW0316UI). This correlation of SQL statements and ABAP source code enables to navigate directly from an SQL statement to the ABAP source code that has initiated the dynamic preparation of the statement.

In SAP BW systems the SQL statement tag also includes the STATUID value for OLAP queries. This allows you to inspect the BW statistics of OLAP queries.

For MCOD landscapes it is recommended to exploit the navigation feature only for the statements belonging to the SAP system you are working in.
37.7 Data Set Statistics

Using the Data Set Statistics button on the main panel of transactions ST04 and DB2 allows you to analyze performance data about the data sets belonging to the DB2 page sets. Performance data for the last completed statistics interval, which is defined by the DB2 ZPARM DSSTIME, is displayed.

The default value for DSSTIME is five minutes. Upon completion of an interval, the statistics values are reset. Bear in mind that IFI only retrieves statistics for data sets for which there is at least one I/O operation per second. Additional filters are available to further restrict the amount of data sets displayed. Data set statistics are collected by IFCID 199, which is contained in the statistics trace class 8. The SAP system aims to ensure that this trace class is always active. For more information, see Automatic Start of DB Traces [Page 171].
37.8 Installation Parameters

To display the installation parameters of the DB2 subsystem, choose one of the following methods:

- Choose Installation parameters on the Database Performance Analysis: DB2 UDB for z/OS screen.
- Call transaction ST04 or DB2 and choose Installation parameters.

The first tab provides an overview of the most relevant parameters. All the SAP-specific parameters, which are described in DB2 System Parameters [Page 50], are presented on that screen.

You can display more detailed information on the following subsets of parameters:

- Buffer pools
- Data sharing
- Storage
- Tracing
- Locking
- Protection and data definition
- Logging
- Application programming defaults
- DDF
- Miscellaneous

To generate a list of a subset of the installation parameters, choose List format. The list of values can be downloaded or printed.

If DB2 data sharing is employed, it is also possible to view the installation parameters of the other members of the data sharing group.
37.9 PTF Status

Since the PTF status of a DB2 subsystem is not displayed, and to help you to keep your system at the latest maintenance level, the database monitor provides a diagnostic function. You can access this function by calling Installation Parameters in transaction ST04 and choosing DISPLAY MEPL in the menu. The DB2 utility DIAGNOSIS is called via the Stored Procedure DSNUTILS. The result, the module entry point list (MEPL), is then displayed in raw format.
37.10 Automatic Start of DB Traces

The SAP system tries to ensure that the relevant DB2 traces are always on. These traces are an invaluable source of performance indicators, which allow you to tune the DB2 subsystem directly and to get appropriate diagnostics information in case of performance problems. The periodic job RSDB2_COLLECT_HOURLY accomplishes this task. It runs at the startup of the SAP system and once every hour. Its goal is to start both accounting trace classes 1, 2 and 3, IFCID 318 of performance trace class 30 and IFCID 199 of statistics trace class 8. However, if the job determines that accounting trace class 2 has been stopped, it does not restart that trace.

In exceptional circumstances it may be advisable to turn off these traces and to make sure that they are not restarted by the SAP system. To do this, call transaction DB2 and access the SAP Collector Settings screen. To disable the automatic start of the traces, enter TOFF in the command field. To switch on the traces, enter TON.
37.11 DB Alert Router

RFCOSCOL also serves as database alert router that catches DB2 alerts via the IFI interface of DB2. Since catching exception events is a waiting task, the RFCOSCOL starts a separate alert router thread for each DB2 subsystem and for each system. This thread exclusively listens to DB2 alerts.

Each alert router thread connects to the DB2 subsystem for which the exception events should be collected and starts an appropriate statistics trace. The statistic trace is assigned to an OP buffer, into which DB2 writes the exception events. When an event occurs, DB2 informs the alert router thread that reads the OP buffer. After this call, the OP buffer is emptied by DB2.

Integration

The following are implementation considerations concerning the DB alert router:

- Be aware that there are only eight DB2 OP buffers per DB2 subsystem. Monitor programs - such as the RFCOSCOL - allocate OP buffers exclusively. To be able to run the database alert router, you must ensure that at least one OP buffer is unassigned.
- Automatic start of an alert router thread can be prohibited by using the corresponding environment variable.

The alert router thread reformats the exception events data and saves the data in specific tables for later display. Immediately, the database alerts become visible immediately in transactions ST04 and DB2. However, there may be short delay until the alerts for deadlocks, timeouts, extents, active log shortages and long-running transactions are integrated into the CCMS Monitor Set (transaction RZ20). That is due to the fact that new alerts are fed into the CCMS Monitor Set by a periodic job. This job is normally executed every five minutes. Thus, the maximum delay is five minutes. The database alerts are kept for 30 days and then deleted.

The alert router thread catches all exceptions events of the DB2 subsystem and exception events belonging to non-SAP data that may coexist in the same subsystem. Also in a data sharing environment, the alert router thread catches exceptions of all members in the same data sharing group. In addition, in an MCOD environment you get the exception events of all SAP systems.

Be aware, since the exception event transactions do not filter, the scope of the exception events is larger than the SAP system you are running on. This feature allows you to have a single resource for monitoring.

Exception Events

There are different types of events:

- Deadlocks
- Timeouts
- Active log shortages
- Long-running transactions
- Lock escalations
- Data set extents

Long-running transactions are either long-running units of recovery (UR), which change at least one row, or long-running read-only transactions.
For the long-running URs, the amount of time which is required to roll back DB2 units of work depends on the number of DB2 logs written. That implies that long-running units of work that accomplish a lot of UPDATE, INSERT or DELETE activity are rolled back slowly. Transactions are rolled back for example if the application explicitly issues the SQL ROLLBACK statement or at restart of DB2 when DB2 previously abended.

The negative effects of long-running URs are that they restrict access to the affected resources during rollback and that they prolong DB2 restart times. This kind of alert is raised when DB2 detects long-running URs depending on the ZPARMs URCHKTH and URLGWTH. For details on these ZPARMs, see the SAP installation documentation.

Long-running read-only transactions may also have a negative impact on the system. For example, they may cause long lock suspensions if the REORG utility waits until such transactions complete and other transactions queue up behind the claim that the REORG utility acquires. The DB2 system parameter LONG-RUNNING READER on panel DSNTIPE (ZPARM LRDRTHLD) controls the threshold on the duration of read-only transactions that makes DB2 consider them long-running readers and raise and alert.

### Starting the DB Alert Router

When invoked by the SAP system, RFCOSCOL starts an alert router thread. If RFCOSCOL works for multiple SAP systems, it starts a thread for each DB2 subsystem and for each schema. If a corresponding thread has already been started, no new attempts are made to start it. If the rfcoscol is started from withing transaction ST04, the system also tries to start the rfcoscol alert router thread. Also, the SAP system (or, more specifically, report RSDB2_COLLECT_HOURLY) tries to start the alert router thread at startup of the SAP system and periodically every hour. Also, the alert router thread can be started manually. To do this, call either transactions ST04 or DB2 and choose DB Alert Router. Then choose Manually start Alert Router.

The DB2 trace class 31 is used for the alert router. It is assumed that only rfcoscol uses this trace class. If DB2 traces of class 31 are found when the alert router is started and these are not assigned to an alert router of an SAP system, they are stopped. In this way, it is ensured that DB2 traces that hang when an rfcoscol crashes do not pile up. Instead, they are cleaned up with the next start of an alert router.

When using data sharing, only one alert router is started to collect events for the complete data sharing group, even if rfcoscols are active for different members.

To check if the DB alert router is running:

1. Call transaction ST04 or DB2.
2. Choose Thread Activity
3. Sort the column plan name.

   If the alert router is running, a corresponding DB2 thread with a plan name specified for RFCOSCOL is displayed.

### Analysis Monitors

From the CCMS monitor set, you can access the following analysis monitors:

<table>
<thead>
<tr>
<th>Monitor</th>
<th>Transaction</th>
<th>Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deadlocks</td>
<td>DB2D</td>
<td>–</td>
</tr>
<tr>
<td>Timeouts</td>
<td>DB2T</td>
<td>–</td>
</tr>
<tr>
<td>Active log shortages</td>
<td>–</td>
<td>RSDB2LOGSHORT</td>
</tr>
<tr>
<td>Long-running transactions</td>
<td>DB2U</td>
<td>–</td>
</tr>
<tr>
<td>Extents</td>
<td>DB02</td>
<td>–</td>
</tr>
<tr>
<td>Lock escalations</td>
<td></td>
<td>RSDB2LOCKESCAL</td>
</tr>
</tbody>
</table>
All monitors listed above are also accessed by calling transaction ST04 or DB2.

In the monitors, you can analyze the details of an event. You can select a time period and display events that occurred during that time. To display a hierarchical list of the events, choose display. For detailed information on an event, double-click an entry.

**DB Alert Settings**

The alert router has the following standard settings for each extent alert threshold values:

- Low threshold = 25
- High threshold = 80

If the DB2 feature to optimize extent sizing is enabled by setting ZPARM MGEEXTSZ to YES, which is highly recommended, then the extent monitor is not relevant, because DB2 ensures that the maximum number of extents is not reached.

If the low threshold is exceeded, a yellow alert for each new extent of the data sets above 25 will be shown in the CCMS Monitor Set (transaction RZ20). If the high threshold is exceeded, a red alert will be shown in the CCMS Monitor Set.

To react to this alert, you can choose the analysis tool in the CCMS Monitor Set, for example, by double-clicking the node. This leads you to the extent monitor (transaction DB02). It is possible to change the secondary quantity and to alter it from the extent monitor.

Under normal circumstances, these settings should be sufficient. If you want to change the low and high threshold values of the extent alerts:

1. Call transaction ST04 or DB2.
2. Choose DB Alert Settings.
3. Choose Change. This enables you to change the low and high threshold values.

**Deleting Table Entries of Alert Information**

SAP provides a program that automatically deletes table entries of database alerts that are older than 30 days. The program is set to run once every day. You can also delete the alerts in the alert monitor. This does not, however, delete the table entries in the history table and this means you can display events for up to 30 days even after the alert has been deleted.
38 Database Tools
38.1 Proactive Redirection of Application Servers to Other DB2 Subsystems

In DB2 data sharing topologies, the need might arise to redirect the work processes of an SAP application server to a different DB2 member of the data sharing group. One scenario is that a specific member is supposed to undergo maintenance and thus has to be stopped. Optimally, this operation should not be noticed by end users. Therefore, the SAP application server offers the capability to dynamically reconnect its work processes to a different DB2 member. The configuration options are described in the SAP installation guides.

To actually initiate the reconnection to a different DB2 subsystem, call transaction DB2 or ST04. From these screens, you can inspect the currently connected database host for each work process and then redirect the work processes. To check the status of work process to database host mapping, you can choose *Active DB Connections*.

The database names that are specified for the work processes are neither DB2 subsystem nor DB2 member names. Instead they are the logical names of the database connection as specified in the connection profile `connect.ini`.

The *DB Connection List* shows all possible logical connections for the application server. Among others, it gives the DB2 SSID of the logical connection and the z/OS system on which the DB2 subsystem is running. By double-clicking a connection, the work processes of the application server are reconnected to the subsystem, which is specified by the connection.
### 38.2 Data Sharing Topology

On occasion, it might not be clear which SAP application server is connected to which DB2 member when data sharing is employed. In this situation, it is useful to view the data sharing members and their connected SAP application servers.

To access this information:

1. Call transaction ST04 or DB2.
2. Choose *Data Sharing Topology*.

![Data Sharing Topology - Correlation AppServer/DB2 Member](image-url)
38.3 DB2 System Catalog

Access

To access the DB2 catalog browser, call one of the following transactions:

- Transaction DB2C.
- Transaction DB2 or ST04 and choose DB2 Catalog Browser.

You use SQL statements to access parts of the catalog:

1. Enter your statement.
2. Choose Execute.
3. The result is displayed as a list.

Only SELECT statements on tables created by SYSIBM are allowed. This ensures that user access to tables is restricted to read-only mode and access is limited to the DB2 catalog.

A maximum of 36 statements can be executed with this transaction.

You can only enter one statement at a time. Statements must not end with a semi-colon.

The SQL statement is sent directly to the DBMS and does not undergo any further checks. Any errors in the SQL statement will lead to a database error.

Query Results

The query result is displayed as a list. Depending on the query, this table can become very wide and may exceed the maximum table width. If this happens, you will be prompted to modify the output. You can restrict the maximum length of character fields. You can also use Field Selection deactivate the columns you do not need to display. You can also select how many rows you want to display in the list. Choose Maximum rows for selection and specify the number of rows. To clear statements from the screen, choose Clear.
38.4 z/OS System Log

To display z/OS system logs, choose one of the following:

- Choose z/OS System Log on the Database Performance Analysis: DB2 UDB for z/OS screen.
- Call transaction DB2 and choose z/OS System Log.

This function is not available under JES3, since the program to get the system log (SDSF) does not exist. However, if you have a third-party-product working similar to SDSF, that is, a program that writes the console output into a partitioned data set, you can adjust the JCL job template. To do this, use the JES Interface and edit the job GET_CONSOLE created by SAPR3.

You need TSO access to display z/OS system logs. Both TSO and SAP system user names must be identical.

A sequential data set for the upload of z/OS jobs must exist with the following characteristics:

<table>
<thead>
<tr>
<th>Organization</th>
<th>PS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record format</td>
<td>VB</td>
</tr>
<tr>
<td>Record length</td>
<td>133</td>
</tr>
<tr>
<td>Block size</td>
<td>27930</td>
</tr>
</tbody>
</table>

Enter the name of this data set and the name of the SDSF HASPINDEX using transaction DB2J:
38.5 DB2 Commands

Authorization Profiles for DB2 Commands
SAP provides some DB2 commands which can only be changed with the authorization profiles S_DB2_COMM or S_DB2_DBADM.

You can, however, create, change, and delete your own DB2 commands with the authorization profiles S_DB2_EXPC or S_DB2_DBADM. In addition, you can decide when creating a command whether S_DB2_ALLU or S_DB2_EXPC is necessary for its execution with these authorization profiles.

With the authorization profile S_DB2_ALLU you can only execute DB2 commands that have the “command user” ALLUSER.

For more information about the default authorization profiles, see Authorization Profiles [Page 30] in Basic Operations.

Initial Screen
To display the initial Commands: Command list screen, choose one of the following:

- Choose Administration → System Administration → Monitor → Performance → Database → Activity → DB2 Commands
- Call transaction ST04 or DB2 and choose DB2 commands.

Executing a DB2 command from the SAP system produces the same result as executed from the z/OS system log.

Maintaining DB2 Commands
The following functions are available:

<table>
<thead>
<tr>
<th>Function</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute</td>
<td>Select a command and choose Execute. The next screen displays the result of the command.</td>
</tr>
<tr>
<td>Display</td>
<td>Select a command and choose Display.</td>
</tr>
<tr>
<td>Change</td>
<td>Select a command and choose Change. Make your changes and choose Save. You may also delete a command here.</td>
</tr>
<tr>
<td>Create</td>
<td>Choose Create. Make the necessary entries and choose Save.</td>
</tr>
</tbody>
</table>

You can execute the changed command in the DB2 Commands: Change command screen without saving it.

You can execute the new command in the DB2 Commands: Create command screen without saving it.

To return to the initial screen from the subsequent DB2 Commands: Display/Change/New command screens, choose Command list.
38.6 Buffer Pool Tuning

One of the most important objects in DB2 performance monitoring and tuning are buffer pools. After SAP system installation, all the table spaces and all the indexes are backed by some predefined buffer pools as described in DB2 Setup [Page 48]. Delivering a pretuned setup is impossible because customer environments are so special that a general solution cannot be applied. The recommended initial buffer pool attributes are sufficient for the functional verification of the product, but they are not optimal in terms of performance.

The following describes what you should do to tune DB2 buffer pools for optimal performance.

To tune the buffer pools in the SAP system environment, the administrator has to:

- Establish a base for performance evaluation
- Create a Top Tables Categorization List
- Isolate the tables in their own table spaces
- Determine tables-to-buffer pools mapping
- Determine buffer pools attributes

Each of these steps is described in more detail below.

Be aware that this procedure only works on the tables of one SAP system only. With MCOD, the procedure needs to be accomplished for each system. If the topology is such that each SAP system is equipped with a dedicated set of buffer pools, you must reflect this in table DB2BPTUNE. The table contents of this table can be altered by means of the Data Browser (transaction SE16).

Buffer pool tuning is an step-by-step process. Therefore, regularly perform the described procedure to:

- Base your tables-to-buffer pools assignments on a sample typical for your installation
- Detect changes in the workload
- Adjust the buffer pool attributes (sizes, thresholds)

Establishing a Base for Performance Evaluation

To evaluate the effects of tuning and detect development of new bottlenecks and performance deficiencies, you need to establish a base for performance evaluation. This can be done by collecting and storing the performance data over a longer period of time, but most importantly before and after any tuning activities.

The following data can be used to identify changes in the workload and throughput:

- Number of getpages per reporting interval
- Number of buffer updates per reporting interval
- CPU utilization

The following data can be used to detect reaching critical thresholds:

- Prefetch disabled due to no storage or no engines
- Asynchronous write disabled due to no engines
- Data Manager critical threshold
- Sort merge passes degraded
- Work file prefetch disabled
The most common reasons for reaching the thresholds are:

- Reducing VPSIZE too much
- Setting VPSEQT too low
- Setting deferred write thresholds too high

In such cases, reverse the negative effect by appropriate adjustments.

The following data can be used to check if storage is overcommitted:

- Page-ins for read and write
- z/OS paging activity

The following data can be used to measure effectiveness of the tuning:

- Overall hit ratio, which is derived as:
  \[
  \frac{(\text{total getpages} - \text{total pages read})}{\text{total getpages}}
  \]
  where total pages read = total synchronous reads +
  \[
  \text{pages read by sequential prefetch +}
  \]
  \[
  \text{pages read by dynamic prefetch +}
  \]
  \[
  \text{pages read by list prefetch}
  \]

- Random hit ratio, which is derived as:
  \[
  \frac{(\text{random getpages} - \text{random synchronous reads})}{\text{random getpages}}
  \]

- Read rate, which is derived as:
  \[
  \frac{\text{(total synchronous reads + all type prefetch reads)}}{\text{per interval}}
  \]

- Buffer updates per pages written
- Pages written per write I/O, which is derived as:
  \[
  \frac{\text{pages written}}{\text{(synchronous writes + asynchronous writes)}}
  \]

- Write rate, which is derived as:
  \[
  \frac{\text{(synchronous writes + asynchronous writes)}}{\text{interval}}
  \]

- z/OS DASD and Cache activity ratios

**Creating a Top Tables Categorization List**

The buffer pool tuning tool DB2B helps you to create the top tables categorization list. This includes determining which tables are referenced most often, what is the access pattern, and what is the current table size.

**Procedure**

1. Call one of the following transactions:
   - Transaction DB2B
   - Transaction ST04 or DB2 and choose Buffer Pool Tuning.

2. Table categorization is based on the data from transaction ST10. Statistics are available for various time intervals such as since startup, per day, and per week. Choose the interval that most closely represents your typical workload. Exclude time periods in which activities are done that are not performed as a daily routine, such as client copy or a large transport.

   The ST10 statistics on database activity do not take logical accesses on SAP cluster tables into account. Therefore, transaction DB2B does not give recommendations for these tables.
3. To display the top tables categorization list, choose **Table categorization**.

Sort the list with the AccessFrequency column to identify the top 20 to 30 entries.

The columns are named as follows:

- **Executes**
  
  \[
  \text{Executes} = \text{Fetch} + \text{Update} + \text{Insert} + \text{Delete}
  \]

- **Changes**
  
  \[
  \text{Changes} = 100 \times \left(\frac{\text{Update} + \text{Insert} + \text{Delete}}{\text{Executes}}\right)
  \]

- **Access Type**
  
  If \(\frac{\text{RowsAffected}}{\text{Executes}} > 5\)
  
  Then AccessType = sequential
  
  Else AccessType = random

- **Access Freq**
  
  \[
  \text{AccessFrequency} = \max(\text{Executes}, \text{RowsAffected})
  \]

- **TabSize**
  
  If \(\text{NPAGES} > 250 \text{ AND RECLENGTH} < 4056\) OR \(\text{NPAGES} > 30 \text{ AND RECLENGTH} > 4056\)
  
  Then Size = large
  
  Else Size = small

- **Category**
  
  The access category is a table attribute that describes what kind of access is typically done for the table. It also takes into account the table size.

**Access Category Derivation**

<table>
<thead>
<tr>
<th>Category</th>
<th>Derivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>AccessType=Sequential, Changes&lt;10%</td>
</tr>
<tr>
<td>C2</td>
<td>AccessType=Sequential, Changes&gt;10%</td>
</tr>
<tr>
<td>C3</td>
<td>AccessType=Random, Changes&lt;10%, TabSize=small</td>
</tr>
<tr>
<td>C4</td>
<td>AccessType=Random, Changes&lt;10%, TabSize=large</td>
</tr>
<tr>
<td>C5</td>
<td>AccessType=Random, Changes&gt;10%, TabSize=small</td>
</tr>
<tr>
<td>C6</td>
<td>AccessType=Random, Changes&gt;10%, TabSize=large</td>
</tr>
<tr>
<td>C7</td>
<td>Insert and/or Delete only</td>
</tr>
<tr>
<td>C8</td>
<td>This is a category reserved for VLOG, VDATA, VBHDR, and VBMOD because of a particular access pattern for these tables.</td>
</tr>
<tr>
<td>C9</td>
<td>Catalog</td>
</tr>
<tr>
<td>K2</td>
<td>C2, C5-C8 tables with 32 KB pages</td>
</tr>
<tr>
<td>K3</td>
<td>C1, C3, C4 tables with 32 KB pages</td>
</tr>
<tr>
<td>L2</td>
<td>C2, C5-C8 tables with 8 KB pages</td>
</tr>
<tr>
<td>L3</td>
<td>C1, C3, C4 tables with 8 KB pages</td>
</tr>
<tr>
<td>M2</td>
<td>C2, C5-C8 tables with 16 KB pages</td>
</tr>
<tr>
<td>M3</td>
<td>C1, C3, C4 tables with 16 KB pages</td>
</tr>
</tbody>
</table>
N1 Non-categorized tables with 16 KB pages
N3 Non-categorized tables with 32 KB pages
N4 Non-categorized tables with 4 KB pages
N8 Non-categorized tables with 8 KB pages
NI Non-categorized indexes
W3 32 KB page work files
W4 4 KB page work files

As the number of getpages is not available within the ST10 output (and is also important for categorizing tables) the procedure makes an assumption that there is a getpage for each row returned. This is another reason for regular buffer pool monitoring and adjusting the initial settings that are based on the ST10 output only.

For SAP BW systems, we recommended you assign dedicated buffer pools to the page sets of star schema fact and dimension tables.

The following parameters from table RSADMIN specify the buffer pools of these tables:

- `DB2_FACT_BPOOL_DATA`: Buffer pool for table spaces of fact tables
- `DB2_FACT_BPOOL_INDEX`: Buffer pool for index spaces of fact tables
- `DB2_DIM_BPOOL_DATA`: Buffer pool for table spaces of dimension tables
- `DB2_DIM_BPOOL_INDEX`: Buffer pool for index spaces of dimension tables

To maintain entries in the RSADMIN table, use report SAP_RSADMIN_MAINTAIN (see SAP Note 536074).

Sugg. BP

SAP and IBM recommend the following initial mapping:

### Initial Tables-to-Buffer Pools Mapping

<table>
<thead>
<tr>
<th>Buffer Pool</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP0</td>
<td>Catalog</td>
</tr>
<tr>
<td>BP1</td>
<td>4 KB page work files</td>
</tr>
<tr>
<td>BP2</td>
<td>Non-categorized 4 KB page tables</td>
</tr>
<tr>
<td>BP3</td>
<td>Indexes for non-categorized tables</td>
</tr>
<tr>
<td>BP4</td>
<td>Category C8 tables and indexes</td>
</tr>
<tr>
<td>BP5</td>
<td>Category C7 tables and indexes</td>
</tr>
<tr>
<td>BP6</td>
<td>Category C6 tables and indexes</td>
</tr>
<tr>
<td>BP7</td>
<td>Category C5 tables and indexes</td>
</tr>
<tr>
<td>BP8</td>
<td>Category C4 tables and indexes</td>
</tr>
<tr>
<td>BP9</td>
<td>Category C3 tables and indexes</td>
</tr>
<tr>
<td>BP10</td>
<td>Category C2 tables and indexes</td>
</tr>
<tr>
<td>BP11</td>
<td>Category C1 tables and indexes</td>
</tr>
<tr>
<td>BP8K0</td>
<td>Non-categorized 8 KB tables</td>
</tr>
</tbody>
</table>
You can change a buffer pool proposal that is provided used for a tablespace directly from the table categorization screen if you do not want to accept the suggested setting. The buffer pool must be active. To do this:

1. ALTER the buffer pool under *Sugg. BP*.
2. Select the entry from the tables categorization screen and choose *Alter TS*.

An ALTER TABLESPACE statement is generated and executed.

**Tip**
You can only change a buffer pool of a tablespace when the tablespace is stopped. At present, this can only be done manually. If you do not stop the tablespace, you get a runtime error with SQL error -626.

To be able to execute the ALTER TABLESPACE command, the user that the SAP work processes employ to connect to DB2 needs one of the following DB2 authorizations:

- **SYSADM**
- **SYSCTRL**
- **USE** privilege for the buffer pool

### Isolating Tables in Their Own Tablespaces

Tablespace rather than table is the object of most DB2 utilities (most notably REORG). Also, the buffer pool assignment is done on a tablespace basis. DB2 monitoring is better supported for tablespaces than tables. This is why it is better to isolate frequently accessed and/or large tables in their own tablespaces.

At installation time a large number of tables are already isolated in their own tablespaces. However, for some sites there could be additional tables that need to be isolated. Also, consider partitioning of the tables that are likely to grow very large. For more information, see *Storage Parameters [Page 301]*.

### Determining Buffer Pool Attributes

The size of the buffer pools depends on the amount of available central storage, the number of SAP Systems and other non-SAP system workload that is performed in the same CEC. Therefore, only relative sizes can be given under the assumption that a fixed amount of central storage is allocated to the SAP System. The buffer pool attributes depend on the category of tables that are assigned to the buffer pool.
The *Buffer Pool Settings* table gives an example of buffer pool definitions. Consider these settings as an initial iteration only in a continuous exercise of the buffer pools monitoring and tuning. Use DB2 performance monitors to examine all the relevant indicators of buffer pools efficiency such as hit ratios, critical thresholds reached, write ratios and so on. Based on these indicators and new samples of ST10, adjust buffer pool assignments and parameters to achieve optimal performance. Be particularly aware of hitting critical thresholds and overcommitting central or expanded storage.

It is important to understand that transaction DB2B is not a perfect source of indicators about tables’ access patterns. For example, very random access, treadmill effects, and buffer update reference patterns cannot be detected without using additional techniques. A workaround is to temporarily isolate a table with an unusual or unexpected access pattern into its own buffer pool and use buffer pool statistics that is now implicitly applicable to that table for better understanding of the access pattern. To access buffer pool tuning, call transaction DB2B and choose *Simulate BP settings*. Enter how much central storage is available for all buffer pools.

The sizes represent the state after the tables that account for 80% of accesses have been reassigned from the initial buffer pool setting as described in section 3 *DB2 Setup*. As the entire procedure is most likely to be iterative (20-30 tables at a time) you need to adjust the values accordingly.

The following is an example of buffer pool settings based on 600 MB central storage for buffer pools.

**Buffer Pool Settings**

<table>
<thead>
<tr>
<th>BP</th>
<th>Category</th>
<th>VPSIZE</th>
<th>VPSEQT</th>
<th>DWQT</th>
<th>VDWQT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP0</td>
<td>Catalog</td>
<td>1890</td>
<td>50</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>BP1</td>
<td>Sort work files</td>
<td>4725</td>
<td>100</td>
<td>70</td>
<td>50</td>
</tr>
<tr>
<td>BP2</td>
<td>Non-cat. 4 KB tables</td>
<td>11340</td>
<td>40</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Indexes for non-categorized</td>
<td>17010</td>
<td>40</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>tables</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BP4</td>
<td>C8 tables and indexes</td>
<td>945</td>
<td>10</td>
<td>70</td>
<td>50</td>
</tr>
<tr>
<td>BP5</td>
<td>C7 tables and indexes</td>
<td>945</td>
<td>30</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>BP6</td>
<td>C6 tables and indexes</td>
<td>6615</td>
<td>30</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>BP7</td>
<td>C5 tables and indexes</td>
<td>7560</td>
<td>30</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>BP8</td>
<td>C4 tables and indexes</td>
<td>19845</td>
<td>30</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>BP9</td>
<td>C3 tables and indexes</td>
<td>22680</td>
<td>30</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>BP10</td>
<td>C2 tables and indexes</td>
<td>4725</td>
<td>60</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>BP11</td>
<td>C1 tables and indexes</td>
<td>11340</td>
<td>60</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>BP8K0</td>
<td>Non-cat. 8 KB tables</td>
<td>567</td>
<td>40</td>
<td>50</td>
<td>30</td>
</tr>
</tbody>
</table>
Depending on the number of tables in a buffer pool, the $VDWQT$ can be different to what is recommended earlier. Set it so that the number of tables is slightly larger than the ratio $DWQT/VDWQT$. If there is only one table in the buffer pool, set $VDWQT=DWQT$.

To modify your buffer pool settings, call transaction DB2B:

1. Select the buffer pools you want to change on the Buffer Pool Suggestions screen.
2. Make your changes to the settings and choose Alter BP.

The buffer pool settings are not applicable for SAPinst. Use the sizes and parameters as recommended in section 3 DB2 Setup.

To be able to execute the `ALTER BUFFERPOOL` command, the user that the SAP work processes employ to connect to DB2 needs one of the following DB2 authorizations:

- SYSOPR
- SYSCTRL
- SYSADM
38.7 EXPLAIN

The EXPLAIN function uses DB2 EXPLAIN which uses a special table, PLAN_TABLE, that stores the results of the EXPLAIN function. If there is not a PLAN_TABLE, the EXPLAIN function creates the following tables:

- Plan table <SCHEMA>.PLAN_TABLE
- Statement table <SCHEMA>.DSN_STATEMENT_TABLE

These two tables are located in a tablespace whose name is generated by DB2. The corresponding database is called SYOOX<CCC>, where <CCC> are three randomly generated characters.

At present, the statement table is not exploited.

The DB2 EXPLAIN function can be invoked either by providing the SQL statement text or by specifying the statement ID (EXPLAIN STMTCACHE STMTID) or statement token (EXPLAIN STMTCACHE STMTTOKEN) from the DB2 statement cache. The advantage of specifying the statement ID or statement token is that EXPLAIN provides a description of the access path that is stored for the statement in the statement cache. Therefore, it allows you to inspect exactly the access path that the DB2 optimizer chose for a given statement. Explaining a statement using the statement text only simulates the DB2 optimizer at the current point in time where the catalog statistics may be different to the time when the statement was actually prepared.

If a statement that is manually entered in transaction ST05 is to be explained, SAP passes the statement text to EXPLAIN. In case a statement from the cached statement statistics panel of transaction ST04 should be explained, SAP uses the EXPLAIN option STMTCACHE STMTID. Statements from the SQL trace in transaction ST05 are usually explained by means of the option STMTCACHE STMTTOKEN. As the latter two options only work if a statement is still in the dynamic statement cache of DB2, the SAP system detects such situations and automatically invokes DB2 EXPLAIN providing the statement text. On the bottom of the panel that displays the result of EXPLAIN, it is indicated if the access path that is displayed is from the statement cache.
If an SAP user calls the EXPLAIN function, the statement is explained into the plan table and the resulting rows are read. The query number that is used in the plan table in column QUERYNO depends on the EXPLAIN option. For EXPLAIN with the statement text, 1548979326 is always used. For the other options, DB2 inserts the statement ID as query number in the plan table. If a statement is explained using the statement token, the STMTTOKEN field of the plan table additionally contains the statement token of the statement. The SAP system always deletes the rows in the plan table that would interfere with a statement to be explained. Enqueues are used to make sure that multiple users of EXPLAIN are serialized properly.

There are two views on the PLAN_TABLE:

- Standard view (plan table data presented in a hierarchical view)
- Expert view (content of the plan table)

If EXPLAIN is invoked with the statement text, then you have the possibility to dynamically turn on parallelism for the explanation of the access path of each individual SQL statement.

You can access EXPLAIN in the following ways:

- Call ST05 and choose Explain one SQL statement.
- Call transaction DB2 and choose Explain Statement.
- Call transaction ST04 and choose Cached Statement Statistics or Thread Activity.

If you have problems using EXPLAIN within the SAP System (duplicate records, invalid SELECT statement when reading the plan table), the <SCHEMA>.PLAN_TABLE is probably not defined correctly.

You can solve this problem by dropping the tablespace DSNDB04.PLANTABL or deleting all entries from the tables <SCHEMA>.PLAN_TABLE and <SCHEMA>.DSN_STATEMENT_TABLE. Then retry the EXPLAIN within the SAP System.
Statements used by the EXPLAIN function to create the tablespace are listed in the SAP documentation SAP on IBM DB2 UDB for z/OS: Administration Guide.
38.8 DB System Check

The DB System Check was originally designed to monitor the state of the database and to be used as a collection point for error messages and warnings that pertain to the database. With the introduction of the CCMS monitor set, the DB System Check has lost a lot of its importance. For this reason, DB System Check is only partially implemented on DB2 UDB for z/OS.

DB System Check allows you to check the installation parameters of your DB2 subsystem against the recommended values listed in DB2 Setup [Page 48]. The parameters that are recommended to be kept at default values are not checked.

These are passed on and recorded automatically in the CCMS monitor set.

Results of DB System Check

To view the results of the DB System Check, call one of the following transactions:

Call transaction DB16.

Call transaction DB2 and choose Checks/Settings → DB2 subsystem.

To run a report that checks the parameters against the recommended values listed in DB2 Setup [Page 48], choose Perform check. The report produces a list that informs you whether there is an error, a warning or whether the parameter is OK.

To display statistics, double-click an item in the list.

To display a summary, choose Summary.

To see details of the warning or error, choose Details. The warning or error message, the planned and actual values, and a corrective action message are displayed.

The warnings are automatically copied from the DB System Check to the CCMS Monitor Set Overview.

If you delete logs in the DB System Check, the corresponding record in the alert monitor is not affected. This means you might have to delete the corresponding alerts in the CCMS monitor set separately.
38.9 DBSL Trace

The DBSL trace allows you to trace CLI statement. It should only be used by experts. Tracing can have a considerable impact on the performance of the entire system.
DB2 Connect supports the generation of statistical information on the performance of the network between SAP application servers and database servers. While the SAP application server is running, you can switch the networks performance statistics on and off in SAP transaction ST04 or DB2. On tab strip Performance Tuning or on tab strip Traces/logs, select either the button DB2 Connect Diagnostics or DB2 Connect, respectively, to get to the panel that handles network statistics. To switch on the collection of network statistics, you need to configure the following information under DB2 Connect Diagnostics (see the graphic under “DB2 Connect Diagnostics Panel”):

- Define the time in seconds after which the statistical information is externalized and reset.
- Specify the number of snapshots that are taken. If you do not specify a number, statistics are collected until the collection process is explicitly stopped.
- Specify the application servers for which network statistics are collected
- Specify the coordinating server. This is the application server on which a batch program is periodically scheduled. The batch program invokes statistics snapshots on all application servers for which statistics are collected.
- The batch program invokes statistics snapshots on all application servers for which statistics are collected.

**DB2 Connect Diagnostics Panel**

The effective minimum sampling rate is 60 seconds. This limit is imposed by the general batch infrastructure, which is exploited to accomplish the snapshot processing.

To stop the collection of network statistics, select the Stop collection button. This can also be done if the number of samples was specified when starting the collection.

The granularity of the statistics is at the level of application servers. Therefore, you can specify the application servers for which statistics are collected. To choose a server, select its name in the list and then Collect for server. This can be accomplished repeatedly. To choose all servers at once, select Collect for all.
There needs to be exactly one application server that acts as batch server for the collection process. Choose an application server from the list and select Set batch server to declare it to be the batch server. Make sure that there is at least one background work process available on that server. Otherwise, the batch job that performs network statistics snapshots is delayed until a background work process becomes available. As a consequence, the intervals grow arbitrarily. The name of the batch jobs is DB2 network statistics. They are created under the name of the user who starts the collection process. You can check batch jobs in transaction SM37. There should never be more than one network statistics batch job active.

To monitor the collection process, which takes place in background processing, select the checkbox ABAP trace (SM21)? prior to starting the collection of network statistics. The trace information is written to the SAP System Log (transaction SM21).

**DB2 Connect Network Statistics Panel**

The statistical information on network performance is displayed in a panel that can be reached from the DB2 Connect Diagnostics panel by clicking on Display statistics:
39.1 Network Turnaround Time

The DB2 Connect Network Performance Statistics provide several statistics derived from the network turnaround time. This is illustrated in the figure below.

The network turnaround time (DN) is defined as the sum of the request transmission time and the response transmission time.

The request transmission time is defined to start when the first data packet of a request leaves the client (T1) and end when the server receives the last data packet of that request (T2).

The response transmission time is defined to start when the first data packet of a response leaves the server (T3) and to end when the client receives the last data packet of that response (T4).

Since the clocks on the client and server sides are not synchronized, only the time differences on each side can be utilized: DC is the time difference (T4-T1) on the client side, DS is the time difference (T3-T2) on the server side, and the network turnaround time (DN) is DC-DS.

This means that any processing time on the database server, such as within DB2, is not part of the network turnaround time.

Network turnaround time may include several traversals of the network, such as when the request or response is split into multiple packets, or in the case of retries (in which case one or more TCP/IP acknowledgement timeouts are also included). It also includes the processing time within the TCP/IP protocol stacks on the application server side and the database server side, as well as dispatching delays on both sides.
39.2 Statistic Counters

For each request/response pair, DB2 Connect measures the network turnaround time and maintains the timers and statistics in transmission groups. The grouping of statements is accomplished according to the number of transmissions that are required for a statement:

- 2 transmissions
- 3 to 7 transmissions
- 8 to 15 transmissions
- 16 to 64 transmissions
- more than 64 transmissions

Once the time interval that is specified for the statistics interval has elapsed, the network statistics batch job performs a snapshot of the statistics counters and writes them to table DB2NETSTATS. They are then reset to 0. At this point, the new statistics are immediately available in transaction ST04 for analysis.

The following values are shown in the initial network statistics panel (Figure 2):

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application server</td>
<td>Starting time of the sampling interval.</td>
</tr>
<tr>
<td>Date</td>
<td>Date of the network statistics snapshot.</td>
</tr>
<tr>
<td>Time</td>
<td>Time of the network statistics snapshot.</td>
</tr>
<tr>
<td>Transm./SQL</td>
<td>Transmissions per SQL statement</td>
</tr>
<tr>
<td>Avg. NTT (ms)</td>
<td>Average network turnaround time in milliseconds.</td>
</tr>
<tr>
<td>SQL stmts</td>
<td>Number of SQL statements</td>
</tr>
<tr>
<td>SQL chains</td>
<td>Number of SQL chains. To speed up processing, DB2 Connect chains multiple SQL statements when appropriate.</td>
</tr>
<tr>
<td>Time &lt; 1 ms</td>
<td>SQL chains with elapsed time &lt; 1 millisecond</td>
</tr>
<tr>
<td>Time &gt; 500 ms</td>
<td>SQL chains with elapsed time &gt; 500 milliseconds</td>
</tr>
<tr>
<td>Avg. request (KB)</td>
<td>Average request size in KB</td>
</tr>
<tr>
<td>Avg. response (KB)</td>
<td>Average response size in KB</td>
</tr>
</tbody>
</table>

By selecting a snapshot from the list and clicking on “Details”, you can analyze a snapshot in more detail:
The detail view shows the following SQL chain distributions:

- Time distribution (network turnaround time)
- Request size distribution
- Response size distribution
39.3 Measuring Network Turnaround Time

Use SAP transaction ST04 to measure network performance:

- To get a rough feeling for your network's performance, use the function DB2 Ping. This function sends and receives data packets across the network the same way that your transactions are running, and reports the time (in milliseconds) spent on the network. To guarantee that the returned number represents an average, a sample of 10 send/receive cycles is performed and the mean value is returned. The following graphic depicts the DB2 ping process and contrasts it with the standard TCP/IP ping:

![Diagram showing DB2 Ping process compared to standard TCP/IP ping]

- To become more confident about your network performance, use the function Network Statistics. Run the statistics for at least one day and analyze them in ST04. The important counters are the average response times for the packet sizes being sent.
39.4 CLI Tracing

From the DB2 Connect Diagnostics panel you can also collect CLI trace information and display CLI traces. The performance impact of the CLI trace can be significant. Therefore, do not turn it on if not absolutely needed.

Error! Objects cannot be created from editing field codes.

The CLI trace can be turned on either for all work processes or for a single work process. If you start the trace by clicking the button Start trace, the SAP profile parameter `dbs/db2/cli_trace` is set accordingly. A value of 1 causes CLI trace data to be collected. A value of 0 stops the CLI trace. If the CLI trace is collected for a single work process only, the number of the work process is concatenated with the string 1. For example, if the CLI trace is turned on for work process 12 only, `dbs/db2/cli_trace` would be set to 1,12.

The profile parameter `dbs/db2/cli_trace` can also be changed online in transaction RZ11.

The CLI trace data is written to a dedicated file per work process in the SAP work directory. The naming convention is `p<operating system process ID>t1.cli`.

To view a CLI trace file, select a single work process in the Work process pull down menu and select Show trace.
40 Tables and Indexes Monitoring
40.1 Requirements

How Data Is Collected

The tables and indexes monitor collects status information about the tables, tablespaces and indexes of the SAP system or (with MCOD) of all SAP systems within an MCOD landscape. It gets some of this information directly from the DB2 catalog, for example the number of tables. However, some information has to be collected at the operating system level, for example all space information.

Therefore, some of the information presented by the monitor is obtained using z/OS utilities (LISTCAT function of the IDCAMS program). To do this, a JCL job that performs a LISTCAT over the whole subsystem is generated and submitted from within the SAP System. It may take some time until the job has been finished and the output of the JCL job has been transferred to the SAP System. The stored procedures DSNACCJS, DSNACCJQ, DSNACCJF and DSNACCJP are used for execution of the JCL jobs.

Requirements for Stored Procedures

See the prerequisites in JES Interface [Page 246].

Requirements for JCL Job Submission

The JCL job is generated from a job template by substituting given profile parameter values for certain variables. You must enter these profile parameter values before z/OS jobs can be submitted from the SAP System.

For the tables and indexes monitor, you must enter the JES hold output class before z/OS jobs can be submitted from the SAP System. Use the JES Interface, function Profile Parameter, to enter this value.

We recommend that you also maintain all other profile parameters. See JES Interface for more information.

Requirements for SAP Objects

Even if not explicitly mentioned, it is assumed that all SAP objects have been created by <SCHEMA>. 

### 40.2 Tables and Indexes Monitor

#### Where the Displayed Information Comes From - The Refresh Button

The information provided in the tables and indexes monitor is not collected by the monitor itself. The initial screen is empty. You must choose the Refresh button at least once to be able to display data.

To display current values when needed, you can refresh the initial screen on request. Choosing *Refresh* immediately releases program RSDB2T2M as a background job. This job invokes the previously mentioned *LISTCAT* function.

This feature is not to be used for regular updates of the displayed information. Instead the program should be scheduled periodically to provide reliable history information.

By default, the *LISTCAT* output file is deleted from the application server after data analysis. You can prevent the file from being deleted by entering **KEEP** in the command field on the initial screen of transaction DB02. This affects only the next *Refresh* run.

#### Updating Data

Use the DBA Planning Calendar to update the data. For more information, see the DBA Planning Calendar [Page 266] in *Database Administration*.

#### Initial Screen

To access the *Database performance: tables and indexes* initial screen, choose one of the following:

- Choose **Administration** → **System administration** → **Monitor** → **Performance** → **Database** → **Tables/Indexes**
- Call transaction DB02.
- Call transaction DB2 and choose **Storage Mgt.** → **Tables and Indexes**.
Database Performance: Tables and Indexes

Information Displayed on the Initial Screen

SAP System:

- **DB system**
  
  DB2 indicates that you are running the SAP System on DB2 UDB for z/OS.

- **SAP system IDs**
  
  Name of the SAP systems residing in the database. The graphic shows an MCOD system (Multiple Components in One Database) with three systems named M5C, M5D and KDF in the same subsystem. If you do not use MCOD, there will still be only one name shown.

  Under rare circumstances, there will be one or more entries written with the following syntax: SCHEMA=schema-name. One example is SCHEMA=SAPR3. The systems, to whom the corresponding entries belong, did not broadcast their existence to the other systems yet. You should log on to the corresponding system and call transaction DB2J. The entry should be updated automatically and is visible with the correct name the next time you display the Database performance: tables and indexes initial screen. For more information concerning this broadcast, see **MCOD and CCMS [Page 47]** in Basic Operations → Other Operations and Considerations.
• **Date/time of this analysis**
  
  Date and time of the last update of the initial screen, that is, date and time of the last start of program RSDB2T2M.

**Tablespaces:**

• **Total number**
  
  Number of tablespaces in the subsystem.

• **Total space (KB) (more exactly, high allocated RBA)**
  
  Allocated space of the VSAM data sets representing tablespaces, summed over all tablespaces created by subsystem.

• **Total free space (KB) (more exactly, high allocated RBA - high used RBA)**
  
  Difference of allocated and used space of the VSAM data sets representing tablespaces, summed over all tablespaces in the subsystem.

  The total amount of storage needed is larger due to unusable space on the tracks.

Indexes:

• **Total number**
  
  Number of indexes in the subsystem.

• **Total space (KB) (more exactly, high allocated RBA)**
  
  Allocated space of the VSAM data sets representing indexspaces, summed over all indexspaces in the subsystem.

• **Total free space (KB) (more exactly, high allocated RBA-high used RBA)**
  
  Difference of allocated and used space of the VSAM data sets representing indexspaces, summed over all indexspaces in the subsystem.

• **Missing from database in <SAP system ID>**
  
  Number of indexes defined in the ABAP Dictionary, but non-existing in the database. Exceptions registered in table DBCHK are considered.

• **Missing from ABAP Dictionary in <SAP system ID>**
  
  Number of indexes existing on the database and created by <SCHEMA> but not defined in the ABAP Dictionary. Exceptions registered in table DBCHK are considered.

The following information is listed under **Tables:**

• **Total number**
  
  Number of tables created in the subsystem.

• **Total space (KB)**
  
  Used space of the VSAM data sets representing tablespaces, summed over all tablespaces created in the subsystem.

• **Missing from database in <SAP system ID>**
  
  Number of tables defined in the ABAP Dictionary, but non-existing in the database. Exceptions registered in table DBCHK are considered.

• **Missing from ABAP Dictionary in <SAP system ID>**
  
  Number of tables existing on the database and created by <SCHEMA> but not defined in the ABAP Dictionary. Exceptions registered in table DBCHK are considered.
40.3 Checking Consistency

To check the consistency between the database and the ABAP Dictionary or to display more detailed information about database objects, choose Checks in the Database Performance: Tables and Indexes screen.

You can choose between the following options:

- **Missing indexes**
  Provides a check on existence and uniqueness of indexes in the ABAP Dictionary as well as in the database. In this respect this is a subset of the Database ↔ Dictionary check, but it automatically rechecks and presents up-to-date information. This recheck is faster than the recheck in the Database ↔ Dictionary check.

- **Database ↔ Dictionary**
  Performs a consistency check between the database and the ABAP Dictionary concerning tables, indexes and views. You have the choice between displaying the information of a prior check, or doing a lengthy recheck. It is possible to create items that are missing from the database.

- **SAP System**
  Calls transaction SICK, which checks the consistency between the SAP kernel and the database concerning items, such as release number, character set and some critical structure definitions like SYST, T100, TSTC, TDCT or TFDIR.

The above checks can also be accessed by calling transaction DB2 and choosing Checks/Settings.
40.4 Database Space Statistics

To display the database space statistics, choose Space statistics under SAP System in the Database Performance: Tables and Indexes screen.

This function provides a history of database space information as shown on the initial screen. You can choose between a daily, weekly or monthly view of the data.
40.5 Detailed Information on Tablespaces

To display detailed information on tablespaces, choose Detailed analysis under Tablespaces in the Database Performance: Tables and Indexes screen.

A list of all tablespaces in the subsystem is provided as well as size information on any one of them. The number of tablespaces displayed is limited to the set maximum number of hits in the Data Browser. To enter the selection criteria for the tablespace list, choose DB Analysis → New Selection. You can change the number of displayed tablespaces under Data Browser settings. All settings entered in this screen are user-specific and are permanently valid in the Data Browser. Double-click a line of the list for more detailed information on the selected tablespace. You can choose the following options:

- **Tables and indexes**
  A list of all tables contained in the selected tablespace is generated. Double-click a line to display detailed information on the table and its indexes.

- **Detailed analysis**
  Information on the selected tablespace as provided by the DB2 catalog tables sysibm.sysdatabase and sysibm.sysdatabasepart is shown. In addition, there is storage information available (for example, number of extents, primary and secondary quantity, allocated and free space, list of volumes) as returned by the LISTCAT function. Since the catalog entry giving the number of pages is updated by the RUNSTATS utility, date and time of the last RUNSTATS are also provided.

- **History**
  The size of the selected tablespace is listed in intervals of days, weeks or months, as desired.

To display information on the dynamics of tablespace sizes, choose Space statistics under Tablespaces in the Database Performance: Tables and Indexes screen. You can choose between day, week or month as the unit of time. Tablespaces only appear in the list if they were subject to any changes in allocated or free space, or the number of extents. The space history for a selected tablespace can be accessed by double-clicking the list.
40.6 Detailed Information on Indexes

To display detailed information on indexes, choose *Detailed analysis* under *Indexes* in the *Database Performance: Tables and Indexes* screen.

A list of all indexes in the subsystem is provided as well as size information on any one of them. The number of indexes displayed is limited to the set maximum number of hits in the Data Browser. To enter the selection criteria for the indexes list, choose *DB Analysis → New Selection*. You can change the number of displayed indexes under Data Browser settings. All settings entered in this screen are user-specific and are permanently valid in the Data Browser. Double-click a line of the list for more detailed information on the selected index.

- *Detailed analysis*

  Information on the selected index as provided by the DB2 catalog tables `sysibm.sysindex` and `sysibm.sysindexpart` is shown. In addition, there is storage information available (for example, number of extents, primary and secondary quantity, allocated and free space, list of volumes) as returned by the `LISTCAT` function. Since the catalog entry giving the number of pages is updated by the `RUNSTATS` utility, date and time of the last `RUNSTATS` are also provided.

To display a list of missing indexes, that is, indexes defined in the ABAP Dictionary that do not exist on the database and vice versa, choose *Missing indexes*. 
40.7 Detailed Information on Tables

To display detailed information on tables, choose Detailed analysis under Tables in the Database Performance: Tables and Indexes screen.

This function generates a list of tables in the subsystem. A selection screen gives you the option of limiting the range to tables, table spaces and databases matching a given pattern. To display more information on the selected table, double-click the list.

You can choose the following options:

- **Detailed analysis**
  
  Returns information on the table and all its indexes as provided by the DB2 catalog tables `sysibm.systables` and `sysibm.sysindexes`.
  
  The size of the table is approximated by the product of the number of pages containing data and the pagesize. Since the catalog entry giving the number of pages is updated by the `RUNSTATS` utility, date and time of the last `RUNSTATS` are also provided.

- **Structure**
  
  Shows type and length of all columns of the selected table as defined in the ABAP Dictionary and created on the database, respectively.
40.8 Extent Monitor

You can access the extent monitor by choosing one of the following:

1. Choose Extent Monitor in the initial screen of the table and index monitor
2. Call transaction DB2 and choose Extents.

If you have enabled the DB2 feature to optimize extent sizing by setting ZPARM MGEXTSZ to YES (which is highly recommended), then the extent monitor is not relevant because DB2 ensures that the maximum number of extents is not reached.

Extent Monitor Data

The threshold levels in the extent monitor are taken from the DB Alert settings. By default, the low threshold is 25 and the high threshold is 80.

The extent monitor shows a table of all tablespaces and indexes that have more extents than specified under low threshold. (See also “DB Alert Settings” in DB Alert Router [Page 172]).

This alert data in this table is collected from three sources:

- LISTCAT Job
  Whenever a LISTCAT job is run, the number of extents is evaluated and written to the extent monitor table if the threshold is exceeded.

- COLLECT_DAILY Job
  During the COLLECT_DAILY Job, the stored procedure DSNACCOR is invoked with the same thresholds as configured in the DB Alert settings. When any tablespace or index exceeds these thresholds, DSNACCOR generates an alert that is not only shown in transaction RZ20, but also placed into the extent monitor table.

- RFCOSCOL as DB Alert Router
  In order to get the very latest extent information, RFCOSCOL in its role as database alert router is exploited (see Database Performance Monitor [Page 157]). It sends specific exception events, such as data set extents to the SAP System. These events are incorporated into the table shown in the extent monitor.

The objects are split into two levels of urgency, according to the number of extents:

- **Severity 1:**
  Number of extents > high threshold
  If the number of extents exceeds the high threshold, a reorganization will be required.

- **Severity 2:**
  high threshold > number of extents > low threshold
  The object is marked for observation. You may have to increase the secondary quantity

The objects are specified using their name and the database (for tablespaces, type TS) or the creator (for indexes, type IX). For partitioned tablespaces or indexes the affected partition is identified, otherwise the Partition column contains a 0. For tablespaces or indexes that have several data sets, the data set in question is identified in the Dataset column by its last qualifier.
Changing the Secondary Quantity

Only users with the authorization profile S_DB2_DBADM (DB2 UDB for z/OS database administrator) can change the size of the secondary quantity from the extent monitor. Page sets that belong to a different SAP system cannot be manipulated in that way.

The Current SECQTY column shows the current size of the secondary quantity in KB. To change this value, select the corresponding lines in the table, enter the required size in the field New SECQTY, and choose Alter Space. If successful, the column Current SECQTY shows the new (changed) value, the column Last ALTER contains the date of the last change (directly after the last change, that is, the current date). The column Extents before shows the number of extents before the last change (initial: 0), so that you can see how quickly the tablespace or index occupies new extents after ALTER.
40.9 Volume Free Space

You can use this function to see how much freespace there still is on a volume.

1. Call transaction DB02.
2. Choose Volume freespace.
3. Enter the required volume in the following dialog:

An z/OS job is generated and executed under your user ID. This obtains the information directly from the z/OS host.

See the information on the SAP system user ID and TSO password in Requirements [1].

After executing the job, whose runtime depends on the size of the volume, you will see the freespace on the volume in cylinders, as well as the number of free data set control blocks (DSCBS) on the volume.
40.10 Troubleshooting for DB2 Refresh

The following provides information on how to deal with problems encountered with the REFRESH job.

Example Problem 1

Symptom
The refresh job has been executed, some values are displayed, but all values for the space information are 0.

Cause
The JCL job that gathers the space information has probably been aborted.
If you do not specify any profile parameters, the submission of the JCL job is aborted and you find an entry in the system log. The system log entry will also be written if any errors occur in the JCL submission service. Since there is no space information available in this case, you get the initial values for the tablespace sizes, that is, 0.

Solution
Check the system log and make sure that all your profile parameters have been maintained with the JES Interface (transaction DB2J).

Example Problem 2

Symptom
The refresh job fails. The return code of the LISTCAT job is 8.

Cause
At least one of the data sets processed by the LISTCAT job has been migrated by HSM.

Solution
Change the JCL skeleton LISTCAT_LEVEL in the JES interface (transaction DB2J). Replace the parameter ALLOC with ALL. This enables LISTCAT to cope with migrated data sets (NON-VSAM).
However, the size of the job output more than doubles with this parameter. If you are sure that your job has failed due to migrated data sets, you could also consider using maxcc=0 and keeping ALLOC.

Example Problem 3

Symptom
The DB02 refresh job executes without errors, that is, transferring the job to the host runs smoothly, as does the execution of the job. Nevertheless, no size specifications are delivered back about tablespaces and index spaces.

Cause
The MSGLEVEL=(1,1) specification or something equivalent is missing in the job header of the JCL Interface (Transaction DB2J). This means that the actual size information from the LISTCAT job is not collected. The result file on the application server is, therefore, only a few KB large.
Solution

Add the MSGLEVEL=(1,1) specification or something equivalent into the job header. This will enable the LISTCAT job to also return the size information for page sets.
41 CCMS Monitor Set

To access the CCMS monitor set, choose one of the following transactions:

- Call transaction RZ20.
- Call transaction DB02 and choose CCMS monitor set.

You can access the monitors using the standard SAP CCMS Monitor Templates or using your own monitor that you have customized to display selected elements of the monitor tree structure. For more information on the CCMS monitor set, for example, for customization, see the SAP Online Documentation.

The following tree elements are used under Database → <SID> → DB2 Universal Database for z/OS:

- Backup/restore → backup
- Tablespace with no backup
- Performance → runstats
  Tables needing a RUNSTATS
- Performance → long running units of recovery
  See DB Alert Router [Page 172].
- R/3 consistency → DB2 installation parameters
  DB2 installation parameters that differ from the required or recommended values.
  See DB System Check [Page 192].
- Space management → extents
  Tables or indexes that exceed the extent thresholds. See DB Alert Router [Page 172].
- Space management → reorganization → reorg tablespace, Space management → reorganization → reorg index
  Tablespace and indexes needing reorganization
- Health → Object in restrict mode
  The database objects, which are in some restricted state, are listed here in the format DBName. TSName for tablespace and DB Name.IndexspaceName for indexes.
- Health → active log
  Active log shortages. See DB Alert Router [Page 172].
- Health → deadlocks
  See DB Alert Router [Page 172].
- Health → timeouts
  See DB Alert Router [Page 172].
- Backup/restore → backup
  Tablespace with no backup

The following nodes always exist in the CCMS monitor set. Do not delete them.

- R/3 consistency → collecting bootstrap → daily (DO NOT DEL)
Program that runs an automatic daily check. The tool checks whether a page set requires RUNSTATS, REORG or BACKUP is needed or whether there are table spaces that do not have a backup.

- **R/3 consistency → collecting bootstrap → hourly (DO NOT DEL)**
  - Program that runs every hour. The tool checks whether the objects are in restrict mode.

- **R/3 consistency → collecting bootstrap → 5 minutes (DO NOT DEL)**
  - Program that runs every five minutes. It updates database alerts that have been reported by the database alert router (RFCOSCOL).

Be aware that each of these nodes collects at most 20 single alerts. If any of the nodes exceeds 20 alerts, there will be one collective alert displayed instead, indicating that there are many alerts. You can display all these alerts using the analysis tool, for example, by double-clicking on the collective alert.

### Real Time Statistics

Real Time Statistics (RTS) are collected for easy and inexpensive detection of database objects needing some DBA intervention. The collected data include a large number of statistics values such as the number of rows inserted, deleted, changed since the last RUNSTATS, REORG or COPY. DB2 always generates in-memory statistics for each table space and index space. When the statistics is to be externalized, DB2 examines the in-memory data, calculates the new totals, updates the new real-time statistic tables with the new totals and resets the in-memory data. This process is an asynchronous task. The tables where the real-time statistics is stored are SYSIBM.TABLESPACESTATS and SYSIBM.INDEXSPACESTATS in the DSNRTSDB.DSNRTSTS table space.

The data collected in the RTS is examined by a stored procedure named DSNACCOR, which recommends running a utility based on complex calculations with this data. The SAP system utilizes DSNACCOR instead of keeping track of changes itself. Besides accurate recommendations, the benefit of DSNACCOR is that even objects not belonging to SAP or belonging to different SAP systems in an MCOD environment (see Basic Operations [Page 18], section “MCOD and CCMS”) are considered in the calculations. Be aware that you will get a larger number of alerts.

It is generally recommended to use DSNACCOR to determine the database objects that need maintenance in SAP environments.

The steps necessary for correctly installing RTS and DSNACCOR are described in Installing Real Time Statistics (RTS) Using WLM [Page 69].

For more information on both subjects, see the IBM documentation DB2 UDB for z/OS V8 Utility Guide and Reference.

Using DSNACCOR, you can tune the thresholds for recommendation yourself:

4. Call transaction ST04 or DB2.
5. Choose DB Alert Settings.
6. Choose Change. This enables you to change all the thresholds DSNACCOR uses. Under normal circumstances, the default settings should be sufficient. For more information on each of the values and its role in the calculation for recommendations, refer to the online help and to the IBM documentation DB2 UDB for z/OS V8 Utility Guide and Reference.

**RUNSTATS needed**

This checks whether update statistics (RUNSTATS) should be recommended for any of the tablespaces. Tablespaces that qualify for update statistics are displayed and entered into table DBSTATC at each run of RUNSTATS needed with TOBDO flag set to X.

Alerts are only raised if the **Active** flag in the DBSTATC table is **not** set to N for the tablespaces concerned. If only a few tablespaces (less than 20) are affected, an alert is shown for each individual tablespace. If more tablespaces are affected, a single collective alert is displayed to inform the user that RUNSTATS is recommended for a number of tablespaces. You can display these tablespaces using the analysis tool.

If you want to exclude tables from this procedure, you can set the **Active** flag to N in the DBSTATC table in maintenance transaction SM30. In other words, an alert is only raised when the **Active** flag in the DBSTATC table is **not** set to N for a tablespace.

With DB2 V7, DSNACCOR also provides a possibility to prevent alerts on a tablespace or indexspace via the exception table. We also support this feature, and so you have the choice which one of the tables DBSTATC or exception table to use. We recommend using the exception table, since it will prevent treatment also for objects not belonging to this SAP system and for any of the other utilities. For installation of the DSNACCOR exception table, refer to the SAP installation documentation.
You can specify any tablespace or indexspace by inserting its name into column `NAME` and its database name into column `DBNAME`. You can insert any string into column `QUERYTYPE` or you can leave it empty.

The SAP system recognizes the keywords `RUNSTATS`, `REORG` and `COPY` for exclusion of the respective utility. Take care to separate each keyword from the others with a space. For `REORG` you may specify single partitions, for which `REORG` is excepted by specifying `REORG=n1, n2,...,nm` where `n` is a partition number. Make sure there is no space between the numbers and the comma. The sequence of the numbers is irrelevant.

You want to exclude partitioned tablespace `TESTDB.TESTTS` from `RUNSTATS`, `COPY` and also from `REORG` on partitions 1, 3, 7 and 13. Issue the following SQL command in SPUFI:

```
INSERT INTO DSNACC.EXCEPT_TBL(dbname, name, querytype)
VALUES('TESTDB', 'TESTTS', 'RUNSTATS REORG=1,3,13,7 COPY')
```

The SAP system will automatically fill the exception table with a predefined set of tablespaces on which `RUNSTATS` is not recommended and the corresponding indexspaces. For more information, see the section “When `RUNSTATS` is due” in Performance Tuning Considerations [Page 96]. If you insert other tablespaces, the indexspaces of the indexes on tables in that tablespace (associated indexspaces) are added automatically with the same `QUERYTYPE`. The SAP system will only add them if they are not yet inserted. This means that if you change the `QUERYTYPE` column of a tablespace in the exception table, the entry of the associated indexspaces will not be automatically updated. You must do this manually. The corresponding tablespace of the associated indexspaces can be identified using the information in the columns `ASSOCDB` and `ASSOCTS`.

You want to update the recommendation for tablespace `TESTDB.TESTS` (see above). Now it should be excluded from `COPY` and `REORG` on partitions 1, 2, 5, and 10. Its associated indexspaces have to be updated accordingly. Issue the following SQL command in SPUFI:

```
UPDATE DSNACC.EXCEPT_TBL SET querytype = 'COPY REORG=1, 5, 2, 10'
WHERE ( dbname = 'TESTDB' and name  = 'TESTTS' )
   OR (assocdb = 'TESTDB' and assocts  = 'TESTTS' );
COMMIT;
```

When leaving `QUERYTYPE` empty or inserting a string containing no keywords, the SAP system treats this as if keywords `RUNSTATS` and `REORG` were defined.

### REORG needed

The `RSDBA_COLLECT_DAILY` program, that runs once every day with default settings, checks which tablespaces and indexes need to be reorganized by invoking DSNACCOR. The tablespaces to be reorganized are entered in table `DB2REOTS` and the indexes are entered in table `DB2REOIX`. Indexes in a tablespace that is to be reorganized are not entered in table `DB2REOIX` because these indexes are automatically reorganized with `REORG TABLESPACE`.

### Reorganization Information Access

You can access reorganization information in the `DB2REOTS` for tablespaces or the `DB2REOIX` table for indexes. To access these tables, call transaction SE16 and enter the name of the corresponding table.
In the NOREORG column you can determine whether a tablespace or index in general should be excluded from REORG. If you enter N (for NO) for a tablespace or index, no REORG is executed on this tablespace or index. If you enter Y (for YES) a REORG is executed. Y is the default value.

You can achieve the same result by inserting the name of the tablespace resp. indexspace into DSNACCOR exception table. For installation and use of the DSNACCOR exception table, see the IBM documentation *DB2 for z/OS V8 Utility Guide and Reference*.

The tablespaces or indexes that need to be reorganized are set to Y (for YES) in the NEEDREORG column. Once a REORG has been executed on the tablespace or index, the value is set to N (for NO).

If you schedule the REORG from outside the SAP system, the value is not set to N (for NO). It will only be reset after the next run of the RSDBA_COLLECT_DAILY program.

The CCMS monitor set (transaction RZ20) lists the database name and the tablespace name for the tablespace, and the creator and the index name for the index when there are fewer than 20 tablespaces or indexes. If more than 20 tablespaces or indexes are affected, a single collective alert is displayed to inform the user that REORG is recommended for a number of tablespaces or indexes. You can display them using the analysis tool.

### No Backup

The RSDBA_COLLECT_DAILY program checks whether there are any tablespaces or indexes defined with COPY YES in the subsystem without a backup using the stored procedure DSNACCOR.

You can display the result set using the analysis tool, for example by double-clicking the alert. See the section on the CCMS Backup Monitor for more information.

### Restrict Mode

The RSDBA_COLLECT_HOURLY program checks whether there are any tablespaces or indexspaces in the subsystem in *Restrict* mode using the stored procedure DSNACCOR.

You can display the result set using the analysis tool, for example, by double-clicking the alert.
42 Switching on WLM Management

This section only applies if your application server is on z/OS.

The SAP profile parameter `rdisp/prio/wlm/enabled` or the environment variable `rdisp_prio_wlm_enabled` control whether the SAP Web Application Server is managed by WLM.

- The SAP Web Application Server processes are WLM managed if the values are set to 1:
  - `rdisp/prio/wlm/enabled = 1`
  - `rdisp_prio_wlm_enabled = 1`

- The SAP Web Application Server processes are not WLM managed if both the SAP profile parameter and the environment variable are not specified, which is the default, or if the value is set to 0:
  - `rdisp/prio/wlm/enabled = 0`

The following are syntax examples of SAP profile parameter `rdisp/prio/wlm/enabled` and the environment variable `rdisp_prio_wlm_enabled`:

**SAP profile parameter `rdisp/prio/wlm/enabled`:**

```
rdisp/prio/wlm/enabled = <number>
```

where `<number>` is an integer value of 0 to switch WLM off or 1 to switch WLM on. Also, if `<number>` is a non-zero value other than 1, WLM is switched on.

By default, `rdisp/prio/wlm/enabled` is not specified.

**Environment variable `rdisp_prio_wlm_enabled`:**

```
rdisp_prio_wlm_enabled = <number>
```

where `<number>` is an integer value of 0 to switch WLM off or of 1 to switch WLM on.

Also, if `<number>` is a non-zero value other than 1, WLM is switched on.

By default, `rdisp_prio_wlm_enabled` is not specified.

When switching on WLM for the SAP Web Application Server, you must have defined an appropriate WLM service definition on Z/OS for the subsystem type `SAP`.

**Tracing with WLM**

If WLM management is switched off for the SAP Web Application Server, the developer trace log files for the processes, for example:

```
dev_ms, dev_disp, dev_w<n> (<n> = number_of_work_process)
```

will contain the following lines (with trace level 1):

```
WLM call SAP_wlm_joinWorkUnit
WLM call SAP_wlm_joinWorkUnit returned with 4
```
Process should not be WLM managed

If WLM management is switched on for the SAP Web Application Server, the developer trace log files for the processes, for example:

```plaintext
dev_ms, dev_disp, dev_w<n> (<n> = number_of_work_process)
```

will contain the following lines (with trace level 1):

```plaintext
WLM call SAP_wlm_joinWorkUnit
SAP Application Server is WLM managed
WLM call SAP_wlm_joinWorkUnit returned with 0
Process joined successfully to WLM enclave with
WLM workload qualifier PK  = <sapsid>
WLM workload qualifier CI  = <sapsid>[1-3]<short_process_class>[4-6]
WLM workload qualifier TN  = <process_class>
WLM workload qualifier PR  = <sapsys>
WLM workload qualifier AI  = <hostname>[1-32]<sapsys>[33-34]
WLM workload qualifier UI  = <user_id>
```

where

- `<sapsid>` is the SAP system ID;
- `<short_process_class>` is one of DSP, MSG, ENQ, GWY, ICM, DIA, BTC, UPD, UP2, or SPO;
- `<process_class>` is one of DISPATCH, MSGSRV, GATEWAY, WEBSRV, DIALOG, BATCH, NQUEUE, UPDATE, UPDATE2 and SPOOL according to the type of work a process has to do;
- `<sapsys>` is the SAP system number;
- `<hostname>` is the name z/OS system where the SAP application server is running;
- `<user_id>` is the user ID running the SAP application server.

The ranges in brackets `[n-m]` indicate first and last position of the preceeding value.

If, in addition, WLM management is enabled on transaction level (SAP profile parameter `dbs/db2/use_wlm = 1` or environment variable `dbs_db2_use_wlm = 1`, see SAP Database Administration Guide: IBM DB2 Universal Database for z/OS, “Transaction Based DB2 Accounting and Workload Management” and “Environment Variables”) the developer trace log files for SAP work processes will also contain messages in the following form if the trace level is set to 3.

```plaintext
Process joined successfully to WLM enclave with
WLM workload qualifier PK  = <sapsid>
WLM workload qualifier CI  = <sapsid>[1-3]<short_process_class>[4-6]
WLM workload qualifier TN  = <process_class>
WLM workload qualifier PR  = <sapsys>
WLM workload qualifier AI  = <hostname><sapsys>
WLM workload qualifier UI  = <user_id>
```
WLM workload qualifier SPM = <end_user_id>[1-14]<sap_transaction_code>[17-34]

WLM workload qualifier PC = <abap_program_name>

• where <sapsid>, <short_process_class>, <process_class>, <sapsys>, <hostname> and <user_id> are as above; and

• <end_user_id> is the SAP end-user ID;

• <sap_transaction_code> is the SAP transaction code;

• <abap_program_name> is the ABAP program name.

<end_user_id>, <sap_transaction_code> and <abap_program_name> are transaction-related values.

The WLM workload qualifiers (PK), (CI), (TN), (PR), (AI), (UI) and for transactions (SPM) and (PC) indicate that the above mentioned values classify the enclave to which the process is joined and that these values have to be specified as WLM work qualifiers in the classification rule of a WLM service definition on z/OS for subsystem type SAP in order to associate this enclave to a WLM service class. A WLM service class defines the priority given to a process joined to an enclave associated with that WLM service class.

The WLM workload qualifiers (PK) (SAP system ID), (TN) (process class) and (PR) (SAP system number) are used only for downward compatibility to WLM service definitions made for SAP 6.20. See also the following sections in Planning Guide: z/OS Configuration for SAP on IBM DB2 Universal Database for z/OS “Workload Management (WLM)” and “Workload Management - Classification of SAP Workload in WLM”.

For more information on WLM management for SAP Web Application Server, WLM and enclaves, see the SAP documentation Planning Guide: z/OS Configuration for SAP on IBM DB2 UDB for z/OS, and the IBM documentation z/OS V1R4.0 MVS Planning: Workload Management, z/OS V1R4.0 MVS Programming: Workload Management Services – or the WLM information at www-1.ibm.com/servers/eserver/zseries/zos/wlm
Database Administration

This section covers the following topics:

- Backup and Recovery Options [Page 226]
- JES Interface [Page 246]
- DBA Planning Calendar [Page 266]
43 Backup and Recovery Options

Overview

Backup and recovery are processes that ensure that an SAP database can be re-instated with minimal disruption in operations after any kind of hardware, software, operational or environmental error or outage. These processes are a crucial factor in system availability and reliability and need careful assessment of their requirements, understanding of the processes, and skillful planning, development and use of the procedures.

Some of these processes are performed automatically by DB2 without any outside intervention, such as recovering the SAP database to its consistent state just before a z/OS system crash or an abnormal termination of DB2. Automatic recovery happens at the next DB2 start. For other processes, there are integrated tools in DB2, some of which can be used for building efficient and reliable backup and recovery procedures.

The backup and recovery procedures need to be set up by database administrators for each individual SAP database. Their characteristics depend on:

- System availability requirements
- SAP database change rate
- SAP database size
- Hardware and software resources

Basically, the higher the value of any of the first three factors, the more demanding the backup and recovery procedures.

An SAP database includes all the tablespaces, indexes and DB2 catalog and directory entries (practically all the catalog and directory tablespaces and indexes) that are pertinent to the SAP system. With MCOD, the set of SAP databases that share a DB2 subsystem or data sharing group form an MCOD landscape. From the viewpoint of operational and semantic integrity, an SAP database or MCOD landscape as a whole needs to be considered as a single unit of recovery. In other words, if a single SAP tablespace needs to be recovered to a point in time, all other SAP tablespaces and indexes need to be either also recovered to the same point in time, or already be at the state that they had at the time. A prior point in time recovery is an example when the entire SAP database or MCOD landscape might need to be recovered, while a recovery to the current state is an example when only damaged tablespaces and indexes must be recovered; the rest is already at the current level.

Some general recommendations relevant to backup and recovery in the SAP system environments are:

- Planning
  - Understand DB2 backup and recovery processes.
  - Assess the factors that influence the characteristics of the backup and recovery processes.
  - Develop procedures for all types of backup and recovery situations that might arise in your installation.
  - Perform these procedures at a time when they do not interfere with normal operations.
  - The recovery to a prior point in time is more complicated if you allow critical, non-SAP applications to be managed by the same DB2 subsystem (or DB2 data sharing group) as your SAP database. We strongly recommend dedicating a
DB2 subsystem or DB2 data sharing group to a single SAP database or MCOD landscape only.

- Operations
  - Use dual logging for the active log, archive log and bootstrap data sets.
  - Place the copies of the active log data sets and bootstrap data sets on different DASD volumes.
  - Do not discard archive logs that are more recent than the earliest consistent copy of any SAP tablespace, or even older than that, depending on your need for a prior point in time recovery.
  - Produce object-based copies (image copies of tablespaces) even if your main backup and recovery strategy relies on volume-based backups.
  - Consider producing multiple backup copies.
  - Keep back-level backups to extend the interval within which is possible to perform a prior point in time recovery. This also reduces the impact of backup data sets that might be damaged (inconsistent).
  - Avoid backing up tablespaces that contain inconsistent data (intra-page data inconsistency). Use the COPY utility option CHECKPAGE, and the DSN1COPY, DSN1CHKR and CHECK INDEX utilities to detect such inconsistencies in the users and catalog/directory tablespaces.
  - Make backups of the DB2 catalog and directory, especially after the activities that involve a lot of DDL, such as initial load, major transports, or release upgrade.
  - To speed up recovery, use more and larger active logs, consider archiving to disk, or be sure to have enough tape drives. Also, keep the buffer pools and log buffers at the values recommended for the SAP system (in other words, large). Recovery to the current state may be accelerated by using the DB2 Tracker feature.

The following sections present a summary of backup and recovery options that could be used in the SAP system environment. For more detailed information on backup and recovery, see

- **DB2 for z/OS Administration Guide** (“Operation and Recovery”) from IBM
- **DB2 for z/OS Utility Guide and Reference** from IBM
- **DB2 for z/OS Command Reference** from IBM
- **DB2 for z/OS Data Sharing: Planning and Administration** from IBM

You can also find information on the Split Mirror Backup/Recovery Solution at service.sap.com/split-mirror

This solution is a sophisticated method for generating very fast online backups, quick system recoveries and disaster recovery options.
43.1 Backup

The appropriate backup procedure is a key factor for data recovery. In simple terms, the recovery process consists of selecting a backup of a DB2 object or volume on which DB2 objects reside. All the changes (recorded in the log) are applied that occurred between the last log point at which all changed pages were reflected on disk before the backup was taken and the time the recovery is requested to. Theoretically, DB2 could recover objects from the log only, but this is not recommended.

The main characteristics of a backup procedure are:

- Frequency, in other words, how often an object is backed up
- Tools used to produce backups

The optimal backup procedure is a trade-off between its usage of resources (CPU, DASD, tapes) and increased contention with other concurrent activities in the system on one hand, and the speed of recovery on the other. The shorter the log apply phase, the faster the recovery. Frequent backups of all of the data pages containing committed rows only, provide the fastest recoveries. However, this has a higher cost in resources and impedes the concurrent activity in the system to an extent that might not be acceptable.

There are two main types of backup. In this documentation these are referred to as **online backup** and **offline backup**.
43.2 Online Backup

An online backup of an object tablespace, partition, index or volume) is a copy of the object during which continuous, concurrent read/write activity on the object is allowed. Therefore, except for some processor and DASD overhead, the online backup has no impact on the concurrent SAP activities.

As it can contain uncommitted data, such a backup alone is never enough for the object’s recovery. DB2 complements it with the log.

There are two types of online backup:

- Object-based
- Volume-based

Object-Based Online Backup

Object-based backups are image copies of DB2 tablespaces, partitions and selected indexes. The DB2 COPY utility with the SHRLEVEL (CHANGE) option is an efficient tool for creating this type of backup. The utility generates backups of DB2 tablespaces, partitions and indexes.

The following table lists other useful COPY options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FULL</td>
<td>Specifies whether a full or incremental image copy is to be created. The incremental image copy is a copy of only those data pages that have been changed since the last backup.</td>
</tr>
<tr>
<td>CHANGELIMIT</td>
<td>Allows you to let DB2 decide whether to take a full or incremental image copy, depending on the number of pages changed since the last image copy. Unless you regularly take full image copies, we recommend that you use the CHANGELIMIT specification where only one value is specified to avoid the situation where no full image copy exists.</td>
</tr>
<tr>
<td>COPYDDN</td>
<td>Allows you to create up to four identical copies of the tablespace.</td>
</tr>
<tr>
<td>RECOVERYDDN</td>
<td></td>
</tr>
<tr>
<td>CHECKPAGE</td>
<td>Checks page consistency within a page and makes subsequent DSN1COPY checks superfluous. Note that the checks performed by CHECK INDEX and DSN1CHKR (for tablespaces that contain internal links) are not done by CHECKPAGE.</td>
</tr>
<tr>
<td>PARALLEL</td>
<td>Can significantly improve COPY performance by parallel copying of objects specified in the same COPY control statement</td>
</tr>
</tbody>
</table>

Volume-Based Online Backup

Volume-based backups are copies of the volumes on which DB2 objects reside.

This type of backup requires the availability of disk subsystems capable of generating very fast volume copies, such as IBM ESS or RVA, EMC Symmetrix, HDS Lightning, or StorageTek SVA. Volume-based backups can be captured in two ways:

- Under the control of DB2 (this requires z/OS V1R5)
- Outside the control of DB2 while suspending DB2 log activity
Starting with z/OS V1R5, the DB2 utility **BACKUP SYSTEM** invokes DFSMShsm fast replication services to take volume-based backups of all volumes on which DB2 objects reside. This method provides for a very efficient and non-disruptive means of taking online backups of the entire SAP database. A prerequisite for this method is to have a disk subsystem that supports ESS FlashCopy.

The **BACKUP SYSTEM** utility takes either a full system backup or a data-only system backup. Using the utility option **FULL**, the backup contains both logs and data. The utility option **DATA ONLY** takes a backup of data only. Recovering a system to a prior point in time using the **RESTORE SYSTEM** utility only requires backing up data. If data and logs are copied, normal DB2 restart recovery can be used to recover the system to the point in time of the backup copy. This kind of recovery is very fast, but is limited to the points in time when backups were taken.

The procedure for making volume-based backups under the control of DB2 is just to invoke **BACKUP SYSTEM**.

Alternatively, volume-based backups can be taken outside the control of DB2. Coupled with the DB2 feature to suspend and resume DB2 logging activity on demand, this approach is also very efficient. However, it is more complex and requires suspending the DB2 log.

The procedure for making volume-based backups outside of DB2 is outlined below:

- **Suspend DB2 logging activity by issuing the DB2 command **SET LOG SUSPEND**

  At this time, DB2 initiates a system checkpoint (in non-data sharing environments only), writes any unwritten log buffers to DASD, updates the BSDS with the high-written RBA, and acquires the log-write latch to prevent any further log records from being created. This will prevent any further updates to the database. A highlighted message (**DSNJ372I**) is displayed to indicate that logging has been suspended. The scope for this command is single-subsystem only. This means that the command must be entered for each member when running in a data-sharing environment.

  Although the **SET LOG SUSPEND** command initiates a checkpoint, it does not wait for all the data pages to be written out. Because of this, you can improve the performance of subsequent recoveries by minimizing the number of data pages that are not written out at the time the disk volume copies are taken. This can be done by triggering a checkpoint five to ten minutes earlier. An on-demand checkpoint is triggered by issuing the DB2 command **SET LOG LOGLOAD(0)**.

- **Take disk volume copies of all the volumes containing DB2 user and system data**

  This step should be done quickly. This is possible by exploiting modern DASD subsystems. Otherwise, suspending update activity for too long can cause timing-related events, such as lock timeouts for IRLM diagnostic dumps when delays are detected. The system backup obtained by this can be used for starting DB2. This start will be performed in the same way as DB2 restart after an abnormal system termination. The inflight units of recovery will be rolled back, and this brings the SAP database to a consistent state.

- **Resume DB2 logging activity by issuing command **SET LOG RESUME**

  At this time, the log-write latch is released and updates to the database are resumed. The console message is then deleted.
43.3 Offline Backup

An offline backup is a copy of a quiesced object (no uncommitted units of recovery accessing the object) during which concurrent write activity on the object is not allowed. As a result, all the data is in a committed state, which means that this backup alone could be used for the object’s recovery, but only to the point in time at which the offline backup was taken.

Offline does not mean that either the DB2 subsystem or the SAP system must be offline. Concurrent read/write activity can continue on all other objects, as well as read-only on the object being backed up. However, for the objects that are heavily updated by many concurrent users and in that sense considered central to the SAP database, quiescing and preventing the write access during the offline backup can be extremely disruptive to the entire SAP system. This particularly applies if the method for taking the offline backup is not fast enough.

Like online backups, the offline backups can also be object-based or volume-based.

- **Object-based offline backup**
  
  The DB2 **COPY** utility with the SHRLEVEL (REFERENCE) option is an efficient tool for creating offline backups. Other useful **COPY** options are outlined in the table in the subsection “Object-based online backups” above, and these options are also applicable to offline backups. In addition, the most useful option for this purpose is **CONCURRENT**, which can significantly reduce the time during which the object is unavailable for write activity.

- **Volume-based offline backup**
  
  This type of backup is normally used in the context of creating the entire SAP database/MCOD landscape offline backup. They include all the tablespaces, indexes, DB2 catalog and directory, logs and other control objects taken during the time when no write activity is allowed in the system.

SAP database/MCOD landscape offline backups are a very restrictive way of backing up the data and should be used only where the SAP operations can afford such a downtime (for example, if there are some periods of no activity). On the other hand, the SAP database/MCOD landscape offline backups provide excellent prior point in time recovery targets.

DB2 and z/OS offer a number of ways to implement such a backup. The following examples demonstrate how the offline backup could be obtained:

**Offline Backup with COPY**

1. Quiesce the SAP database. In other words, make sure that no users with update intentions are present in the system. You can achieve this using the following commands:
   
   ```
   START DATABASE (*) ACCESS (RO)
   START DATABASE (DSNDB01) ACCESS (UT)
   START DATABASE (DSNDB06) ACCESS (UT)
   ```

2. Run the DB2 **COPY** utility:
   
   ```
   COPY TABLESPACE tspace1
   COPY TABLESPACE tspace2
   ```

   List all the SAP and DB2 catalog directory tablespaces.
Note that you can make the process more efficient by copying indexes and specifying multiple objects on the same COPY invocation.

COPY TABLESPACE DSNDB01.SYSUTIL
COPY TABLESPACE DSNDB06.SYSCOPY
COPY TABLESPACE DSNDB01.SYSLGRNX

To speed up the process, create a number of COPY jobs that can be run in parallel. You can take advantage of intra-COPY parallelism by specifying a list of objects to be copied in the same COPY invocation using the option PARALLEL. The elapsed time can be improved significantly if you group the tablespaces in a way that reduces DASD path contention.

Instead of using the COPY utility that operates on a tablespace and index level, you can make copies of all the volumes on which the SAP data resides by using the disk system fast backup capabilities (where available). Where these advanced features are not available, the DFSMSdss DUMP and COPY functions can be used instead.

3. Restart the DB2 subsystem and databases to allow normal access.

Offline Backup with Concurrent Copy

Another example of creating an offline backup of the SAP database uses CONCURRENT COPY:

COPY:

COPY TABLESPACE tspace1
TABLESPACE tspace2
.
.
. List all the SAP and DB2 catalog and directory tablespaces except SYSUTIL, SYSCOPY and SYSLGRNX
.
. Note that indexes can also be specified.
CONCURRENT

COPY TABLESPACE DSNDB01.SYSUTIL CONCURRENT
COPY TABLESPACE DSNDB06.SYSCOPY CONCURRENT
COPY TABLESPACE DSNDB01.SYSLGRNX CONCURRENT

This method does not need the separate quiesce and restart steps. The database activity will be quiesced and made available again automatically. However, be aware that the concurrent copy might fail in the phase of “hardening” the data (physical copy), in other words, after the logical copy has successfully completed and the SAP database is made available for read and write. In this case, the copy as a whole has not succeeded and the offline backup has not been created. After removing the cause of the problem, you need to repeat that process.
43.4 Backup Procedure Recommendations

For each SAP system installation, you should choose a backup procedure that is optimal to its particular needs and conditions. The following are recommendations for a backup procedure that should be used initially by all the SAP system installations and adjusted later according to the specific needs and conditions.

- After a successful SAP system installation or upgrade, migration, system copy or a prior point in time recovery, make an offline backup of the SAP database.
  
  Consider this a mandatory step.

- If the operations schedule allows, make occasional offline backups of the SAP database. If there is not enough time available to back up the entire SAP database, make offline backups of the heavily updated and critical tablespaces.
  
  This is not a mandatory step, but is nevertheless highly recommended. Offline backups are very efficient targets for a prior point in time recovery, especially if taken by a tool that copies both tablespaces and indexspaces.

- For any SAP system, catalog and directory tablespace regularly create online backups. How often the backup should be taken and whether it is full or incremental, depends on the tablespace's change rate. As a starting point, we recommend that you run the backup jobs every 1-2 days and specify \texttt{CHANGELIMIT(10)}. Once you categorize your tablespaces into heavily, moderately, or lightly updated tablespaces, you can change the frequency of their backups to daily, weekly, or monthly, respectively. You can use transaction ST10 to categorize tables by their access pattern. Alternatively, you may follow the recommendations of the stored procedure DSNACCOR and backup the objects as recommended.

  Producing regular online backups is a mandatory step of any backup procedure.

  With the \texttt{CHANGELIMIT} option you might end up with seldomly created full backups, which is not efficient from the recovery point of view. For this reason, make sure you have a full backup created periodically by specifying \texttt{FULL(YES)} or \texttt{CHANGELIMIT(0)}. Also consider running the \texttt{MERGECOPY} utility that consolidates a full backup and a number of incremental backups into a new, more recent full backup.

  The index recovery time can be significantly improved if the recovery is based on the index copy rather than on the index rebuild. In general, we do not recommend copying every SAP index, especially for small indexes. However, large indexes with a lower update rate should be considered for copying, optimally whenever the underlying tablespace is copied.

- Where available, use the disk system capabilities of generating fast volume copies, ideally using the \texttt{BACKUP SYSTEM} utility or combined with the \texttt{SET LOG SUSPEND/RESUME} commands to produce fast online system backups.

  If produced at the time when there are no long-running, non-committing jobs in the system, these backups provide for very efficient prior point in time recoveries. For the highest flexibility in recovery scenarios, separate DB2 data, logs BSDS, catalog/directory and ICF catalog on different sets of volumes that would allow independent restore for any of these objects. If using \texttt{BACKUP SYSTEM} logs, BSDS and associated ICF catalogs should reside in the same copy pool. Likewise, the DB2 data, catalog/directory and associated ICF catalogs should also reside in the same copy pool.

  It is very important to understand that these volume-based backups cannot replace the object-based backups (image copies produced by the \texttt{COPY} utility). There are cases where the recovery will require an image copy, and the volume-based backup would in
this case not be sufficient. Also, in the case of a single object recovery to currency, an image copy of that object is the best basis for such a recovery.

If volume-based backups are taken outside the control of DB2, i.e. without the BACKUP SYSTEM utility, then you must use the LOGONLY option of the RECOVER utility to recover data from the past.

A separate procedure should be developed for a disaster recovery. There are numerous ways and tools to implement the procedure. For example, the IBM Disk Storage Systems Web page at

www.storage.ibm.com

describes an advanced approach that involves Peer-to-Peer Remote Copy (PPRC) and fast volume backups. Disaster recovery is not described in detail in this documentation. For more information on disaster recovery, see the IBM documentation High Availability for SAP on zSeries Using Autonomic Computing Technologies.
43.5 CCMS Backup Monitor

The CCMS backup monitor is provided by the SAP System to analyze the object-based backups that have been made.

To access the initial screen, call transaction DB12 or choose:

Tools → Administration → Computing Center Management System → DB Administration → Backup logs

Backup Lists

To display several backup lists, choose the appropriate function under Backup information in the CCMS Backup Monitor - Overview of Backup Status screen. These lists can only contain tablespaces owned by SAP or SYSIBM and tablespaces in the directory database DSNDB01.

Backup Lists

<table>
<thead>
<tr>
<th>Function</th>
<th>Backup List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Backups</td>
<td>The last successful backup of each tablespace is shown. The date next to Last successful tablespace backups on the Overview of Backup Status screen refers to the oldest backup within this report. In the case of partitioned tablespaces, the last complete tablespace backup and the last single partition backups performed after that are listed. The backups are displayed with the database name, tablespace name, and timestamp. Information is only shown for tablespaces for which backups have been performed. For a description of the different columns, refer to the table below.</td>
</tr>
<tr>
<td>Backup overview</td>
<td>All backups stored in table SYSIBM.SYSCOPY are shown and terminated copies are highlighted. The backups are displayed with the database name, tablespace name, and timestamp.</td>
</tr>
<tr>
<td>Partial recovery</td>
<td>Partially recovered tablespaces are displayed. Partially recovered tablespaces have been recovered to a point in time using the RECOVER utility with the option TORBA or TOCOPY. If this only applies to a few tablespaces, the data may be inconsistent due to logical dependencies between tables. See the IBM documentation DB2 for z/OS Administration Guide and DB2 for z/OS Utility Guide and Reference for more information. The tablespaces are displayed with the database name, tablespace name, and timestamp.</td>
</tr>
<tr>
<td>Tablespaces without backup</td>
<td>This function displays all database and tablespaces that have no backup.</td>
</tr>
</tbody>
</table>

This list displays the backup history by reading the relevant data from the catalog table SYSIBM.SYSCOPY. Backups performed with tools that do not report to this table will not appear in the CCMS backup monitor.

Once a day, the objects requiring a backup are determined by stored procedure DSNACCOR, which uses real time statistics, and reported to the CCMS Monitor Set. For more information, see CCMS Monitor Set [Page 217] in Monitoring and Performance.
Instead of determining these objects again, the list *Tablespaces without backup* in the CCMS Backup Monitor uses that information. If an analysis is supposed to take place immediately, a background job can be scheduled by choosing *New analysis*.

**CCMS Backup Monitor - Last Successful Tablespace Backups - Show Details**

The columns displayed in the *CCMS Backup Monitor* are described in the Online Help.

For more information, see the SAP documentation *SAP on IBM DB2 UDB for z/OS: Database Administration Guide* and the IBM documentation *DB2 for z/OS SQL Reference*. 
43.6 Recovery

The DB2 utilities RECOVER and RESTORE SYSTEM are used to recover data.

RECOVER allows you to recover DB2 objects at the following levels:

- tablespaces
- indexes
- partitions
- individual data sets
- individual pages

RESTORE SYSTEM on the other hand considers whole DB2 subsystems as single entities and always recovers them completely.

RECOVER Utility

The RECOVER utility can recover data to the following states:

- State captured in a particular backup (the TOCOPY option)
- State at the time corresponding to a Relative Byte Address (the TORBA option) used in a non-data sharing environment or a Log Record Sequence Number (the TOLOGPOINT option) used in data sharing environments.
- Current state by not specifying any of the above options.

The RECOVER utility also has the LOGONLY option, which allows you to recover the data using the log only starting with a backup that is created outside of the DB2 control (for example, storage subsystem fast copy capabilities).

For improved performance, the RECOVER utility supports both inter-RECOVER and intra-RECOVER parallelism. The inter-RECOVER parallelism option is used to submit multiple RECOVER jobs concurrently. The intra-RECOVER parallelism option is even more efficient, because both the restore and log apply phase use parallelism. To exploit this option, specify multiple objects on the same RECOVER execution. The option PARALLEL is used to request parallelism in the restore phase. The log apply phase will be parallelized depending on the amount of storage allocated for the process. The value is given in the system parameter LOGAPSTG. Its default value is 100 MB which should be fine for most point in time recoveries.

Indexes can be recovered either by rebuilding them (using the REBUILD utility) or recovering them (using the RECOVER utility provided that the index is defined with the COPY YES option).

RESTORE SYSTEM Utility

The RESTORE SYSTEM utility recovers a DB2 subsystem to a prior point in time. First, it restores volume copies that have been produced by the BACKUP SYSTEM utility. Then, it automatically applies the log to recover to an arbitrary point in time. The DSNJU003 (Change Log Inventory) utility with the CRESTART SYSPITR option allows you to specify the point in time to which RESTORE SYSTEM recovers. It creates a conditional restart control record (CRCR), which truncates logs for system point in time recovery.

The option LOGONLY of RESTORE SYSTEM specifies that volumes have already been restored and that only log records are applied.

To recover a DB2 subsystem or data sharing group to the point in time at which a BACKUP SYSTEM FULL copy was taken, use HSM RECOVER * COPYPOOL(cpname) GEN(gen) to restore the database copy pool and the log copy pool and restart DB2.

Depending on the time to which the data should be recovered, there are two types of recovery: to the current state or to a prior point in time.
43.7 Recovery to the Current State

A recovery to the current state is generally less demanding and usually needed more often than a prior point in time recovery.

A typical example of current state recovery is a DASD volume failure that resulted in a loss of all or some of the data on the volume. The procedure in this case is to find out which tablespaces and indexes had resided on the volume and recover only these tablespaces and indexes, or even only the partitions or individual data sets that are affected. The rest of the system is already at the current state (from the operational and semantic integrity viewpoint) and need not be recovered.

For current state recovery, backups known to DB2 (taken by the COPY utility) or those unregistered in DB2 (such as the volume-based backups) can be used. The choice depends on the recovery case. For example, if the entire volume needs to be replaced, a volume-based backup as the basis for the RECOVER LOGONLY process is the most efficient, as long as the backup captures all of the data sets that were residing on the faulty volume. On the other hand, if a single tablespace needs to be recovered (especially if it crosses multiple volumes), the recovery will be most efficient if it uses a valid image copy of the tablespace taken by the COPY utility.
43.8 Recovery to a Prior Point in Time

This type of recovery is used to re-instate the SAP database at some previous point in time. All the changes that occurred after that time will be lost and the system will appear as it was at that time in the past. The decision to bring the system back in time must be carefully considered.

When to Use Prior Point in Time Recovery

A typical situation when a prior point in time recovery might be needed is an application program logic error that introduced unwanted changes into the system that could not be "reverse engineered". In some cases, the prior point in time recovery and the loss of data associated with it can be avoided by writing "compensating transactions". However, this can only be done by highly-skilled specialists with deep expertise in both the SAP system as an integrated system and the problem application area. In all other cases, a prior point in time recovery is the only safe course of action.

Available Options

There are different methods to accomplish a prior point in time recovery of an SAP database. Depending on which time is selected as the recovery target point and whether volume-based backups are available, the recovery methods can be categorized as follows:

- Recovery to the state at the time an offline backup of the SAP database was created
- Recovery to the state at the time the SAP database was quiesced
- Recovery to any prior point in time using object-based backups
- Recovery to the state at the time a volume-based online backup of the SAP database was created
- Recovery to any prior point in time using volume-based backups

Common to most of these techniques is that the SAP data that is not stored in DB2 (for example, SAP application-based archived data) cannot be recovered to the same prior point in time. In principle, that is not a problem, as this data is not considered a recoverable resource from the database perspective.

Improving Recovery Performance

You can speed up the methods that are based on the `RECOVER` utility by splitting the job into multiple parallel recovery streams and avoiding the DASD path contention. Keep in mind that the `REUSE` option of the `RECOVER` and `REBUILD` utilities will significantly reduce the overall recovery elapsed time. Also, the SAP system should be stopped and access to DB2 either restricted to recover jobs only (by specifying `START DB2 ACCESS(MAINT)`), or completely denied (by specifying `STOP DB2`) depending on the recovery method used.

Which of the recovery method will be used depends on:

- How fast the data must be available again
- How far back you can afford to recover the system to, prior to the point when the system got damaged
- Availability of offline backups
- Whether indexspaces were included in the backup
- Availability of quiesce points

The recovery methods are described in more detail in the following sections.
Recovery to the State at the Time an Offline Backup of the SAP Database Was Created

This is the easiest and fastest prior point in time recovery method. At the same time, it is also the most restrictive.

First of all, it requires that offline backups of the SAP database are created, which potentially causes long periods of system unavailability, which some SAP system installations cannot tolerate. Secondly, depending on the backup frequency, it can bring the system much further back than necessary. For example, if the offline backups of the SAP database are scheduled weekly on Sundays, and the data was damaged on Friday, all of the changes made from the last Sunday to Friday would be needlessly lost.

This method can be implemented by the TOCOPY option of the RECOVER utility, storage subsystem based volumes restore, DFSMSdss RESTORE, and so on, depending on how the offline backup was created. For example, if you created an offline SAP database backup by making the storage subsystem driven copies for all the involved volumes, restoring the copies of these volumes and starting DB2 completes the recovery.

Recovery to the State at the Time the SAP Database Was Quiesced

Recovery to the state at the time the SAP database was quiesced depends on the existence of "system quiesce points". These are the points in time when there are no uncommitted update transactions in the system. Such a point is represented by the corresponding RBA (or LRSN). There are a number of ways to quiesce the SAP database. The options are listed here from the least restrictive to the most restrictive to concurrent SAP system activity:

- ARCHIVE LOG MODE (QUIESCE) TIME(n) command
- QUIESCE utility
- START DATABASE ACCESS (RO) command for all DB2 databases
- STOP DB2 MODE (QUIESCE)

The least disruptive method to quiesce a DB2 system is the ARCHIVE LOG command. While other methods can be used, they are not recommended due to the serious impact on the concurrent online transactions.

After the ARCHIVE LOG MODE (QUIESCE) TIME(n) command is issued, the new update transactions wait for the command completion. The command is successfully completed if all the update transaction that were running at the time when the command was issued were committed before the time specified in the TIME option expired. Otherwise, the command fails and the quiesce point has not been established.

The TIME value should be 5-10 seconds lower than the resource timeout installation parameter value. That will prevent timeouts on the transactions that are queued after the ARCHIVE LOG command. A convenient way to specify the TIME value is to omit the option and accept the default that corresponds to the QUIESCE PERIOD installation parameter, provided that the parameter is set according to the above consideration.

The ARCHIVE LOG command also has the potential of being quite disruptive to the concurrent SAP activity. The command can repeatedly fail to succeed due to concurrent long-running, non-committing transactions. This results in an inability to establish quiesce points and cumulatively have a bad effect on the concurrent update activity.

The RBA value that corresponds to the quiesce point established in this way is recorded in the bootstrap data sets (BSDS) and on the MVS console.

Once a quiesce point has been established, the SAP database can be recovered to that point if the need arises. The TORBA or TOLOGPOINT options of the RECOVER utility should be used. You can speed up the recovery process significantly if only the objects that have been changed since the recovery target time are recovered. For more information on recovery to any prior point in time using object-based online backups, see the next section.
The method *Recovery to the State at the Time the SAP Database Was Qiesced* is more advantageous than the method *Recovery to the State at the Time an Offline Backup of the SAP Database Was Created*. This is because establishing a quiesce point is less disruptive to the SAP system than creating an offline backup of the SAP database. Therefore, it is more likely that the recovery will be possible to a point that is closer to the required time and the loss of data will be reduced. On the other hand, establishing frequent quiesce points may impair performance, which is not acceptable for some SAP system installations. This is why the recovery methods described in the following sections are generally more attractive because they are based on online backups and do not require system quiesce points.

### Recovery to Any Prior Point in Time Using Object-Based Online Backups

This recovery method uses the conditional restart technique and is the least obstructive to everyday operations in terms of creating all the prerequisites for a prior point in time recovery of the SAP database.

The main characteristics of this method are that neither offline backups nor quiesce points need to be provided, which makes it the prime choice in 24 x 7 SAP system environments. It can also bring the system closest to the time when the SAP database is known to be semantically and operationally consistent.

This recovery method assumes that a set of valid object-based backups (for example, tablespace, partitions and index image copies taken by the `COPY` utility) is available.

Conceptually, the recovery method includes the steps outlined below. Be aware that these steps do not provide a detailed, ready-to-run process; they only outline the most important points and considerations your recovery procedure will need to take into account.

1. Determine which RBA or LRSN approximates the time \( T \) you want to bring the system back to. This RBA is defined here as the target RBA, and implies LRSN as well unless explicitly stated differently.
   - Translate the time \( T \) (given as a timestamp) into its `STCK` format, that is, the LRSN to which you want to restart the data sharing group. To translate the time into its `STCK` format, you may use the Store Clock Utility. This JavaScript tool is available at [http://www.ibm.com/support/docview.wss?rs=64&context=SSEPEK&uid=swg24001778](http://www.ibm.com/support/docview.wss?rs=64&context=SSEPEK&uid=swg24001778).
   - Alternatively, in non-data sharing you may determine which log data set covers the interval that contains time \( T \) by using the print log map (`DSNJU004`) utility. Run `DSN1LOGP SUMMARY` on the above identified log data set and determine which RBA is the closest to the time \( T \). This RBA needs to be a multiple of 4096.

   If possible, try to avoid selecting target RBA that falls in a long running unit of recovery and would cause lengthy backouts.

2. In data sharing, delete CF structures.

3. Create a list of objects that need to be recovered.

   Namely, it is likely that for a large number of objects the current DASD contents is identical to the contents at the time corresponding to the target RBA. In other words, a lot of objects have not changed since the target RBA and currency. The objects that have changed need to be recovered. You can find these either by running `REPORT RECOVERY` for all the tablespaces in the system or by running `DSN1LOGP SUMMARY`. The log needs to be scanned from the last checkpoint before the target RBA (from the checkpoint’s begin RBA, providing that the checkpoint completed) to the currency.

   In addition to these, you need to recover the objects that were `REORG`ed with `NO LOG` (or `LOADED` with `NO LOG`, but the `LOAD` utility use in SAP system environments is rare), or that were dropped since the target RBA. The objects that were created since the target RBA can be ignored from the consistency viewpoint, but you might want to
identify them as well in order to delete (AMS DELETE) the corresponding orphan data sets.

The REORGed objects can be identified by selecting the matching SYSCOPY rows before the catalog is recovered to the target RBA.

The objects that were dropped or created since the target RBA can be found by matching the result of SELECT of all the rows from SYSTABLESPACE and SYSINDEXSPACE with the corresponding underlying data sets in the ICF. The SELECT must be done after the DB2 catalog is recovered to the target RBA. The dropped or created objects can be found more efficiently if you regularly trace the DROP and CREATE events (DB2 performance trace IFCID 62).

Finding only the objects that have to be recovered can very significantly reduce the total elapsed time for the system recovery. You should prepare and test this procedure in advance (execs to create REPORT or DSN1LOGP input job specifications based on the current data, analyze the output and create appropriate RECOVER and REBUILD specifications).

4. Copy BSDS and all the logs that contain RBAs that are later than the target RBA. This will allow you to repeat the recovery in case you decide you want to recover the data again, but to a later point in time.

Use DSNJU003 to create a conditional restart record. Set ENDRBA (ENDLRSN) to target RBA and leave all other CRESTART options at their defaults. To recover to any log point, you may specify a LRSN also in non-data sharing environments. If DB2 does not find an appropriate checkpoint record for ENDRBA in BSDS, you can use the CHKPTRBA option of the CRESTART statement to specify a checkpoint. Using the DSN1LOGP SUMMARY(ONLY) option, you can find a valid checkpoint for ENDRBA in message DSN1153I.

Start DB2, but previously update system parameters (panel DSNTIPS) and specify DEFER ALL.

This option means that all the objects that were in the started state at the target RBA will not be started at the next DB2 start, in other words, will not go through the normal restart process. However, that DEFER does not affect processing of the log during restart, in other words, DB2 still processes the appropriate log range, but the logged operations are not applied to the deferred start data sets.

During the start, a number of pages might be placed in the LPL/GRECP. These pages will be removed from the LPL/GRECP in the course of the corresponding tablespace and index recoveries that are done subsequently in the procedure.

Recover catalog and directory tablespaces and indexes (only those identified as to need the recovery) in exactly prescribed order to the current point in time, in other words, with no TOCOPY or TORBA/TOLOGPOINT. The order and some other special considerations are described in the IBM documentation DB2 for z/OS: Utility Guide and Reference in “RECOVER TABLESPACE” under “Recovering Catalog and Directory Objects”.

Recover the selected (only those identified as to need the recovery) tablespaces and indexes (with COPY YES) to the current point in time, in other words, with no TOCOPY or TORBA/TOLOGPOINT.

5. Rebuild the remaining indexes on the tablespaces recovered in the previous step.

Do not forget to reinstate RESTART ALL in DSNTIPS. This will allow DB2 to do normal restart processing during the subsequent subsystem starts.

6. It is possible that some of the transactions that use asynchronous update protocol are not fully rolled back after the system recovery, but they can be identified by transaction SM13 and an appropriate action taken. If you want to decide what to do with these transactions, instead of the SAP system deleting them at start-up time, you can set
rdisp/vbreorg to 0 for the first SAP system start-up after a prior point in time recovery (the default is 1).

7. Make an offline backup of the SAP database.

In some very special cases that should never be exercised without indepth expertise in SAP Web Application Server and applications (regularly with a direct SAP involvement), a prior point in time recovery consists of writing compensating transactions for some tables and recovering only a subset of the SAP database. In such cases the conditional restart method cannot be used on the target system, but it can still play a role in the overall recovery. In other words, the conditional restart recovery method can be performed on a system that is a copy (auxiliary or temporary) of the target system, and only selected tablespaces brought back to the target system.

**Recovery to the State at the Time a Volume-Based Online Backup of the SAP Database Was Created**

As the online, volume-based backup includes every relevant DB2 system and user data set, you can use such a system backup for starting DB2. All the volumes (data and log) need to be restored and DB2 normally started. This start will be performed as in the case of DB2 restart after an abnormal system termination. The inflight units of recovery will be rolled back, which brings the SAP database to a consistent state.

After DB2 comes up, you can use AMS to reformat the volumes that were added after the recovery point time. This does not affect consistency of the system, but removes orphaned data sets and extents.

In data sharing environments, you must force a group restart by purging all coupling structures from the coupling facility using the `SETXCF FORCE` command before any DB2 data sharing members are started.

This is a very simple yet powerful way of recovering an SAP system. It is important, however, that you are sensitive to at what time such a volume-based backup is taken. You should avoid doing so during long running units of recovery. Also, when you use this method, the system can only be recovered to this specific point when the backups were taken. For recovering to an arbitrary point in time using the volume-based online backups, you need to follow the procedure described in the next section below.

**Recovery to Any Prior Point in Time Using Volume-Based Online Backups**

Recovering to an arbitrary point in time with the RESTORE SYSTEM utility uses either backups that were taken by the BACKUP SYSTEM utility or it recovers by processing the log only assuming that volumes were manually restored. Be aware that these steps do not provide a detailed, ready-to-run process; they only outline the most important points and considerations your recovery procedure will need to take into account.

1. Determine which RBA or LRSN approximates the time \( T \) you want to bring the system back to. This RBA is defined here as the target RBA, and implies LRSN as well unless explicitly stated differently.
   
   o Translate the time \( T \) (given as a timestamp) into its STCK format, that is, the LRSN to which you want to restart the data sharing group.
   
   o Alternatively, in non-data sharing, you may determine which log data set covers the interval that contains time \( T \) by using the print log map (DSNJU004) utility. Run DSN1LOGP SUMMARY on the above identified log data set and determine which RBA is the closest to the time \( T \). The RBA needs to be a multiple of 4096. If possible, try to avoid selecting target RBA that falls in a long running unit of recovery and would cause lengthy backouts.
2. Run DSNJU003 Change Log Inventory with the CRESTART SYSPITR option to specify 
the log truncation point that corresponds to the point in time to which the system is to 
be recovered. For data sharing systems, repeat this step on all members specifying the 
same LRSN truncation point.

3. In data sharing, delete CF structures.

4. Start DB2. In data sharing, start all active members.

5. Submit the RESTORE SYSTEM utility job. If the backup has already been manually 
restored, use the LOGONLY option.

6. Recover objects that are marked in recover pending state and rebuild objects that are 
marked in rebuild pending state.

7. It is possible that some of the transactions that use asynchronous update protocol are 
not fully rolled back after the system recovery, but they can be identified by transaction 
SM13 and an appropriate action taken. If you want to decide what to do with these 
transactions, instead of the SAP system deleting them at start-up time, you can set 
rdisp/vbreorg to 0 for the first SAP system start-up after a prior point in time 
recovery (the default is 1).

8. Make an offline copy of the DB2 subsystem or data sharing group.
44 JES Interface

The JES Interface (transaction DB2J or transaction DB2 → Check/Settings → JCL Submission Service) enables you to submit and monitor any z/OS JCL job from within the SAP System. It serves not only as a standalone transaction, but also as an interface used by other transactions.
44.1 Profile Parameters

We recommend that you enter values for all profile parameters that can be maintained with the Profile parameter function of the JES Interface. For example, you must maintain the profile parameter values before z/OS jobs can be submitted from the SAP System.

The following table lists the profile parameters:

<table>
<thead>
<tr>
<th>Profile Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 run library</td>
<td>Library that contains program DSNTIAD</td>
</tr>
<tr>
<td>MGMTCLAS, STORCLAS, DATACLAS (SMS)</td>
<td>Optional parameters. If not specified, the defaults of the corresponding ACS routine are used.</td>
</tr>
<tr>
<td>Volume count</td>
<td>Overwrites the value defined in DATACLAS, if needed</td>
</tr>
<tr>
<td>Partitioned data set</td>
<td>Partitioned data set for uploading z/OS jobs. It must exist. You may also specify the job names and member names for jobs to be uploaded.</td>
</tr>
<tr>
<td>Console → Console output dataset</td>
<td>Sequential data set for the requested part of the z/OS system log. It must exist. It will be overwritten if requested again. The format must be as follows:</td>
</tr>
<tr>
<td></td>
<td>Organization: PS</td>
</tr>
<tr>
<td></td>
<td>Record format: VB</td>
</tr>
<tr>
<td></td>
<td>Record length: 133</td>
</tr>
<tr>
<td></td>
<td>Block size: 27930</td>
</tr>
</tbody>
</table>

These parameters are used by all authorized SAP system users.
44.2 SYSPLEX Failover Support and Connection Profile

Use

SYSPLEX failover support is the ability of the SAP system on DB2 to redirect application servers to a standby database server if the primary database server becomes inaccessible. When the application server detects such a situation, it automatically performs the redirection after rolling back the current transaction. You can also proactively redirect application servers to a standby database server using transactions ST04 or DB2.

Prerequisites

SYSPLEX failover support can only be used if DB2 data sharing is used. All primary and standby database servers must be members of the same data sharing group.

For more information, see the SAP documentation SAP on IBM DB2 UDB for z/OS: Database Administration Guide and the SAP online documentation BC SAP High Availability in the SAP Library.

Properties of Database Failover

In contrast to older SAP releases, more than one failover host is supported.

If failover is already set up on your system, failover is supported as before. All environment and variable profiles are analyzed as before. As long as no new functions are required, no conversions are necessary.

Failover (in the case of an error) is displayed by an entry in the system log. You can now also display more detailed information about the current connection status (active DB connections) and also display a DB connection list. To do this, call transaction ST04 or DB2 and choose Active DB Connections or DB Connection List. For more information on these functions, see the SAP documentation SAP on IBM DB2 UDB for z/OS: Database Administration Guide.

Connection Profile

To use more than one failover host, you need a connection profile. The name of the file is normally connect.ini and should be located in the following directory:

UNIX: /usr/sap/<sid>/SYS/global
Windows: \<SAPGLOBALHOST>\sapnt\<sid>\global

As soon as this file exists, the previous profile parameters are no longer analyzed for the connection and the failover. Instead, information is retrieved from the connection profile. This file has a special structure, which corresponds to the structure of a Windows ini file.

There are different sections, for example, [COMMON], in this file. You can recognize these sections by the square brackets. There may be several codes in each section, which have a value.

PORT=4711

describes the PORT code, which has the value 4711.

Logical Connection

A logical connection is a symbolic name for an actual DB connection and contains all the relevant information for the connection. The logical connection corresponds to a section in the connection profile. All of the codes that describe this connection are defined with this section.
This section describes the Logical1 logical connection, which creates a connection to the DBS1 subsystem DBL Connect Database Alias DBS1.

**Failover List**

The failover list describes all of the logical connections that can be used if a DB failover is required. The sequence given in the list specifies the order in which the links should be used. This type of failover list can be available for three different sections.

First, you can define a section for each individual application server.

If the name of the application server is server1, the section could appear as follows:

```
[server1]
CON1=Logical1
CON2=Logical2
CON3=Logical3
```

In this case, the server1 application server would first connect to the Logical1 logical connection. In the case of an error, it would connect to the Logical2 logical connection and if this is incorrect, also to the Logical3 logical connection.

Secondly, groups are defined the [GROUPS] section. Groups can contain a number of servers.

```
Group1=server1,server2
Group2=server3,server4
```

Group1 contains the server1 and server2 application servers, and Group2 contains the server3 and server4 application servers.

A failover list can be defined for each group. You can do this in the same way as for an individual application server.

Thirdly, a failover list can be specified globally in the [DEFAULT_GROUP] section. This section is used if no section is defined for this application server and also if no section is defined for a group that contains this application server.

**Sections**

The sections [COMMON], [GROUPS], [DEFAULT GROUPS], and [<Logical Connection>] are described in more detail below:

- **[COMMON]**
  
  This section describes global settings. All values are optional.

- **[GROUPS]**
This section defines groups of application servers, and is optional. Only the group name is valid as a key word (indicated as `<Group>` under (DEFAULT_GROUP) below.) A list of application servers is valid as the value of the key word. The names of the application servers must be separated by commas.

- [DEFAULT_GROUP]
  
  `<Group>`
  
  `<Application server>`

  All of the above sections are optional and contain the failover list. You must ensure that at least on section applies for each application server. Valid key words are all values in the form of CONx, where x is a sequential number starting from 1. All numbers from 1 to the highest number used must be represented.

  ```
  [DEFAULT_GROUP]
  CON1=Logical1
  CON3=Logical3
  
  The above failover list is not valid because an entry for the CON2 key is missing.
  ```

- [<Logical Connection>]

  This section contains all of the relevant information for the DB connection. Different key values may be listed depending on the platform of the application server (z/OS, UNIX, Windows). On UNIX and Windows platforms, DB2 Connect is used as connectivity.

  The following key values can be used:

<table>
<thead>
<tr>
<th>Key Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSID</td>
<td>DB2 Connect: Database alias</td>
</tr>
<tr>
<td></td>
<td>z/OS: Name of the DB2 subsystem or Group Attach Name</td>
</tr>
<tr>
<td>HOST</td>
<td>Name of the database host.</td>
</tr>
<tr>
<td>PLAN</td>
<td>z/OS: Plan name to which the database library (usually dbdb2slib.dll) was linked</td>
</tr>
<tr>
<td>LOCATION</td>
<td>DB2 Connect: DB2 Location name</td>
</tr>
<tr>
<td></td>
<td>z/OS: Name of the location if a remote access was defined through the CDB (not recommended)</td>
</tr>
<tr>
<td>PORT</td>
<td>DB2 Connect: DDF listening port</td>
</tr>
<tr>
<td>RETRY_CNT</td>
<td>Number of attempted calls to this connection until the next connection should be used</td>
</tr>
<tr>
<td>SLEEP_TIME</td>
<td>Number of seconds between the attempted connections</td>
</tr>
</tbody>
</table>

  You can change the name of the connection profile with the `dbs/db2/conn_profile` profile parameter, or the corresponding `dbs_db2_conn_profile` environment parameter.

  The following example illustrates a possible structure of a connection profile:

  ```
  Three application servers are available. Two of them are running on AIX (aix1, aix2) and one on z/OS (hostb). In addition, a DB2 data sharing group is
  ```
configured with three data sharing members. The first member (DBS1) is running on the first LPAR and can be accessed under the hosta host name. The other two members (DBS2, DBS3) are running on the second LPAR and can be accessed under the hostb host name. Both application servers under AIX should be combined for one group (aixgrp) and should use all three data sharing members for the failover. Only the data sharing members DBS2 and DBS3 should (and can) use the application server on z/OS.

Based on this information, the connection profile would have the following structure:

```
[GROUPEM]
aixgrp=aix1,aix2

[aixgrp]
CON1=DBS1_on_hosta
CON2=DBS2_on_hostb
CON3=DBS3_on_hostb

[hostb]
CON1=DBS3_on_hostb_zOS
CON2=DBS2_on_hostb_zOS

[DBS1_on_hosta]
SSID=DBS1
HOST=hosta
PORT=4001
LOCATION=LDBS1

[DBS2_on_hostb]
SSID=DBS2
HOST=hostb
PORT=4002
LOCATION=LDBS2

[DBS3_on_hostb]
SSID=DBS3
HOST=hostb
PORT=4003
LOCATION=LDBS3

[DBS2_on_hostb_zOS]
SSID=DBS2
```
PLAN=PLAN640

[DBS3_on_hostb_zOS]
SSID=DBS3
PLAN=PLAN640

- For more information on failover, see SAP Note 402078.
44.3 Prerequisites

- **SAP System**
  
  To use the JES Interface, a user needs as a minimum requirement authorization for database administration, in other words, the profile `S_A.ADMIN` must be entered for the administrator. The additional profile `S_DB_DBADM` authorizes the administrator to change any user's JCL jobs and the TSO password.

- **Database System**
  
  The user needs to be authorized to run the job submitted on the database system.

- **z/OS System**
  
  The communication between the SAP System and the z/OS host is made possible by using stored procedures provided by Control Center.

  To use the stored procedures and execute utilities and JCL jobs, you must make sure:
  
  - You have installed the stored procedures according to the instructions in [Installing Stored Procedures Using WLM](Page 65).
  - You have TSO access and a TSO user ID
  - Your SAP system user ID is identical to the TSO user ID (create a new SAP system user ID if necessary)
  - The system knows your TSO password (call transaction DB2J and choose `Password` to enter your password)
  - You have authorization to submit JCL jobs and the authority to use the utilities in question.
  - The user needs authority to create and read data sets under its own high level qualifier (HLQ) on the default volume.
44.4 Listing z/OS jobs

Specify an SAP system user ID in the Creator field of the initial screen of the JES Interface to display a list of all existing z/OS jobs created by this user.

To see all existing z/OS jobs in the SAP System, choose List all jobs.

All utility statements that are used for execution via stored procedures have a name that starts with DSNUTILS:.
44.5 Creating and Saving z/OS JCL Jobs

To create a job, enter a job name and choose Create.

The character strings ‘$$’ and ‘*’ are not allowed in job names. An empty string as job name is also not valid.

You are then prompted to specify one sign to distinguish your job on the z/OS side. This sign will be appended to your user ID at runtime to build up your job card ('//<user id><job sign> ....').

Choose Continue to access the SAP editor. The variable $JOBHEAD is already inserted in your text. It will be replaced by a user specific jobcard at runtime. How to create a user-specific jobcard is described in the section Creating and Changing an Individual Jobcard on the following page. Type in the JCL job and choose Save to save the job and leave the editor.
44.6 Changing z/OS JCL Jobs or Stored Procedure Utility Statements

To change a job or a stored procedure utility statement, select an existing job and choose Change. Enter a job sign and choose Continue to access the editor.

If you are neither the creator of the selected job nor the DB2 database administrator (authorization profile S_DB2_DBADM), the editor appears in read-only mode.

If you want to protect your changes to stored procedure utility statements from updates by SAP, you can do so in Profiles at tab strip General. Select Protect the utility statement skeletons and save the profile.
44.7 Deleting z/OS JCL Jobs

To delete a job, select an existing job and choose Delete job in the Execute menu. You are only allowed to delete your own jobs unless you have the authorization profile S_DB2_DBADM.

Stored procedure utility statements cannot be deleted.
44.8 Displaying JCL Jobs and Stored Procedure Utility Statements

To display a job, select an existing job and choose Display. All variables in a JCL Job (for example, $JOBHEAD) are replaced by the real values.
44.9 Entering a TSO Password

If you submit a JCL job the stored procedures log you on to TSO. To establish the connection your TSO password must be known to the system. To avoid you having to enter a password each time you submit a job, your TSO password (and user ID) is saved in an encrypted way. If there is no valid password stored in the SAP System, you will be prompted to enter your TSO password.

You can also choose Password to create/change your TSO password. This may be necessary if you want to submit z/OS jobs via the DBA Planning Calendar (transaction DB13).
44.10 Creating and Changing an Individual Jobcard

The user specific jobcard is used to replace the variable $JOBHEAD in the JCL job text. If you do not create your own jobcard, the following default is used:

//$USER$SUFFIX JOB USER=$USER,CLASS=B,MSGCLASS=X,
//MSGLEVEL=(1,1)

The variable $USER will be replaced by the SAP system user ID and the variable $SUFFIX will be replaced by the <job sign> which you define by creating a job. If you choose Job head, the SAP editor is called with a default jobcard. You can change this jobcard if you want. For example, you may add the SYSAFF parameter. Make sure you save your changed jobcard by choosing Save.

Ensure that there are no empty lines in your jobs.
44.11 Submitting z/OS JCL Jobs Asynchronously

To submit a JCL job, select an existing job and choose Submit or Display and Submit.

The stored procedure DSNACCJS logs you on TSO with your SAP system user ID (the first seven bytes of your user ID are used) and submits the JCL job. The FTP submission service does not wait until the JCL job is finished. You get control immediately after submitting the JCL job on z/OS. The status of your job is registered as submitted in the SAP System.
### 44.12 Checking the Status of the Job

You can monitor the status of your z/OS jobs by choosing *Job status*. A list of z/OS jobs submitted by SAP system user(s) appears, providing the following information:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobname</td>
<td>SAP system name of the z/OS job</td>
</tr>
<tr>
<td>Jobno</td>
<td>z/OS job identifier</td>
</tr>
<tr>
<td>Submitted by</td>
<td>User ID of the user who has submitted the job</td>
</tr>
<tr>
<td>Sub date</td>
<td>Day when the job was submitted</td>
</tr>
<tr>
<td>Sub time</td>
<td>Time when the job was submitted</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the z/OS job after submission or last refresh</td>
</tr>
</tbody>
</table>

Possible values of the *Status* field are:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>submitted</td>
<td>Directly after submission</td>
</tr>
<tr>
<td>unknown</td>
<td>No information found on z/OS about the job</td>
</tr>
<tr>
<td>active</td>
<td>Job still active on z/OS</td>
</tr>
<tr>
<td>finished</td>
<td>Job has finished on z/OS</td>
</tr>
<tr>
<td>not run</td>
<td>No z/OS initiator assigned to the job. Maybe no initiator</td>
</tr>
<tr>
<td></td>
<td>started for the CLASS used by that job.</td>
</tr>
<tr>
<td>hold</td>
<td>z/OS job assigned to an initiator, but all initiators started for the</td>
</tr>
<tr>
<td></td>
<td>CLASS used by that job are busy.</td>
</tr>
</tbody>
</table>

By choosing *Refresh* the status information of all jobs displayed on the list are refreshed.

To display the status of z/OS jobs submitted by other SAP system users, change the user ID in the *Submitted by* field and choose *Display Jobs*. 
44.13 Deleting the Job Output on z/OS and on the Client

To delete the z/OS job output and the status information of the job, select one or more jobs and choose *Delete output*. The status information on the job, the job output on the z/OS side and the job output on the client side is then deleted. You can only delete the jobs you have submitted unless you have the authorization profile `S_DB2_DBADM` and corresponding authorizations in RACF.
## 44.14 Displaying the Job Output

To display the job output:

1. Select one or more lines in the status list.
2. Choose *Display output*.

This displays a list of output files belonging to your selected jobs. It contains the following information about the output files:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobname</td>
<td>SAP name of the z/OS job</td>
</tr>
<tr>
<td>Filename</td>
<td>File name on the client side. The file name contains the z/OS job ID and an extension numbered consecutively</td>
</tr>
<tr>
<td>Size</td>
<td>Size of the file in bytes on the client side</td>
</tr>
<tr>
<td>Date</td>
<td>Date when the file was transported from z/OS to the client side</td>
</tr>
<tr>
<td>Time</td>
<td>Time when the file was transported from z/OS to the client side</td>
</tr>
</tbody>
</table>

To display the file contents:

1. Select one or more lines of the list.
2. Choose *Display files*. 

44.15 Cleaning Up the JES joblog Directory

The JCL Submission Service uses stored procedures as the JES interface. In order to make them compatible with the old protocol via FTP, the JCL jobs are first placed as file *jcljob* on the application server in the directory:

**AIX:** /usr/sap/<SID>/SYS/global/JESjoblog

**Windows:** \usr\sap\<SID>\SYS\global\JESjoblog

The jobs are either executed directly or placed in a partitioned data set.

If the job was executed directly, the job output is then saved again to the JES joblog directory.

To display the contents of the directory, choose **Extras → Clear directory...**

You can display the contents of selected files by choosing **File → Display**, and can delete the files by choosing **File → Delete** if required.
Some database administration tasks are very time-consuming, or can only be carried out when the database is in a particular state. Other tasks must be repeated regularly, for example backups. You can schedule and coordinate these tasks using the DBA Planning Calendar of the SAP system.

To use the DBA Planning Calendar, choose one of the following:

- Choose DB Administration → Planning Calendar from the CCMS menu.
- Call transaction DB13.
- Call transaction DB2 and choose DBA Planning Calendar.
- Call transaction DB13 and choose DBA local Calendar.

Actions can be scheduled in advance using background processing. These actions are then executed automatically.
45.1 Preparations for Using the DBA Planning Calendar

All database administration tasks must be secure. Therefore, authorization checks must be made for certain operations in the SAP System, in the database system and on the z/OS host.

- **SAP System**
  
  In the SAP System, a user needs authorization for database administration and background job scheduling to use the DBA Planning Calendar.

  The administrator must have the authorizations `S_RZL_ADMIN` and `S_BTCH_ALL`, which are included in the operator profile `S_A.ADMIN`.

- **Database System**
  
  The user must be authorized to run the DB2 utility corresponding to a certain administration task.

  You need to have set up the Stored Procedures according to the instructions in *Installing Control Center Procedures Using WLM* [Page 71].

  See the DB2 UDB for z/OS documentation *Utility Guide and Reference* for more information.

- **z/OS Host**
  
  See *JES Interface* [Page 246] for information on how to set up z/OS specifics.

  You must maintain all your profile parameters with the JES Interface (transaction DB2J, button *Profile parameters*).

  Depending on your security setup, you may need to provide user and password in JES Interface (transaction DB2J, button *Password*).

  Jobs generated from the DBA Planning Calendar are partly based on the data from the table and index monitors (DB02). Make sure that the table and index monitor data is regularly updated.

  The size information gathered by transaction DB02 is used for data set allocation, for example, in *REORG* and *COPY* jobs.
### 45.2 Plannable Actions

The main function of the DBA Planning Calendar is to specify starting times and parameters for database actions, which then run without interaction with the administrator. You need to make sure that the necessary resources are available before execution time.

The following plannable actions are currently realized for DB2 UDB for z/OS:

- **Backup for all SAP tablespaces (full image copy)**
- **Increm. backup for all SAP tablespaces**
- **Backup of suggested objects**
- **Recovery of one SAP tablespace**
- **Recovery of one SAP index**
- **Rebuilding of one SAP index**
- **Online REORG of one SAP tablespace**
- **Online REORG of one SAP index**
- **Online REORG on suggested tablespaces**
- **Online REORG on suggested indexes**
- **Update statistics for one SAP object**
- **Update statistics for all SAP objects**
- **RUNSTATS on objects needing new statistics**
- **Update data for transaction DB02 (tables and indexes monitor)**
### 45.3 Basic Functions

The following basic functions can be executed:

<table>
<thead>
<tr>
<th>Function</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Schedule a new action</strong></td>
<td>Double-click the day on which the action is to be started. The system prompts you to specify parameters for the action. You can assign a certain time delay (in weeks) to avoid having to plan repeated actions more than once. Except for the update of transaction DB02, you can schedule any action either for execution on the OS/390 host or upload a corresponding OS/390 job into a partitioned data set on the host and use an external scheduler. The desired mode is one of the parameters to be specified.</td>
</tr>
<tr>
<td><strong>Change a scheduled action</strong></td>
<td>Select an action and change it.</td>
</tr>
<tr>
<td><strong>Display results of executed actions</strong></td>
<td>Select an action and choose Action logs. Unsuccessful actions are highlighted in red in the calendar.</td>
</tr>
<tr>
<td><strong>Display or change parameters of an action</strong></td>
<td>Click the header for a day and choose Parameters to display the parameters and Change to change the parameters of an action.</td>
</tr>
<tr>
<td><strong>Start an action immediately</strong></td>
<td>The current day's actions can be started immediately.</td>
</tr>
<tr>
<td><strong>Delete a scheduled action</strong></td>
<td>Select a specific day and then the action you want to delete. Choose Delete. This function is only possible for future actions.</td>
</tr>
<tr>
<td><strong>Restart an aborted action</strong></td>
<td>Double-click the header of the day on which the action was aborted. Position the cursor on the relevant action and choose Restart. Choose a date and time for the restart. The job can be executed directly from the SAP System or simply loaded into a partitioned data set on the OS/390 host. In this case, it is possible to manually change the job before execution.</td>
</tr>
</tbody>
</table>

Check regularly that the scheduled actions are running correctly. The calendar allows you to display and check the status of an action. A job log is generated which contains details of the background jobs used. Unsuccessful actions are displayed in the DBA Planning Calendar in a different color.

All scheduled actions can also be initiated immediately if you define the starting point as immediately. The actions then run using background processing. This enables you to execute an action manually if, for example, the planned action was unsuccessful.
45.4 How the DBA Planning Calendar Works

Since the transition from JCL submission via FTP to execution via stored procedures, the way in which the DBA Planning Calendar works has considerably changed. One of the most important changes is the security concept. For more information, see Security Considerations [Page 271].
45.4.1 Security Considerations for Stored Procedures

Up to release 6.20, JCL jobs were executed by logging on via FTP to the z/OS system as the SAP user that has scheduled the DBA action. For this reason, you had to have a TSO user that has the same name as the SAP user and needed to maintain your password in the JES Interface (transaction DB2J, button Password).

Stored Procedures are executed by different users depending on your application server setup:

1. For application servers on z/OS, it is the user who started the application server. Normally this is the <sapsid>adm user.
2. For application servers connecting via DB2 Connect, it is the user connecting to DB2. Normally, this is the DB2 Connect user. For more information, see Setting Up DB2 Connect User ID and Password [Page 87].

In order to ensure a successful execution of the Stored Procedures, the executing user needs an OMVS segment.

Security Implications

Regarding point 1:

This poses no security threat, because <sapsid>adm user already has an OMVS segment. The main advantage is that you do not need to maintain your TSO password in JES Interface (transaction DB2J, button Password) anymore.

Regarding point 2:

DB2 Connect user should have a password that does not expire (see Setting Up DB2 Connect User ID and Password [Page 87]). Some company’s security policies may prohibit giving this user an OMVS segment. Therefore, the security model is maintainable in the JES Interface (transaction DB2J, button Profile Parameters):

- If you choose DB2 Connect User, the DB2 Connect User needs an OMVS segment. This is the most comfortable solution, because you do not need to maintain your TSO password in JES Interface (transaction DB2J, button Password) anymore. It is standard to secure SAP by placing database and application servers behind a firewall. If you adhere to this standard, the impact on security is limited.
- If you choose Administrator, then the old behavior is emulated, by performing a multiconnect to DB2 under the authority of the administrator. Each administrator has to have an SAP user with the same name as his TSO ID and has to maintain his own password. The DBA actions will be executed under the administrator’s TSO user that has scheduled it.
- If you choose User, then you have to specify a user in the related input field that needs an OMVS segment and DB2 SYSCTRL authority. All DBA actions are executed under this user by performing a multiconnect to DB2 under this user’s authority. The advantage is that only this user’s password has to be maintained, and not all SAP administrators need an identically named TSO user to execute DBA actions.

Regarding Security Models with DB2 Connect User and User:

Security Models 1 and 3 are protected from uncontrolled JCL job execution by the SAP security system. In order to execute the relevant administrative transactions, you need operator profile S_A.ADMIN or S_DB_DBADM. Security Model 3 is the recommended option, because SAP support will be able to help you when logged on to your system without needing an identically named TSO user, which is often not possible due to company policies.
45.4.2 Executing a DBA Action

At the scheduled time for a DBA action, an SAP background job starts and executes the action. The action is processed depending on if you wish to have JCL uploaded or the action executed immediately:

Upload JCL

The SAP background job generates a z/OS JCL job based on a z/OS job template. The z/OS job templates are found in the JES Interface (transaction DB2J) and have the creator SAPR3.

The template is adapted to the customer specific configuration by substituting current values for several parameters. Since this implies profile parameter values, make sure that you maintain all profile parameters with the JES Interface (transaction DB2J), function Customizing → Profile parameter.
Once the JCL job has been generated, it is written to the file `<userid>_<application server>_<system number>_w<work process id>.jcljob01` on the application server:

**AIX:** `/usr/sap/<SID>/SYS/global/JESjoblog/`

**Windows:** `\usr\sap\<SID>\SYS\global\JESjoblog\`

Then, the stored procedure DSNACCDS is used to save the JCL Job as a partitioned data set member on the host. The stored procedure DSNACCDS uses the SAP system user ID (which therefore has to be identical to the TSO user ID) and the TSO password entered with the JES Interface (transaction DB2J) for authentication on the host.

The name of the partitioned data set (PDS) has to be maintained in JES Interface (which can be found under `transaction DB2J → Customizing → Profile parameters → Upload → Partitioned dataset for JCL jobs.`) Make sure that all possible Stored Procedure Users have UPDATE authority for this PDS.

For each action in the DBA Planning Calendar, you can define a separate job and member name for upload in the JES Interface [Page 246] at `Profile parameter → Upload`. By uploading data sets with stored procedure DSNACCDS can fail with return code 9990. For this, someone has the partitioned data set allocated that is the target of the upload. To resolve this problem, you can either organize the release of the data set in question or change the profile parameter `Partitioned dataset for JCL jobs` in transaction DB2J at `Customizing → Profile parameter → Upload` and restart transaction DB13 subsequently to reload the profile parameters.
Execute Action Immediately

DB2 online utilities are executed via the stored procedures DSNUTILS and DSNACCMO. DSNUTILS is suitable for a single utility execution, DSNACCMO executes utilities on multiple objects in an optimal parallel degree. Please note that DSNACCMO utilizes DSNUTILS for the actual execution of the utilities.

Both stored procedures are slightly faster than JCL job submission, because JCL interpretation is no longer needed.

The action log shows the input and output parameters of the stored procedures on the summary page. The screenshots are for DSNACCMO, because for DSNUTILS, the action log is similar and easier to understand.

DSNACCMO input and output parameters

<table>
<thead>
<tr>
<th>Input objects</th>
<th>Utility statements</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DB2 Parameters
- Start of the operation: 2004-05-18 16:24:52
- Scheduled action: CBR

DSNACCMO INPUT parameters
- Maximum number of parallel subtasks: 30
- Optimize runtime by sorting workload: NO
- Utility condition, which will stop procedure: AUTHORIZE
- Save restart info for stopped utilities: YES
- Utility-ID stem: APF/SQL418
- Number of input objects to be treated by utilities: 4:199
- Timeout (in seconds) before cancelling DSNACCMO: 6
- Escape character code of DB2 server: 0

DSNACCMO OUTPUT parameters
- Highest return code of DSNUTILS for all utilities: Hex 0004E000
- Number of actual utility executions: 87
- Number of parallel tasks actually started: 30
- Return code for the IF call: 0

Messages for IF call return code > 0

Parallel utility execution completed successfully...$

The most interesting field on this screen is the first output parameter of DSNACCMO named “Highest return code of DSNUTILS for all utilities”. Since DSNACCMO executes DSNUTILS for each input object, this high water mark will tell if there were DSNUTILS executions that failed.

In this example, one DSNUTILS execution failed with return code Hex 0004E000. This is the worst case scenario, because DSNACCMO will stop further execution (contrary to a DSNUTILS return code of 8). That is the reason why the input parameter “Number of input objects to be treated by utilities” has a different number than the output parameter “Number of actual utility executions”.

If you select Output, you will be shown the output of the utility executions.
All rows with the same **Object ID** belong to the same utility execution of the input object with this ID (see the **Input objects** button). The marked area shows the output of the utility run that has abended.

The bigger processes like **Backup for all SAP tablespaces** produce huge amounts of output that cannot be analyzed easily. If such a job returned with highest condition code 8, you can search in the output for the string **CODE=8**. If such a job returned with highest condition code **Hex0004E000**, you should search for the string **ABEND**. In the latter case, there can additionally be some utilities with condition code 8, so you should search for both strings.

The button **Input objects** will show which database objects are submitted:
Please note the values for **Statement ID**. They link the object to a utility statement for this execution. The utility statements are shown when you select **Utility statements**.
45.5 Using the DBA Planning Calendar

The transaction DB13 is the DBA Planning Calendar. To display the status of a day's actions, choose the header for a day.

Scheduled actions can be changed or deleted. Executed actions can only be displayed.

Unsuccessful or interrupted actions are shown in red in the calendar and, if there are only unsuccessful or interrupted actions, the header for this day is also highlighted in red.

In the scheduling overview, you can see if any logs were written for an action. By selecting a certain schedule, you can survey the action logs or background job log by choosing Action logs and Job logs. These functions detail all logs written on the day you have selected. Note that some actions do not write logs.
45.6 Backup

The DBA Planning Calendar gives you the option of making a complete backup (full image copy) or an incremental backup (incremental image copy) of all SAP tablespaces or a backup of SAP tablespaces according to the recommendations of stored procedure DSNACCOR.

CHANGELIMIT

The incremental backup uses a CHANGELIMIT threshold that switches to full image copy if this threshold of changes has been reached. Whilst a backup job is running, other programs have read-write access to the relevant tablespace (SHRLEVEL CHANGE).

For MCOD, you must avoid making simultaneous backups on several SAP systems, which have been scheduled on different SAP systems. These can lock each other out, resulting in errors.

Utilities Running in Parallel

The number of utilities running in parallel is controlled with a profile parameter. The higher this number, the faster the execution will be. But do not set this number too high in data sharing environments, otherwise your group buffer pools may run out of space, resulting in utility failure.

Upload Function

If you use the upload function, a separate job for the directory tables SYSIBM.SYSCOPY, SYSIBM.SYSUTILX and SYSIBM.SYSLGRNX is generated, named either FICPYSYS (full image copy) or IICPYSYS (incremental image copy). It is located in the PDS specified for the upload. Make sure that this job is only executed after the other COPY jobs have run, otherwise locking conflicts could arise.

Template Utility

For allocation of data sets, the TEMPLATE utility is used, which has the advantage of being able to automatically size the data sets based upon catalog statistic information. Please make sure that the catalog statistics are reasonably up to date, to prevent Out of Space conditions.

Profile Parameters Available for Backup

The following profile parameters exist for backup. You maintain these with the JES Interface (transaction DB2J) function Customizing → Profile parameters:

- Storage Parameters:
  - HLQ for Backup data sets:
  - MGMTCLAS (SMS)
  - STORCLAS (SMS)
  - DATACLAS (SMS)
  - Volume count
  - Number of GDG generations:

GDG is the abbreviation for “generation data group”. This is the administrative unit for a collection of historically related non-VSAM data sets that are arranged in chronological order. GDGs themselves do not contain user data.

Each data set that is part of the GDG is called a generation data set (or short: generation). The generations contain the user data. A GDG can consist only of a limited number of generations. This number is specified at initial creation of the generation data group base entry that defines the GDG. Whenever a new
generation is created and this number is exceeded, the oldest generation is replaced by the new one. That way, an automatic garbage collection of outdated data sets can be implemented, for instance for backup data sets.

You can configure the number of generations here. The values allowed are 0 to 255. If you specify 0, GDGs are not used by SAP. Any other number will be the number of generations for newly created GDGs in the future. In other words, the "Number of GDG generations" for already existing GDGs will not be changed.

The naming convention for backup data sets is

\[<\text{HLQ for Backup data sets}>,<\text{database}>,<\text{tablespace}>,\text{CCC}\]

if you use GDGs. Otherwise, the data sets have the following naming convention:

\[<\text{HLQ for Backup data sets}>,<\text{database}>,<\text{tablespace}>,\text{CCC}\text{nnnnnn}\]

where \text{nnnnnn} is the utility ID and CCC are three characters describing the type of COPY. The first character is I for incremental or C for full COPY. The second character is L for local or R for remote copy. The third character is P for primary or B for backup copy.

A good number here depends on your environment. Let's assume, that you take up to 3 backups per week (2 incremental and one full backup). If you need to keep the backups of the last 6 months (26 weeks), you should specify \(3 \times 26 = 78\) for "Number of GDG Generations".

- COPY-specific parameters (choose COPY):
  - Number of parallel jobs
  - Change limit:
    Percentage of the changed pages of a tablespace, above which a full copy, not an incremental copy, is made. For more information, see the IBM documentation DB2 UDB for z/OS Utility Guide.
  - Age for MODIFY RECOVERY:
    The MODIFY RECOVERY utility deletes records from the SYSIBM.SYSCOPY catalog table, related log records from the SYSIBM.SYSLGRNX directory table, and entries from the DBD. MODIFY RECOVERY should run regularly to clean up outdated information. This saves much space and, even more important, speeds up processes that access data from the tables mentioned above, for instance COPY utility and RECOVER utility.
    MODIFY RECOVERY has a parameter AGE that can be maintained here. MODIFY RECOVERY deletes all records older than the specified number of days. If you specify 0, MODIFY RECOVERY will not be used.

The recommended number depends on your environment and should be correlated to the "Number of Generations" mentioned above. This is due to the following reasons:

- MODIFY RECOVERY only cleans up records. The backup data sets themselves will not be deleted automatically. On the other hand, only in SYSIBM.SYSCOPY are the names of the backup data sets noted, so after having deleted these records, you will not be able to determine which backup data sets have to be deleted.

There are two solutions to this problem: Either use GDGs or configure in SMS a retention period for Management Class of the data sets in question, that has the same value as the AGE parameter of MODIFY RECOVERY.

- A record in SYSIBM.SYSCOPY without the backup data set is useless.
A backup data set without a record in SYSIBM.SYSCOPY can not be used by the RECOVER utility. You need to take a series of complicated steps to still use it (e.g. DSN1COPY or LOAD and very old archive logs). This should be reserved to emergency and won't be your normal mode of operation.

Here are some recommendations:

- If you do not use GDGs or some other means to discard old backup data sets, do not use MODIFY RECOVERY from within SAP, for example, specify 0 for "AGE for MODIFY RECOVERY".
- The minimal number is the age of the last successful FULL IMAGE COPY.
- Correlate it to "Number of GDG Generations". In the example above, you specified 78 for "Number of GDG Generations", which is sufficient to keep 6 months worth of backups. Correspondingly, you should specify 183 days for "AGE for MODIFY RECOVERY".

The backup jobs use data from the table and index monitor to size the backup data sets correctly, if the statistics data in catalog or in Real-Time Statistics is not available.
45.7 Recovery

If required, damaged SAP tablespaces or indexes can be recovered to the current status. If it is a partitioned table/indexspace, you have the opportunity to specify a single partition which should be recovered. Recovery to a prior point in time is not possible for an individual tablespace for consistency reasons.
45.8 Rebuild Index

If required, you can rebuild damaged SAP indexes. This is done by deleting the old index and creating a new one with the utility REBUILD INDEX.
45.9 Reorganization

Before scheduling a REORG job, you should make sure that you have a reasonably recent backup of the system for disaster recovery.

The REORG job allows the reorganization (by partition, if required) of an SAP tablespace or index during online operation. This means you have both read and write access during the reorganization (SHRLEVEL CHANGE).

Special Considerations for LOB Tablespaces

Note that for LOB tablespaces, an online REORG is not possible and therefore SHRLEVEL NONE is defined. If at least one of the tablespaces, you intend to reorganize, is a LOB tablespace, you are informed. We recommend that you upload the job to z/OS and submit it only during a maintenance window, when the SAP system is down.

For all tablespaces upon which inline RUNSTATS is allowed, inline statistics are generated.

Extents

If the number of extents of the corresponding data sets is larger than four, then the actual REORG is proceeded by an ALTER on the primary and secondary quantity as a separate job step, so that the table or indexspace shows a maximum of four extents after the reorganization. As many sort work and data work datasets are provided as needed.

When a Mapping Table is Necessary

When performing an online reorganization of a tablespace (not an index), you need to create a mapping table is needed (see the IBM documentation DB2 UDB for z/OS Utility Guide). SAP automatically creates a tablespace DSNDB04.QT<nnnnnn> with the stogroup MAPZZQAA. <nnnnnn> is a six digit number. In this tablespace, a table QT<nnnnnn> with a unique index QX<nnnnnn> is created. After successful reorganization, DROP TABLESPACE DSNDB04.QT<nnnnnn> is executed. These tablespaces are reused for faster performance.

To ensure that the CREATE statements run error-free, you must make sure that the stogroup MAPZZQAA exists.

The names QT<nnnnnn> or QX<nnnnnn> are reserved for this application. Do not use these names to create any tablespaces, tables or indexes.

Profile Parameters Available for the REORG

The following profile parameters exist for the reorganization. You maintain these with the JES Interface (transaction DB2J), function Profile parameters:

- Storage Parameters:
  - HLQ for Backup data sets
  - HLQ for work data sets
  - MGMTCLAS (SMS)
  - STORCLAS (SMS)
  - DATACLAS (SMS)
  - Volume count
Overwrites the value defined in \textit{DATACLAS}, if necessary.
  
  - \textit{Number of GDG generations}:
    
    If a value bigger than zero is specified, then the copy and work data sets are allocated as GDGs.

- \textbf{General}:
  
  - \textit{DB2 run library}:
    
    Library containing the program \texttt{DSNTIAD}
  
  - Plan of program \texttt{DSNTIAD}

- \textbf{REORG-specific parameter (choose \textit{REORG})}:
  
  - \textit{Number of parallel jobs}
  
  - \textit{Size of work datasets/unload dataset}:
    
    Space requirement grows with the number of indexes; default 2, increase if necessary
  
  - \textit{Default action when submitting “REORG on suggested tablespaces” non-periodically}:
    
    When \textit{REORG} on suggested tablespaces is planned periodically, LOB tablespaces are not included into the job, because on them ONLINE \textit{REORG} is not possible and severe performance impact on the running SAP system would be the result.

    When planning this action non-periodically, you can configure here if LOB tablespaces are to be included or not. If the job at time of planning will contain a LOB tablespace, a warning will pop up. Before running the job, please consider to take a recent backup of the system and to upload the job, so you can shutdown SAP system previously.

  - \textit{Which COPY options are to be used}:
    
    You can specify, if a remote copy should be taken also.

    The \textit{REORG} jobs use data from the table and index monitor.

\textit{REORG ALL} indexes and tablespaces uses the same parameters as a single \textit{REORG}. 

45.10 Update Statistics (RUNSTATS)

From the DBA Planning Calendar you can update the statistical information for one or all SAP tablespaces or for those tablespaces for which a RUNSTATS is recommended. See also section RUNSTATS needed.

LOB Tablespaces

Many RUNSTATS options are not valid for LOB tablespaces. More precisely, the RUNSTATS jobs for these tablespaces only specify the tablespace name and SHRLEVEL (CHANGE). Other options are ignored when specified. It is possible that running RUNSTATS on a tablespace that owns one or more auxiliary LOB tablespaces will also trigger RUNSTATS on the auxiliary tablespaces.

Update Statistics for One SAP Object

There are several tablespaces whose statistics should not be updated to make the DB2 optimizer favor an index-based access (see also Maintaining DB2 Catalog Statistics). These are excluded automatically from any update for all SAP tables and recommended SAP tables.

If you have a special reason to update statistics for a certain tablespace that falls into the above category, you have to schedule an Update statistics for one SAP object and specify the table or tablespace. A dialog box appears prompting you for confirmation.

Tables DB2NORUN and DBSTATC and DSNACCOR exception table

You might want to exclude tablespaces from the process of updating statistics and raising alerts for several reasons.

To exclude tablespaces from consideration in Update statistics for all SAP objects (RUNSTATS ALL) and RUNSTATS on obj. needing new statistics (RUNSTATS ALERTS) you have to insert the tablespace name and database name into table DB2NORUN using transaction SE16 or manually using SPUFI.

In addition, table DBSTATC can also be used to exclude tables from RUNSTATS ALERTS. As soon as the ACTIVE flag in DBSTATC is set to N, no alert will be raised on this table. This implicitly excludes tables from RUNSTATS ALERTS.

In other words, setting the ACTIVE flag in table DBSTATC only covers RUNSTATS ALERTS, excluding tablespaces from RUNSTATS ALL is only possible via DB2NORUN.

When the statistics for the excluded tablespace are not maintained, they continue to raise alerts. If you want to suppress this behavior, you must set the ACTIVE flag in DBSTATC to N for all tables within this excluded tablespace.

When the statistics for the excluded tablespace are not maintained, they continue to raise alerts. If you want to suppress this behavior, you must set the ACTIVE flag in DBSTATC to N for all tables within this excluded tablespace.

SAP exploits real time statistics and stored procedure DSNACCOR since DB2 V7 (see CCMS Monitor Set [Page 217] in Monitoring and Performance). The exception table of DSNACCOR will exclude any tablespace or indexspace in the subsystem for consideration for alerts on utility runs. The DSNACCOR exception table replaces the tables DB2NORUN, DBSTATC, DB2REOTS, and DB2REOIX completely. While the old tables are still respected for compilation of jobs in DB13, you have the benefit of getting no unnecessary alerts in transaction RZ20, so it makes sense to port your entries in these tables to DSNACC.EXCEPT_TBL. In addition to this benefit, you are no longer restricted
to objects within that special SAP system, which may be interesting if you run MCOD or have non-SAP objects apart from system catalog in the subsystem.

**Profile Parameters Available for RUNSTATS**

The following profile parameters exist for **RUNSTATS**. You maintain these with the JES Interface (transaction DB2J) function *Profile parameters*:

- **RUNSTATS**:
  - *Number of parallel jobs*
    You can choose the number of jobs to be executed in parallel, speeding up the process.
  - *Low/medium/high percentage of rows sampled (under RUNSTATS)*
    You can choose between three levels of accuracy (low, medium, high). These differ in the percentage of lines examined in the table(s). You can set the default level of accuracy and the corresponding percentages.
  - *Age for MODIFY STATISTICS (under RUNSTATS)*
    DB2 V7 RUNSTATS includes a new option, HISTORY. If specified, every time the corresponding catalog tables are inserted or updated, some selected fields are copied to some new catalog history tables. This results in preserving the relevant statistics data across multiple RUNSTATS executions and creates a basis for monitoring the database objects growth and assistance in determining optimal allocation quantities. Additionally, the historical data could be used for reinstating old catalog statistics in case of access path degradation.

    There is a utility named MODIFY STATISTICS to clean up the statistics history catalog tables that can grow considerably large, if you run RUNSTATS as frequently as recommended. MODIFY STATISTICS has a parameter AGE that can be maintained here. AGE can be set from 0 to 32767 days. The utility deletes all records older than the specified number of days. If you specify AGE=0, MODIFY STATISTICS will not be used.

    Since SAP does not yet use the HISTORY option of RUNSTATS, the recommended value is 0. If you have updated the RUNSTATS utility statement to use the HISTORY option, then the recommendation depends on your usage of the data:

    - If you just want to use the historical data for reinstating old catalog statistics in case of access path degradation, a value of 30 will be sufficient.
    - If you want to monitor database objects growth, a value of 365 to 730 days is more appropriate.

- **General**:
  - *Plan of program DSNTIAD* (General parameters section).

For more information on the **RUNSTATS** utility, see *Database Management*, section RUNSTATS.
45.11 Table and Index Monitor Update

The data of the table and index monitor should be updated at regular intervals. In an effort to schedule all CCMS actions centrally, the table and index monitor update is one of the functions of the DBA Planning Calendar.

See Tables and Indexes Monitoring [Page 201] in Monitoring and Performance for more information.
45.12 Restarting Jobs

The utility executions started from the DBA Planning Calendar can be aborted for different reasons. One of these reasons could be an insufficient amount of disk space.

Some DB2 utilities (COPY, REORG TABLESPACE, REORG INDEX, RECOVER INDEX, REBUILD INDEX, RECOVER TABLESPACE) allow you to restart the utility that was aborted.

The DBA Planning Calendar supports this option as well. Restart allows you to either execute the utility immediately or load a JCL job onto the z/OS host, where it is possible to make changes manually to the JCL job before execution.

Procedure

1. Clear the error which caused the abort.
2. Double-click the header of the appropriate day, place the cursor on the aborted action and choose restart.

   The SAP system determines the Utility ID of the chosen action and checks whether the database recognizes a stopped utility with the relevant Utility ID.
3. If the database recognized the utility, then you can schedule the restart job as described above. Otherwise, an error message is displayed.

Profile Parameter for Restart

The following profile parameter exists for the restart in addition to the profile parameters necessary for the utility that will be restarted. You maintain these with the JES Interface (transaction DB2J), function Profile parameters:

- General:
  - Utility restart at phase
45.13 Displaying the DB2 Utility Status

To display the status of all DB2 utilities jobs, choose DB2 utility status on the initial screen.
45.14 Checking the Results of Actions

All actions except “Updating the table and index monitor” scheduled in the DBA Planning Calendar generate logs, and also give the user details of the results of an action. Background jobs are used to schedule actions and these background jobs generate job logs. You can view all the information using the DBA Planning Calendar. In addition, all previous update statistics actions can be displayed.

Select the header for a day to display the status of a scheduled action. This status information is then displayed:

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCHED</td>
<td>The action has been scheduled, but the scheduled time has not yet been reached.</td>
</tr>
<tr>
<td>OK</td>
<td>The action has been completed successfully.</td>
</tr>
<tr>
<td>ERROR</td>
<td>The action has terminated abnormally.</td>
</tr>
<tr>
<td>N.A</td>
<td>No log giving information about the status of the action is available. It is possible that the action is currently being executed.</td>
</tr>
<tr>
<td>TIME</td>
<td>The action was executed until the maximum runtime was reached.</td>
</tr>
<tr>
<td>WARN</td>
<td>The action ended with a warning.</td>
</tr>
<tr>
<td>CANC</td>
<td>The action was canceled.</td>
</tr>
<tr>
<td>ABORT</td>
<td>The action was aborted.</td>
</tr>
</tbody>
</table>

Unsuccessful actions are highlighted in red in the DBA Planning Calendar.

If the submission of an z/OS job fails, you can find the error information in the job log. In this case, the SAP job is shown as not run in the DBA Planning Calendar.

If the submission of the z/OS job is successful, you can choose Action log to get all information about the job. By choosing Action log, a list will be shown where you find the submitted JCL job on z/OS, the z/OS JES job ID and the SAP name of the job.

To display the z/OS job output, double-click the action and choose JES job output. The z/OS job output list is displayed immediately if the list is available on z/OS.

If you need information about the status of the z/OS job, double-click the action and choose JES Interface.
45.15 Adjusting the Jobs

It is possible to adjust the job skeletons that are used in the DBA planning calendar for Upload to suit your individual requirements. This allows you to use brand new features of DB2 utilities that are useful for your shop.

To adjust a job skeleton for the DBA planning calendar, proceed as follows:

4. Call transaction DB2J (JCL Management Entry).
5. Choose list all jobs to display all jobs.
6. Select a job whose skeleton you would like to modify. The names of the jobs are self-explanatory, for example the skeleton of the RUNSTATS job for an individual table (or tablespace) is called RUNSTATS_TABLE. Note that the job must first be selected in the list and then chosen by selecting the button “Select job”.
7. Choose Change to go to an editor in which you can make changes to the skeleton. However, keep the rough structure of the job, as the internal parsing routines will otherwise not be able to deal with the skeleton.

The following is an example of a section of the RUNSTATS_TABLE skeleton:

```
RUNSTATS TABLSPACE $DB.$TS
    TABLE($TB)
    SAMPLE($SAMP)
    COLUMN(ALL)
    INDEX($IX)
    SHRLEVEL(CHANGE)
```

To use the history option, you can change the skeleton here as follows:

```
RUNSTATS TABLSPACE $DB.$TS
    TABLE($TB)
    SAMPLE($SAMP)
    COLUMN(ALL)
    INDEX($IX)
    HISTORY ALL
    SHRLEVEL(CHANGE)
```

8. Save the changes. They will come into effect the next time the relevant job is run.
45.16 Central DBA Planning Calendar

The Central DBA Planning Calendar creates a single point of control for managing multiple databases in SAP system environments. It manages different types and versions of database systems. Generally, it also enables the administration of non-SAP databases.

However, DB2 UDB for z/OS databases can only be managed if they are SAP databases. The Central Planning Calendar can be especially beneficial for managing MCOD landscapes.

To use the Central DBA Planning Calendar, choose one of the following:

- Call transaction DB13C
- Call transaction DB2- Section Performance Tuning- choose Central Planning Calendar.

Preparations for Using the Central DBA Planning Calendar

To manage a remote database using the Central DBA Planning Calendar, you need to set up an RFC connection to the system where the database is located using transaction SM59. The user specified for the RFC connection needs to have a valid TSO equivalent because the TSO user is used to submit scheduled JCL jobs.

Use DB13C, menu item configuration, to determine which systems to be displayed. Use for example ‘add systems’ to define the systems to be displayed in the calendar or use change or delete to update the list of systems.

Select Configuration → copy system if you want to add systems that are already defined in the repository. See the SAP online documentation for more details.

Refresh Display

To refresh the display of the system from which the Central DBA Planning Calendar has been called, choose Local on DB13C. The display for all remote systems can be refreshed using the Remote button. Three options are offered:

- Execute in Dialog (runs in dialog, takes a long time)
- Start in Background (runs immediately as background job)
- Schedule as Job (the scheduled jobs run daily at the specified time)

The execution state of the refresh jobs can be monitored on the local system by selecting ‘Display job log’.

Schedule Jobs

To schedule actions, you select a database. This implicitly calls the DBA Planning Calendar of the selected database.

You can only schedule DBA actions of the database of the selected system. If a DBA action has been specified, you are prompted to select the database systems of the same type for which you want to schedule this action as well. This facilitates the scheduling of identical DBA actions on multiple databases. You can update or delete planned actions for remote systems on the Central Planning Calendar.

The DBA Planning Calendar of each database is also capable of running on SAP systems that make use of a different database.

Monitor

You can monitor DBA actions for the database of the systems that have been registered using the Central Planning Calendar. The user interface of central and local DBA Planning
Calendar are very similar. The main difference is that the central versions is only for viewing DBA actions by systems (from here you can easily switch to local version). The local version shows actions for one system. You can schedule, change, delete or execute actions there.

The Central Planning Calendar shows only jobs that have been submitted from `DB13C - Local Calendar`. Jobs submitted directly on the local Planning Calendar of a remote system are NOT shown when using `DB13 - Local Calendar` for that system.

They can only be monitored using DB13 on the remote system directly. You can use DB13C, menu item “Calendar - Calendar Display” to customize the screen to your needs. The number of weeks to be displayed and the number of lines per week can be changed and saved.

For each day, DB13C provides a summary of the DBA actions of each registered system.

The number of actions and the number of actions with the highest status severity are displayed for each system. Status color-coding for each system indicates if actions have been executed successfully.

By double-clicking an entry, you can display all actions for the databases.

See the SAP online documentation for more information on the Central Planning Calendar.

**Functions**

You can monitor DBA actions for the databases that have been registered using the Central DBA Planning Calendar.

For each day, it provides a summary of the DBA actions of each database. By double-clicking an entry, you can display all actions for that database.

To schedule actions, you select a database. This implicitly calls the DBA Planning Calendar of the selected database.

You can only schedule DBA actions of the selected database. If a DBA action has been specified, you are prompted to select the database systems of the same type for which you want to schedule this action as well. This make is easier for you to schedule identical DBA actions on multiple databases.
Storage Management

This section covers the following topics:

- Changing the Database Layout [Page 295]
- Storage Parameters [Page 301]
- Space Management [Page 320]
46 Changing the Database Layout

The following sections provide you with information on how to change the database layout:

- Creating Tables and Indexes [Page 296]
- Modifying Tables and Indexes [Page 297]
- Moving Tables [Page 299]
- Rules for Self-Defined Objects [Page 300]

For information on how the DB2 database is organized in an SAP system environment, see Database Layout [Page 327].
46.1 Creating Tables and Indexes

Tables

When a new table is created, either by using transaction SE11 or during an upgrade or a transport, the SAP System automatically performs all the necessary steps (for example, creation of database and tablespace if needed). The algorithm depends on the actual buffering attributes of the table:

- Non-buffered tables
  The table is put into a newly created single-table tablespace to allow for individual monitoring and tuning.

- SAP buffered table
  If a multi-table tablespace (#SAP or XSAP) with less than 100 tables can be found with a matching database name (determined by data class, size category, buffering attributes) and page size, it is used. Otherwise a new multi-table tablespace XSAP in a different database is created.

In both cases, default storage parameters are taken from tables TADB2 and TGDB2 if they are not hard-coded within the SAP System. For multi-table tablespaces the primary and the secondary quantities obtained from table TGDB2 are multiplied by 100 and 20 respectively.

Primary Indexes

Primary indexes are considered to be an integral part of a table. Therefore, they can only be created and dropped with the associated table. The default storage parameters of an index are defined in tables IADB2 and IGDB2. Be aware that primary indexes are linked to their base table with an additional:

ALTER TABLE ... ADD PRIMARY KEY (...)

Secondary Indexes

If the base table already exists, the primary and secondary quantities are calculated from the respective values of the table's primary index or the table itself if there is no primary index. Otherwise, the default storage parameters are taken from tables IADB2 and IGDB2.
46.2 Modifying Tables and Indexes

Tables
If the column definitions of a table have been changed, either by a transport or in transaction SE11, the SAP activation process analyzes the modifications and subsequently decides how to act on the database table to achieve consistency between DDIC and DB. If a statement is feasible (for example, if a new table field needs to be added or a VARCHAR field is extended) it is executed. (A check before activation indicates if ALTER TABLE will be used or not.) Otherwise, the SAP System performs the following steps:

1. All non-partitioned indexes on the table are dropped.
2. All views on the table are dropped.
3. The original table is renamed by adding the prefix QCM (for example, TESTTAB becomes QCMTESTTAB).
4. A database and tablespace are provided.
5. The new DB table including its primary index are created according to the new DDIC definition of the table. A temporary table name is used to avoid that this table is erroneously accessed during the conversion. This name consists of a prefix (QCM8) and the first twelve characters of the table name (for example, QCM8TESTTAB).
6. The contents of the original table are transferred to the new table INSERT/SELECT adjusting types and formats.
7. The original table that has been renamed is dropped (including its database and tablespace if they become empty).
8. The newly created database table is renamed to its final name (QCM8TESTTAB → TESTTAB).
9. All secondary indexes are created.

This process is called table conversion.

Primary Indexes
Changes to the key definition of primary indexes are handled in the following way:

- Deleting a key field of a non-partitioned index
  A table conversion is triggered. This is done to ensure the uniqueness of the new primary index.
- Adding a key field of a non-partitioned index
  DROP and re-CREATE of the primary index.
- Partitioned index
  If the primary index is partitioned, every key modification (addition or deletion of a key field) triggers a table conversion.

Secondary Indexes
Changes in the definition of secondary indexes are limited to adding or deleting a key field. Two cases have to be distinguished:

- Non-partitioned indexes
  The modifications result in a DROP and subsequent re-CREATE of the index using the new key definition.
- Partitioned indexes
Partitioned indexes cannot be dropped. The SAP System creates an additional non-partitioned index with the same index ID but a different separator (either "\~" or "\^").

From now on, the SAP System considers the original partitioned index to be an external index, which is not defined in the DDIC.

If the partitioned index is:

- **non-unique**, no further action needs to be taken. The SAP system is able to handle a partitioned index that is not defined in the DDIC.

- **unique**, it puts a constraint on the table which is not known to the DDIC. Therefore, you should partition the table anew using another index that is defined in the DDIC (see "Move Table to Partitioned Tablespace"). Alternatively, you can trigger a conversion using transaction SE14 and choose *Edit → Force conversion*. The partitioned index is then recreated as a non-unique index with a new name: `<TABLENAME>~#`. (for example: `TESTTAB~#`)
46.3 Moving Tables

You can use tools available in the SAP System to move tables:

1. Call transaction SE14 and specify the table name.
2. Choose Edit.
3. Choose Goto → Storage parameter and proceed to the DB2 UDB for z/OS-specific part of transaction SE14.
4. Modify the storage parameters of the table and related objects (details on how to do this are given below).
5. Choose Goto → Back and return to the database-independent part of transaction SE14.
6. Specify the Processing Type and choose Extras → Force conversion to trigger a table conversion.

The SAP System provides tools to process many tables at a time, for example, to move empty tables to multi-table tablespaces. For more information, see Mass Processing [Page 314] in “Storage Parameters”.
46.4 Rules for Self-Defined Objects

The SAP system is able to handle additional stogroups (not listed in tables TADB2 and IADB2) and self-defined databases and tablespaces that contain SAP tables. It is for example possible to:

- Create a segmented or partitioned tablespace and move an SAP table to this new tablespace using DB2 means (SPUFI or an equivalent product) or transaction SE14.
- Combine a self-defined stogroup with a tablespace that contains SAP tables (using ALTER TABLESPACE and REORG)

However, you have to satisfy the following rules to guarantee full SAP functionality:

- The creator (also called schema) of all self-defined objects needs to be the one defined during the installation process. Until SAP release 4.6D, it was called SAPR3. Later installations employ the SAP<SAPSID>.
- When creating stogroups, use the following name ranges:
  - SAPY* and SAPZ* if the creator is SAPR3
  - <SAPSID>Y* and <SAPSID>Z* if the creator is not SAPR3 (for example, ABCYAD, ABCZ1D or ABCZ1I if the SAP System ID is ABC)
- For databases you should use the name ranges Y* and Z*.
- Use tablespace names different from #SAP or XSAP and create tablespaces in databases named Y* or Z*.
- Use the CCSID ASCII option when creating tables in non-Unicode systems. In Unicode systems employ the option CCSID UNICODE.
- Always check the consistency of tables, views, and indexes that you worked on using transaction SE14. Choose Extras → Database object → Check.

If you perform a homogeneous system copy using SAP tools (R3SETUP, R3load, R3ldctl, R3szchk), self-defined DB2 stogroups, databases, and tablespaces are not migrated to the target system. In that case, for all tables, tablespace and database, the naming convention is applied as described in Database Layout [Page 327] under Additional Information.
47 Storage Parameters

The following sections provide you with information on the following:

- General Overview [Page 302]
- Editing Storage Parameters [Page 303]
- Changing the Source of Storage Parameters [Page 305]
- Special Actions [Page 306]
- Directly Changing Storage Attributes [Page 307]
- Moving Tables to Existing Tablespaces [Page 308]
- Isolating and Combining Tables [Page 309]
- Partitioning Tables [Page 310]
- Handling Large Tables [Page 313]
- Mass Processing [Page 314]
- Incremental Conversion [Page 318]
- Troubleshooting [Page 319]
47.1 General Overview

Transaction SE14 provides means to modify and save storage parameters of tables, indexes, and tablespaces. To access the initial screen, choose one of the following:

- Call transaction SE14 and choose Edit → Storage parameter (for tables/tablespaces) or Edit → Indexes → Storage parameter (for indexes)
- Call transaction DB2 and choose Storage Mgt. Specify a table name or an index's base table and its index ID, and choose Edit.

In the following description, only the first access path is usually specified.

When the SAP system creates a new table and its database and tablespace, the storage parameters are taken from one of the following sources:

1. Saved parameters (abbreviated: SVD)
   These are storage parameters that have been edited and saved using transaction SE14. The SAP system uses table DDSTORAGE as the storage medium.

2. Status in the DB2 database (abbreviated: DBS)
   This parameter setting reflects the current situation on the database and is read from system tables such as SYSIBM.SYSTABLES, SYSIBM.SYSTABLESPACE, or SYSIBM.SYSTABLEPART.

3. Default (abbreviated: CMP)
   Some default parameters are defined in tables TADB2, TGDB2, IADB2, and IGDB2; others are hard-coded (for example, BUFFERPOOL or CCSID UNICODE).

The SAP system applies the following logic. If saved parameters (source: SVD) are available, they are used. Otherwise, the system uses the current status on the database (source: DBS) to determine the storage parameters of the target DB2 objects, in particular the tablespace. If the table does not exist, default values are taken. For indexes, the algorithm is similar. Saved storage parameters are always preferred to the current status in the database, which is preferred to the default values.
47.2 Editing Storage Parameters

To edit storage parameters, call transaction SE14.

Displaying Table Attributes

To display table storage attributes:

1. Choose Edit → Storage parameter.
   The initial screen displays important table parameters.
   - Name of tablespace and database
   - Estimated table size from RUNSTATS data (if the table exists and has been analyzed by RUNSTATS)
   - Partitioned index and limitkey information if the table is partitioned

   You can perform the following actions:
   - Collect limitkey data
     (Choose Goto → Limitkey data).
   - Specify a new tablespace
     (Choose Single-table, Partitioned, Multi-table, or Existing in edit mode).
   - Edit/display related DB2 objects
     (Choose Goto → Tablespace / Indexes / Partitioning index / LOB objects).
   - Display background jobs that have been started within transaction SE14
     (Choose Goto → Background jobs).

   See the following subsections for more information.

2. To return to the initial screen of transaction SE14, choose Goto → Back.
   If storage parameters have been edited, a dialog box asks whether your changes should be saved.

Switching to Edit Mode

In edit mode, you can change storage parameters. You can switch to edit mode by choosing Attributes → Display <-> Change.

Displaying/Editing Tablespace Parameters

To display or edit tablespace parameters:

1. Choose Goto → Tablespace.
2. Choose Goto → Back to return to the initial screen.

Displaying/Editing Index Parameters

To display or edit index parameters:

1. Choose Goto → Indexes.
   A list of all indexes defined on the table is displayed.
2. Select Index and choose Goto → Index.
3. Choose Goto → Back.
4. Choose Goto → Back to return to the initial screen.
Steps 2 and 3 are omitted if the table has only one index.

You can also directly access the screen that displays index parameters:
1. Call transaction SE14 and specify the table name.
2. Choose Edit.
3. Choose Indexes
4. Select the index.
5. Choose Continue.
6. Choose Goto → Storage parameter.

**Displaying/Editing Partitioned Index Parameters**

To display or edit partitioned index parameters:
1. Choose Goto → Partitioning Index.
2. Choose Goto → Back to return to the initial screen.

**Displaying/Editing LOB Objects**

To display or edit database objects that are related to a LOB table field (LOB tablespaces or auxiliary tables and their indexes):
1. Choose Goto → LOBs.
   A list of all LOB fields is displayed
2. Select one of the LOB fields (or LOB field partitions if the base table is partitioned).
3. Choose LOB tablespace or LOB index to display/edit storage attributes.
5. Choose Goto → Back to return to the initial screen.

After editing storage parameters within transaction SE14, you have to perform the following steps to adjust table, tablespace, database, and indexes in the database:
1. Choose Attributes → Save.
2. There are two ways to adjust the DB table using a table conversion.
   - Choose Attributes → Adjust in database to start a background job.
   - Choose Goto → Back and Extras → Force conversion. Here, processing type Direct means that online processing is also possible.

See the following sections for more information.
47.3 Changing the Source of Storage Parameters

Storage parameters of tables and indexes can be retrieved from different sources:

**Default Storage Parameters**

Choose Source → Default.

**Current Status in Database**

Choose Source → Status in database.

If the table does not exist, default parameters are displayed (source: CMP). The system also displays a size estimate of the table (pagesize \( \times \) number of pages determined by RUNSTATS), if RUNSTATS results are available.

**Saved Storage Parameters**

Choose Source → Saved.

If no parameters have been saved, the status on the database is displayed (source: DBS).
47.4 Special Actions

Triggering Table Conversions

1. On screen “DB2 Storage Attributes: Table” choose: Source → Saved. 
   The storage parameters that have been saved for this table are displayed. If no stored 
   parameters are displayed, you cannot initiate a table conversion.

2. To trigger the table conversion, choose Attributes → Adjust in database.

3. Specify the start time and choose Save.

4. To monitor the conversion job, choose Goto → Background jobs.

Displaying Current Background jobs

Within transaction SE14 you can start the following background jobs:

- Table conversion (see last subsection)
  Job name: DB-TABL[TABLE NAME]
- Collection of limitkey data (described below)
  Job name: DB2-[TABLE NAME]>[INDEX ID]>[NUMPARTS]

Transaction SE14 provides a direct link to transaction SM37, which displays the status and 
the logs of background jobs:

1. Choose Goto → Background jobs.

2. Choose Goto → Back to return to the initial screen.

Displaying Object Logs

On screen DB2 Storage Attributes: Table or DB2 Storage Attributes: Index, choose Goto → 
Object log.

The most recent action log is displayed.

Incremental Conversion

On screen DB2 Storage Attributes: Table, choose Goto → Incremental conversion.

For more details on how to employ this feature, see section “Incremental Conversion” below.
47.5 Directly Changing Storage Attributes

Some storage attributes of tablespaces and indexes can be changed directly in the database without performing a table conversion, for example, primary or secondary quantity or stogroup. The actions are slightly different for tablespaces and indexes.

**Index**

1. Call transaction SE14 and specify the table name.
2. Choose *Edit*.
3. Choose *Indexes...*, specify index and choose *Storage parameters*.
4. Choose *Attributes → Display <-> Change*.
5. Modify index parameters.
6. Choose *Attributes → Save; Attributes → Adjust in database*.

**Tablespace**

1. Call transaction SE14 and specify the table name.
2. Choose *Edit* and *Storage parameters*.
3. Choose *Attributes → Display <-> Change; Goto → Tablespace*.
4. Modify tablespace parameters.
5. Choose *Goto → Back*.
6. Choose *Attributes → Save; Attributes → Adjust in database*.

For both indexes and tablespaces, the SAP System subsequently performs SQL statements of the form `ALTER INDEX` or `ALTER TABLESPACE`. Note that for most storage parameters you need to run the DB2 utility `REORG` in addition to adjust the data in the database. LOB tablespaces and indexes of auxiliary tables cannot be altered.
### 47.6 Moving Tables to Existing Tablespaces

To move a table to an already existing tablespace:

1. In edit mode, choose *Existing*.
   
   The source is now *REV* (Revised/Not Saved).

2. Specify the names of the database and tablespace.

3. Choose *Attributes* → *Save* to save the target tablespace.

4. To trigger the conversion, choose *Attributes* → *Adjust in database*. 
47.7 Isolating and Combining Tables

The following actions work equally for tables in single-table, multi-table, and partitioned tablespaces.

A table should be moved to a single-table tablespace if it:

- Needs to be reorganized frequently
- Is accessed very often
- Becomes very large
- Grows rapidly in size.

Database performance would decline considerably if tables are kept in a multi-table tablespace in these cases.

On the other hand, tables in a single-table tablespace, which are only small and rarely accessed may be moved to multi-table tablespaces to simplify database administration.

Moving a Table to a Single-Table Tablespace

1. In edit mode, choose Single-Table.
   The source is now REV (Revised/Not Saved).
2. Edit tablespace and index storage parameters (optional).
3. Return to the initial screen.
4. Choose Attributes → Save to save the table parameters.
5. To trigger the conversion, choose Attributes → Adjust in database.

Moving a Table to a Multi-Table Tablespace

1. In edit mode, choose Multi-Table.
2. Edit tablespace and index storage parameters (optional).
3. Return to the initial screen.
4. Choose Attributes → Save to save the table parameters.
5. To trigger the conversion, choose Attributes → Adjust in database.
47.8 Partitioning Tables

To partition a table:

1. Check whether the planned partitioning key is accessed by any `UPDATE` statement within the SAP System:
   
   a. In transaction SE11 choose `Utilities → Where-used list`.
   b. Select `Programs` and continue.
   c. Search for the string "`UPDATE`" in the expanded where-used list.

   If fields of the partitioning key are modified in a `UPDATE` statement, each call of this statement locks the complete table. Therefore, you should consider using a different index for partitioning.

2. Collect limitkey data. (This step is optional.)

3. Move table to partitioned tablespace.

Steps 2 and 3 are described in more detail below.

Collecting Limitkey Data

This transaction enables to collect limitkey data, which can be used to partition a table. It performs an ordered select on the key fields of the chosen index and thereby determines a limitkey setting, which splits the table into parts of equal size.


   The latest limitkey data is displayed, if available.

2. Enter Index ID and number of partitions.

3. Choose: `Limitkey Data → Collect`.

4. Choose `Yes` in the dialog box `Start batch job to collect limitkey data?`.

5. Specify start time and choose `Save`.

6. To monitor the background job, choose `Goto → Background jobs`.

   Depending on the size of the table the runtime of the background job may range from a few minutes to several hours.

7. Choose `Goto → Back` to return to the initial screen.

Displaying Limitkey Data


   The latest limitkey data is displayed, if available.

2. Enter Index ID and number of partitions.

   Limitkey data is displayed, if available.

3. Choose `Goto → Back` to return to the initial screen.

Moving a Table to a Partitioned Tablespace

1. In edit mode, choose `Partitioned`.

   If the table is already partitioned, move on to Step 5.

2. Dialog box: `Use collected limitkey data?`
Changing the Limitkey Specifications of a Partitioned Table

1. Call transaction SE14 and choose Indexes... and select the partitioning index.
2. Choose Storage parameters and enter the DB2 specific part of transaction SE14.
3. Choose Attributes → Display <-> Change .
4. Modify the limitkey specifications below column Values .
5. Choose Attributes → Save .
6. Choose Attributes → Adjust in database . The SAP System executes SQL statements of the form:
   
   \[
   \text{ALTER INDEX ... PART ... VALUES (...)}
   \]
   
   to adjust the limitkey specifications in the database.
7. Execute the DB2 utility REORG on the partitioned tablespace to adjust the table data according to the new limitkey specifications.

Changing the Number of Partitions

1. In edit mode, edit the number of partitions and the limitkey specifications.
2. Choose Attributes → Save .
3. To initiate the table conversion, choose Attributes → Adjust in database .
4. Execute the DB2 utility RUNSTATS on the partitioned tablespace.
Deleting Limitkey Data

1. Choose Goto → Limitkey Data.
2. Choose Limitkey Data → Delete.
3. Answer Yes in the dialog box Limitkey data will be deleted! Continue?.

Subsequently, all the table’s limitkey data that has been collected so far is deleted.
47.9 Handling Large Tables

If a table needs to be partitioned or isolated, you should use DB2 utilities to move the data if the table:

- Has more than 1 million entries
- Is larger than 100 MB

In these cases a standard and even an incremental conversion would take too much time. (Both procedures use SQL SELECT-INSERT to perform the data transfer.) Therefore perform the following steps:

1. Determine the number of rows:
   a. Enter the table name in transaction SE16 and choose **Table → Table contents**.
   b. Choose **Edit → Number of entries**.

2. Create a quiesce point for the entire subsystem (for example, with -STOP DB2). Also, recommended are previous full image copies for all SAP data, to accelerate a RECOVERY if necessary.
   Make sure that afterwards no write accesses to the subsystem are carried out apart from the actions described here.

3. In transaction SE14, choose **Edit → Storage parameters** to modify the table’s storage parameters (for details, see the preceding sections). **Do not initiate a table conversion!**

4. Save the new storage parameters (choose **Attributes → Save**).

5. Copy the table’s contents to a sequential data set using DB2 utilities (for example, DSNTIAUL or REORG with option UNLOAD EXTERNAL).

6. Recreate the table (**Activate and adjust database** with option Delete data in transaction SE14). The SAP System uses the saved storage parameters when creating database, tablespace, table, and indexes.

7. Reload saved table contents into the newly created table (option: **LOG NO**).

8. As a first backup carry out a full image copy of the tablespace. This also cancels the Copy Pending.

9. Check the number of entries using transaction SE16 (see Step 1). status.
47.10 Mass Processing

The SAP System provides a DB2 specific mass processing tool.

To access the tool, choose one of the following:

- Call transaction SA38 and execute program RSDB2MAS
- Call transaction DB2 and choose Storage Mgt → Mass processing.

The following input screen appears:

![DB2/390: Mass Processing](image)

The input values of RSDB2MAS fall into two categories:

- **Filter**
  
  Here you specify filter criteria:

  - Maximum number of hits limits the number of tables selected for subsequent processing
  - Within Table you can specify selection criteria for the table name
  - The radio buttons Empty, Isolated in own tablespace, Buffered by R/3, Tablespace data set exists, and Index data set exists provide additional selection criteria (wild card "*", Yes and No are possible options; if you choose "*" the related selection criteria is not considered).

- **Intended action**
  
  Within this area the intended action has to be chosen. For instance, if you want to isolate tables in a tablespace of their own, choose single-table tablespace.

  You can also specify the usage of DEFINE YES/NO. This is a DB2 storage attribute, which allows users to create tablespaces or indexes with an option to defer the physical creation of underlying VSAM data sets until the very first write (that is, SQL Insert or LOAD utility).
If you choose *Reduce number of databases*, the system recreates single-table tablespaces in existing databases (up to 100 tablespaces in accordance to the naming convention). This option is helpful if you plan to add SAP systems to an existing installation (MCOD installation) and need to overcome the limitations of the old database layout (1:1 relationship between database and tablespace with a maximum number of 65279 databases per DB2 subsystem).

You should choose the option *Generate dependent reports after conversion* if you want to avoid bothering the users with a long report generation process after every mass conversion.

After specifying the input values, choose *Program → Execute*. Subsequently, the SAP System lists tables that meet the selection criteria specified within the input screen.

The five columns *E*, *I*, *B*, *T*, and *X* correspond to the input criteria *Empty*, *Isolated in own tablespace*, *Buffered by R/3* and *Tablespace data set exists*, and *Index data set exists*. Entry *X* in one of these five columns means Yes.

The *Status* column lists the conversion progress, once a mass processing request has been entered. Also, information is displayed on whether a table is partitioned or currently processed by ICNV. The following actions are possible:

- **Function key F2**: Displays a table’s storage attributes
- **Schedule requests**: Enters selected tables for mass processing with the intended action (here: recreation in tablespace of same type with storage parameter *DEFINE NO*).
Schedule requests triggers a standard table conversion with the Database Utility (transaction SE14). Therefore you have to make sure that the scheduled tables are smaller than 50 MB and that they are not accessed by any other user during the conversion process. Otherwise, consider performing an Incremental Conversion (details below).

The tool automatically activates all reports associated with the selected tables. This way, you do not have to get by with long-running compile processes afterwards.

- **Delete requests**: Delete selected tables from mass processing. This is only possible for tables whose conversion has not yet started.
- **Add to ICNV**: Add selected tables to the worklist of the Incremental Conversion (ICNV). For subsequent processing choose Goto → Incremental conversion or call transaction ICNV. For details, refer to section "Incremental Conversion" below or to the online documentation within transaction ICNV.
- **Goto → Database log**: The latest database log of the selected table is displayed.
- **Goto → Database utility (or function key F2)**: Either the Database Utility (transaction SE14) or the Incremental Conversion (transaction ICNV) is accessed.
- **Goto → General mass processing**: Calls the mass processing screen within transaction SE14.
- **Goto → Incremental conversion**: Calls transaction ICNV (Incremental Conversion).
- **Goto → Log overview**: Calls utility to display all logs.
- **Goto → Background jobs**: Displays background jobs related to RSDB2MAS.

**Troubleshooting**

RSDB2MAS issues system log message D40 (**RSDB2MAS: conversion of table xxx cancelled**) if the conversion of a table fails. In that case, call transaction SE14 and choose DB requests → Terminated to analyze and resolve the problem.

**Reducing the Number of Data Sets**

RSDB2MAS is particularly useful if you want to reduce the number of VSAM data sets. The following procedure applies for empty tables **isolated in their own tablespace**.

1. Call transaction SA38 and execute report RSDB2MAS.
2. Specify:
   
   Maximum number of hits    500
   Table         (empty)
   Empty         YES
   Isolated in own tablespace  YES
   Buffered by R/3      *
   Tablespace data  exists   YES
   Index data set exists   *

3. Choose:
   
   o **Recreate in tablespace of same type**
   o with **DEFINE NO**
   o and **Reduce number of databases**
4. The SAP System lists up to 500 empty tables with existing tablespace data sets. Select all tables and choose Schedule requests to enter them for mass processing. Subsequently, all selected tables are recreated using the storage attribute DEFINE NO. The system accumulates up to 100 single-table tablespaces into one database.

5. Repeat steps 1 to 4 until the list of selected tables is empty.

For tables in **multi-table tablespaces** a different procedure should be applied:

1. Call transaction SA38 and execute report RSDB2MAS.

2. Specify:
   
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of hits</td>
<td>100000</td>
</tr>
<tr>
<td>Table</td>
<td>(empty)</td>
</tr>
<tr>
<td>Empty</td>
<td>YES</td>
</tr>
<tr>
<td>Isolated in own tablespace</td>
<td>NO</td>
</tr>
<tr>
<td>Buffered by R/3</td>
<td>*</td>
</tr>
<tr>
<td>Tablespace data exists</td>
<td>*</td>
</tr>
<tr>
<td>Index data set exists</td>
<td>YES</td>
</tr>
</tbody>
</table>

3. Choose:
   
   - Recreate in tablespace of same type
   - with DEFINE NO

4. The SAP System lists up to 500 empty tables located in multi-table tablespaces. Select all tables and choose Schedule requests to enter them for mass processing. Subsequently, all selected tables are recreated using the storage attribute DEFINE NO. In most cases only the indexes are affected by the attribute DEFINE NO because these tables are recreated in existing tablespaces if possible.

5. Repeat steps 1 to 4 until the list of selected tables is empty.
**47.11 Incremental Conversion**

In contrast to the standard table conversion within transaction SE14 the incremental table conversion (ICNV) permits that tables can still be used by the system throughout the data transfer. As a result, there is much more time available for the data transfer and much larger sets of data can be converted. Operation in production mode is only disturbed briefly when you initialize the conversion and when you switch to the new table. There are two ways to access this tool:

- On screen DB2 Storage Attributes: Table, choose Goto → Incremental conversion.
- Call transaction ICNV.

Online documentation on how to perform an incremental conversion is provided within the initial screen of transaction ICNV (info button). The main steps are as follows:

1. Within transaction SE14 edit and save the table’s storage attributes.
2. Call transaction ICNV and choose Edit → Add table to add a table to the ICNV work list.
3. Subsequently, perform the actions described in the online documentation (choose Short description → Ad hoc).
47.12 Troubleshooting

If a table conversion fails, there is no data loss. During the conversion process the original table is renamed (\(<\text{TABNAME}\> \rightarrow \text{QCM}<\text{TABNAME}\)>). The renamed table is only deleted after completing successfully all conversion steps.

The error processing should involve the following steps:

1. Analyze the problem
   a. Call transaction SE14. On the initial screen the following message is displayed: Request: "Adjust"
   b. Choose Analyze Adjustment to analyze the error.
   c. Check the ICLI.*.err file for SQL errors.

2. Continue adjustment
   To continue the conversion, you must perform the following steps.
   a. Remove the cause of the problem.
      In some cases, the storage parameters have to be adjusted (for instance limitkey specifications). This can be done by editing and resaving the storage parameters (source: SVD).
      Also, it is possible that a stogroup is full and additional volumes need to be added. Then ALTER STOGROUP ... ADD VOLUME (...) enables a continuation.
   b. Select Processing Type in transaction SE14 and choose Continue adjustment. Subsequently, the table conversion continues.
48 Space Management
48.1 Adding Volume Space

Full Stogroup

If a stogroup becomes full, any transaction that writes data to tables associated with that stogroup will fail and DB2 returns an appropriate SQLCODE. In that case, you need to add additional volume(s) to the full stogroup by executing the SQL statements:

ALTER STOGROUP<FULL_STOGROUP>
   ADD VOLUMES(<VOLID_NEW1>,<VOLID_NEW2>,...)

You need to specify the volume IDs <VOLID_NEW1>, <VOLID_NEW2>, and so on, of the volumes added.

Full Volume

You do not need to take immediate measures if a single volume becomes full without the associated stogroups running out of space since the stogroups then use alternative volumes. As a precaution, you may add additional volumes to the respective stogroups by changing the database layout as follows:

1. Find all stogroups pointing to the full volume. To do this, execute the SQL statement:

   SELECT SGNAME FROM SYSIBM.SYSVOLUMES
   WHERE VOLID=<VOLID_OLD>

   specifying the volume ID <VOLID_OLD> of the full volume.

2. Add additional volume(s) to all selected stogroups as described above in “Full Stogroup”.
# 48.2 Data Compression

In many cases, using data compression can significantly reduce the amount of DASD space necessary to store the data. Important points are listed below:

**To enable data compression, specify** `COMPRESS YES` on `CREATE` or `ALTER TABLESPACE`.

The default for SAP Unicode systems is automatically `COMPRESS YES`. This is set at installation time.

As the `LOAD` utility is not regularly used in the SAP System, you should run a `REORG` utility after each major tablespace population step such as R3Load or batch input. This reclaims the free space and restores clustering after a major data import.

The data is not compressed until a compression dictionary is built, which is done by the `LOAD` and `REORG` utilities only.

- **Data compression advantages include:**
  - Better DASD space utilization
  - Higher buffer pool hit ratios
  - Fewer I/Os
  - Fewer `getpage` operations

- **Data compression disadvantages include:**
  - Increased consumption of virtual storage
  - CPU overhead (in general)
  - Higher P-lock contention (data sharing only)

- **Define more freespace** (`PCTFREE` and `FREEPAGE`) for the compressed tablespaces that contain frequently updated tables. This will reduce page overflows and result in better performance and less frequent reorganizations.

The freespace induced by data compression does not apply to the indexspaces because indexes are not compressed.

- **Take extra care with the tables that have row lengths close to the page limit (4KB, 8 KB, 16KB, and 32KB).** In most cases, the contents of these rows are random bit sequences that do not compress well and it is likely that even after compression not more than one row will fit into a page.

- **To check whether compressing a tablespace will bring benefits, you can run** `DSN1COMP` on the tablespace.

- **You can determine how effective data compression is by using compression reports** (`DSN1COMP` before compression and a `REORG` report after compression) and catalog statistics (`PAGESAVE` in `SYSTABLEPART` and `PCTROWCOMP` in `SYSTABLES` and `SYSTABSTATS`).

- **As all the character data in the SAP database is defined with variable length,** the benefits of compressing the data might not be enough to justify the associated CPU overheads.
For more information, see the DB2 for z/OS documentation.
Additional Information

This section covers the following topics:

- Transaction Codes [Page 325]
- Database Layout [Page 327]
- Environment Variables [Page 344]
## 49 Transaction Codes

The following table gives an overview of transaction codes that are useful in the administration of R/3 on DB2 for z/OS.

### Transaction Codes

<table>
<thead>
<tr>
<th>Transaction Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB02</td>
<td>Tables and Indexes Monitor</td>
</tr>
<tr>
<td>DB03</td>
<td>Database Parameters</td>
</tr>
<tr>
<td>DB12</td>
<td>Backup Monitor</td>
</tr>
<tr>
<td>DB13</td>
<td>DBA Planning Guide</td>
</tr>
<tr>
<td>DB2C</td>
<td>DB2 Catalog Browser</td>
</tr>
<tr>
<td>DB2J</td>
<td>R/3 JES Interface</td>
</tr>
<tr>
<td>RZ20</td>
<td>Alert Monitor</td>
</tr>
<tr>
<td>S001</td>
<td>ABAP Workbench</td>
</tr>
<tr>
<td>SE11</td>
<td>ABAP Dictionary</td>
</tr>
<tr>
<td>SE14</td>
<td>Database Utility</td>
</tr>
<tr>
<td>SE16</td>
<td>Data Browser</td>
</tr>
<tr>
<td>SE17</td>
<td>General Table Display</td>
</tr>
<tr>
<td>SE30</td>
<td>ABAP Runtime Analysis (SQL statement compare)</td>
</tr>
<tr>
<td>SE38</td>
<td>ABAP Editor</td>
</tr>
<tr>
<td>SE80</td>
<td>Repository Browser</td>
</tr>
<tr>
<td>SE84</td>
<td>Repository Information System</td>
</tr>
<tr>
<td>SM04</td>
<td>Current users on current server</td>
</tr>
<tr>
<td>SM13</td>
<td>Update Records</td>
</tr>
<tr>
<td>SM21</td>
<td>System Log: Local Analysis</td>
</tr>
<tr>
<td>SM31</td>
<td>Table Maintenance</td>
</tr>
<tr>
<td>SM37</td>
<td>Background job list</td>
</tr>
<tr>
<td>SM39</td>
<td>Background Information</td>
</tr>
<tr>
<td>SM50</td>
<td>Work Process Overview</td>
</tr>
<tr>
<td>SM51</td>
<td>Server Overview</td>
</tr>
<tr>
<td>SPO1</td>
<td>Spooler</td>
</tr>
<tr>
<td>ST02</td>
<td>Tuning Summary</td>
</tr>
<tr>
<td>ST03</td>
<td>Workload Analysis</td>
</tr>
<tr>
<td>ST04</td>
<td>Database Performance Monitor</td>
</tr>
<tr>
<td>ST05</td>
<td>Trace Requests</td>
</tr>
<tr>
<td>ST06</td>
<td>Operating System Overview</td>
</tr>
<tr>
<td>ST08</td>
<td>Network Monitoring</td>
</tr>
<tr>
<td>ST10</td>
<td>Table Call Statistics</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>ST11</td>
<td>Trace Logs</td>
</tr>
<tr>
<td>ST22</td>
<td>ABAP Dump Analysis</td>
</tr>
<tr>
<td>STUN</td>
<td>Remote Database Monitoring</td>
</tr>
<tr>
<td>SU01</td>
<td>User Maintenance</td>
</tr>
<tr>
<td>SU50</td>
<td>View/set user defaults</td>
</tr>
<tr>
<td>SU53</td>
<td>Unauthorized transactions</td>
</tr>
</tbody>
</table>
50 Database Layout

The following sections provide you with information on the database layout:

- General Overview
  - Requirements [Page 329]
  - SAP Technical Settings [Page 330]
  - Basic Concept [Page 332]

- Mapping Between the ABAP Dictionary and DB2
  - DB2 Objects [Page 334]
  - Examples [Page 338]
  - Compatibility [Page 340]

- Mapping Between the Java Dictionary and DB2 [Page 342]
50.1 General Overview

This section covers the following topics:

- Requirements [Page 329]
- SAP Technical Settings [Page 330]
- Basic Concept [Page 332]
50.1.1 Requirements

The SAP system requires that the structure and the names of transparent tables and views are identical in the ABAP Dictionary (DDIC) and in the DB2 database (DB). Also, index key definitions must be the same in both. The names of DB indexes have to be formed by concatenating the table name, a separator (either "^" or "~"), and the SAP index name.

There are no other specific SAP-specific requirements on how to name or organize the associated DB2 objects tablespace, database, and stogroup. In principle, all tables could be created with implicit tablespaces in the default database DSNDB04 using stogroup SYSDEFLT. However, this would have a negative impact on performance and database administration.

Additional requirements have to be taken into consideration.

- General requirements
  - High-end performance
  - Efficient monitoring and administration
  - Unlimited use of DB2 code pages

- DB2 requirements
  - Not more than 100 tables per database, considered optimal in terms of performance and operation
  - Efficient application of DB2 utilities, such as RUNSTATS, REORG, and COPY
  - Flexible and simple naming convention that is non-destructive to customer modifications

- z/OS requirements
  - If the Storage Management Subsystem (SMS) manages the extensions of DB2 data sets the following requirements are important:
    - To support SMS, the following z/OS data set names of tablespaces and indexspaces should be used as carriers of storage-relevant SAP system information. For more information, see the subsection "SAP Technical Settings".
      - [VCAT].DSNDBD.[DATABASE].[TABLESPACE].[SUFFIX]
      - [VCAT].DSNDBD.[DATABASE].[INDEXSPACE].[SUFFIX]
      - If the SAP system needs to move a table to a newly created tablespace, the new database and tablespace names should be such that there is no need to change data set qualifiers in the ACS routines.

The above requirements lead to the database layout [Page 327].
SAP Technical Settings

The SAP technical settings of a table allow you to categorize the table according to its space requirements (usage/growth), I/O rates, and so on. You can specify the following technical attributes using transactions SE13 or SE11:

- **Data class (TABART)**

  The data class is a table’s attribute that describes the use of a table. For example, it allows you to differentiate between a table with master data and a table with transactional data. Each data class has an associated 2-character ID (Storage ID) that is used internally by the SAP System to construct the names of DB2 databases. The data class is also directly mapped to DB2 stogroups.

  Table 1 lists available data classes and the corresponding storage IDs and DB2 stogroups.

- **Size category (TABSIZE)**

  The size category (between 0 and 4) describes the estimated space requirements of a table. A higher number identifies a larger table.

- **Buffering**

  Table buffering in an SAP system is one of the most important performance tuning mechanisms. Under certain circumstances tables are stored locally on each application server. The time consuming process of repeatedly accessing the database is thus avoided. The following types of buffering are possible:

  - **Full buffering**
    
    In case of read access, the entire table is loaded into the SAP buffer.

  - **Generic buffering**
    
    The generic key is defined as a part of the primary key. During the access of a record from a table that is generically buffered, all records whose generic key fields correspond to this record are loaded on the application server.

  - **Single-record buffering**
    
    Only those records actually being accessed are loaded on the application server.

  There is no relation at all between buffering in SAP’s buffering on the application server and DB2’s buffering using buffer pools.

<table>
<thead>
<tr>
<th>Data Classes (schema SAPR3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Class (TABART)</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td><strong>Customer data</strong></td>
</tr>
<tr>
<td>APPL0</td>
</tr>
<tr>
<td>APPL1</td>
</tr>
<tr>
<td>APPL2</td>
</tr>
<tr>
<td>USER</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>USER1</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>System data</td>
</tr>
<tr>
<td>CLUST</td>
</tr>
<tr>
<td>POOL</td>
</tr>
<tr>
<td>SDIC</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>SDOCU</td>
</tr>
<tr>
<td>SLOAD</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>S PROT</td>
</tr>
<tr>
<td>SLDEF</td>
</tr>
<tr>
<td>SLEXC</td>
</tr>
<tr>
<td>SSDEF</td>
</tr>
<tr>
<td>SS EXC</td>
</tr>
<tr>
<td>SSRC</td>
</tr>
<tr>
<td>TEMP</td>
</tr>
</tbody>
</table>
50.1.3 Basic Concept

The mapping between tables defined in the ABAP Dictionary and DB2 tables (plus related objects: tablespace, database, and stogroup) is governed by the following basic rules that are applied during SAP system installation and runtime:

- A table that **is not** SAP buffered at creation time is placed into a dedicated, single-table tablespace.
- A table that **is** SAP buffered at the time of creation is placed into an existing or a new multi-table tablespace, which will hold a maximum of 100 tables.
- A table and its indexes belong to the two separate DB2 storage groups that correspond to the table’s data class (*TABART*).
- A database holds a maximum of 100 tables located in either one multi-table tablespace or up to 100 single-table tablespaces.

For information on the naming and the storage parameters of DB2 objects, see the following section.
50.2 Mapping Between the ABAP Dictionary and DB2

The naming convention described in the following is illustrated for an exemplary list of tables and related DB2 objects in the subsection "Examples". See the graphic "Structure of Database Layout" and the table "Naming Convention (Examples)".
50.2.1 DB2 Objects

Stogroups

In this documentation, the term **stogroup** is used to refer to a DB2 storage group. This avoids any confusion that could arise due to using the same term, **storage group**, in two related products, DB2 and Data Facility Storage Management Subsystem (DFSMS).

A DB2 stogroup identifies a list of DASD volumes that hold data stored in DB2 tablespaces and indexspaces. For a given stogroup the corresponding volume names are either stored in the DB2 catalog or associated to the stogroup by means of DFSMS ACS routines.

There is a fixed number of stogroups defined for an SAP System. They are closely related to the data class (TABART). The relation is static and specified in the SAP tables TADB2 and IADB2. Two cases have to be distinguished:

- If the creator/schema associated with the SAP System is SAPR3 the stogroup entries in table TADB2 and TGDB2 are used without any change.
- Otherwise, the first 3 characters SAP of the stogroup entries in tables TADB2 and TGDB2 are substituted by the SAP system ID <SAPSID> when generating a CREATE TABLESPACE or CREATE INDEX statement. For instance, if the <SAPSID> is ABC some of the stogroups used are ABCBTD, ABCCLI, or ABCU1D.

All DB2 stogroups used by an SAP System are associated with the same Integrated Catalog Facility Catalog (parameter VCAT within CREATE STOGROUP statement).

During runtime no additional stogroups are created. One or more data classes are assigned to a pair of stogroups. One is designated for tables (suffix "D" for data), and one for indexes (suffix "I"). This allows you to assign separate sets of volumes for data and indexes to avoid DASD contention. See the "Data Classes" table for a complete list of the DB2 stogroups used by the SAP System.

The naming convention can be used in your ACS routines if you use SMS-managed data sets. In that case, SAP and IBM recommend that you add the stogroup's own name to its list of volumes using:

```
ALTER STOGROUP <STOGRP> ADD VOLUME (<STOGRP>)
```

This is possible because DB2 does not check the existence of the volumes in the VOLUMES clause. That way you pass the stogroup name to the ACS routine, which effectively means passing the information on whether the subject data set is going to store data or indexes.

**Databases**

In the environment of an SAP system installation, **DB2 database** is an object that consists of a DB2 tablespace and all the indexes on the tables contained in the tablespace. Specifically, for a single-table tablespace, the database includes one table and all its indexes.

There are only a few DB2 commands for database control: DISPLAY DATABASE, STOP DATABASE, START DATABASE. The real value of this DB2 object in the SAP system environment is in its name; it carries all the technical settings of the contained tables. For a given table the associated database name is constructed as follows:

```
[STORAGEID][TABSIZE][BUFFER]X[ABC]
```

where the parameters have the following meaning:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>[STORAGEID]</td>
<td>SAP data class identified by 2 characters A0, A1, A2, and so on. The unique relation between data class and...</td>
</tr>
</tbody>
</table>
storage ID is defined in the SAP table `TADB2` and described in the "Data Classes" table.

<table>
<thead>
<tr>
<th>TABLESIZE</th>
<th>SAP size category (1 digit: 0,1,2,3,4)</th>
</tr>
</thead>
</table>
| BUFFER             | Actual buffering on the application server at creation time (1 digit):
|                    | - 0 = no buffering
|                    | - 1 = full buffering
|                    | - 2 = generic buffering
|                    | - 3 = single-record buffering |
| X                  | Placeholder for future developments (earlier releases use '#' as a placeholder) |
| [ABC]              | 3 random characters ([A-Z][0-9]) |

With the naming convention in mind you can determine a table’s technical settings by looking at the appropriate DB2 catalog table (`SYSIBM.SYSTABLES`), or, more importantly, at the underlying data set name (remember that the database name is a constituent part of the underlying data set name). That allows you to control the physical placement of tables and indexes, for example, by passing the technical setting details to an ACS routine.

**Tablespaces**

DB2 tablespace is a common name for one or more data sets used to store a single or multiple tables. In an SAP system installation, the SAP type of buffering on the application server is the only criterion for deciding whether a tablespace stores a single or multiple tables. If a given table is not SAP buffered at creation time, it is stored in a dedicated tablespace that does not contain any other table. Otherwise, the table is put in multi-table tablespace.

The reasons for this division are as follows:

- In principle, all tables that are not buffered on the application server are likely to be accessed frequently via `INSERT` and `UPDATE`. Each of them is a possible candidate for tuning efforts. However, efficient DB2 monitoring and tuning requires that the table is isolated in its own tablespace. Since isolating a large table means down-time and considerable administrative effort, SAP and IBM decided to deliver all non-buffered tables in single-table tablespaces.

- If a table is buffered on the application server, there are only rare DB read accesses. Furthermore, inserts and updates are rather unlikely. They are combined in multi-table tablespaces because DB2 monitoring and tuning of these tables is either less important or of no use.

A multi-table tablespace holds up to 100 tables. The name "XSAP" is assigned to this type of tablespace (up to SAP R/3 Release 4.5B the default name "#SAP" was used.) For a single-table tablespace, the system determines the tablespace name by taking the first seven characters of the associated table name. Naming space qualifiers "/. . ./" are ignored (naming qualifiers are used by SAP and partner companies for add-on products). The character "_" is replaced by "X".

The database and tablespace name of a table in a single-table tablespace are only slightly modified if the table needs to be converted due to structural changes, for example, during an upgrade. In that case, the character "X" is appended to the tablespace name. Therefore, if you use the qualifier

```
[VCAT].DSNDBD.[DATABASE].[TABLESPACE]*.[SUFFIX]
```

in your ACS routines, there is no need to adjust these routines after a table conversion.
DSN00X.DSNDBD.U140XPOI.OURTAB*.I0001.A001

for table /IBM/OURTAB listed in the "Naming Convention (Examples)" table uniquely finds the two alternatives:

DSN00X.DSNDBD.U140XPOI.OURTAB.I0001.A001
DSN00X.DSNDBD.U140XPOI.OURTABX.I0001.A001

Both multi- and single-table tablespaces are initially defined using the default storage parameters:

- LOCKSIZE ROW
- LOCKMAX 1000000
- CCSID ASCII or CCSID UNICODE
- COMPRESS YES (Unicode only)

The following parameters depend on the table’s page size:

**Page Sizes and Their Parameters**

<table>
<thead>
<tr>
<th>Page Size</th>
<th>BUFFERPOOL</th>
<th>PCTFREE</th>
<th>FREEPAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4KB</td>
<td>BP2</td>
<td>20</td>
<td>16</td>
</tr>
<tr>
<td>8KB</td>
<td>BP8K0</td>
<td>10</td>
<td>31</td>
</tr>
<tr>
<td>16KB</td>
<td>BP16K0</td>
<td>5</td>
<td>31</td>
</tr>
<tr>
<td>32KB</td>
<td>BP32K</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

The parameter `SEGSIZE` is calculated from the table’s size category:

\[ SEGSIZE = 8 \times \text{TABKAT} + 4 \]

For more information, see the subsection "SAP Technical Settings".

Other default parameters, such as primary and secondary quantity, or stogroups are specified in the SAP system tables `TADB2` and `TGDB2`. For multi-table tablespaces the primary and secondary quantities in table `TGDB2` are multiplied by 100 and 20 respectively to meet larger space requirements.

The default primary quantity for tablespaces is always at least equal to

\[ \text{PAGESIZE} \times (2 \times \text{SEGSIZE} + 2) \]

to make sure that an empty tablespace allocates only the first extent.

**Indexspaces**

Each DB2 index occupies its own **DB2 indexspace** placed in the DB2 database that contains the base table of the index. Default values for stogroups or primary and secondary quantities are defined in table `IGDB2`. Other parameters are initially defined as follows:

- BUFFERPOOL BP3
- PCTFREE 10
- FREEPAGE 10
**LOB Objects**

Table columns that have the DB2 data types **CLOB** and **BLOB** are used within an SAP System. The following database objects are created for each LOB column and partition:

- **The LOB tablespace** is created in the base table’s database with the following name:
  \[L[TABNAME5][LK]\]
  
  With:
  - \[TABNAME5\] = first 5 characters of the table’s name
  - \[LK\] = 2 random characters ((A-Z)[0-9])

  **The default storage attributes are as follows:**
  - **STOGROUP** = table’s default data stogroup
  - **PRIQTY** 200 **SECQTY** 10240 **GBPCACHE** SYSTEM
  - **BUFFERPOOL** BP40 **LOG** YES **LOCKMAX** 1000000 **LOCKSIZE** LOB

- **For the auxiliary table** the following naming convention is applied:
  
  \#[[COLNAME14][MNO]]
  
  With:
  - \[COLNAME14\] = first 14 characters of the column’s name
  - \[MNO\] = 3 random characters ((A-Z)[0-9])

- **The index on auxiliary table** is created with the same name as the auxiliary table using the following storage parameters:
  - **STOGROUP** = table’s default index stogroup
  - **PRIQTY** 16 **SECQTY** 10240 **FREEPAGE** 10 **PCTFREE** 10 **GBPCACHE** CHANGED
  - **BUFFERPOOL** BP40 **PIECESIZE** 2097152 K

For example, if the non-partitioned table **LOBTSTTAB** has a LOB column named **TESTLOBCOL**, the naming of the related LOB objects could be as follows:

- **LOB tablespace** **LLOBTS5A**
- **Auxiliary table** \#TESTLOBCOL8ZH with index \#TESTLOBCOL8ZH

**Creator / Schema**

⚠️ The creator of each DB2 object belonging to an SAP System needs to be the same. The creator (or schema) has to be specified during the installation of a system (default is SAP<SID>).

Therefore, you need to initiate any DDL statement that deals with SAP objects and is submitted directly on the database (for example, using SPUFI) with:

```
SET CURRENT SQLID = '<SCHEMA>';
```

Always use the option **CCSID ASCII** (non-Unicode systems) or **CCSID UNICODE** (Unicode systems) when creating tables.
50.2.2 Examples

The following table and graphic illustrate the naming conventions outlined above for some exemplary tables.

XTAB represents the example of a table that has been converted at least once after creation. Therefore, the related tablespace's name contains the suffix "x".

Naming Convention (Examples)

<table>
<thead>
<tr>
<th>Table</th>
<th>Data Class</th>
<th>Size</th>
<th>Buffered on application server?</th>
<th>Database</th>
<th>Tablespace</th>
<th>Stogroups</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYTABLE</td>
<td>USER1</td>
<td>0</td>
<td>0 (no)</td>
<td>U100X3HJ</td>
<td>MYTABLE</td>
<td>SAPU1D</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SAPU1I</td>
</tr>
<tr>
<td>TABLE_1</td>
<td>USER1</td>
<td>0</td>
<td>0 (no)</td>
<td>U100X2HA</td>
<td>TABLEX1</td>
<td>SAPU1D</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SAPU1I</td>
</tr>
<tr>
<td>OURTABLE</td>
<td>APPL0</td>
<td>4</td>
<td>2 (generic)</td>
<td>A042XKJH</td>
<td>XSAP</td>
<td>SAPATD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SAPATI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XTAB</td>
<td>APPL2</td>
<td>4</td>
<td>0 (no)</td>
<td>A240XF23</td>
<td>XTABX</td>
<td>SAPPOD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SAPPOI</td>
</tr>
<tr>
<td>YTAB</td>
<td>APPL2</td>
<td>2</td>
<td>3 (single)</td>
<td>A223XFGH</td>
<td>XSAP</td>
<td>SAPPOD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SAPPOI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TADIR</td>
<td>SDIC</td>
<td>3</td>
<td>1 (fully)</td>
<td>DI31X23J</td>
<td>XSAP</td>
<td>SAPDID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SAPDII</td>
</tr>
<tr>
<td>/IBM/OURTAB</td>
<td>USER1</td>
<td>4</td>
<td>0 (no)</td>
<td>U140XPOI</td>
<td>OURTAB</td>
<td>SAPU1D</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SAPU1I</td>
</tr>
</tbody>
</table>

Table (continued) Data Set Names (Catalog Name DSN00X)

<table>
<thead>
<tr>
<th>Table</th>
<th>Data Set Names (Catalog Name DSN00X)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYTABLE</td>
<td>DSN00X.DSNDBD.U100X3HJ.MYTABLE.I0001.A001</td>
</tr>
<tr>
<td>TABLE_1</td>
<td>DSN00X.DSNDBD.U100X2HA.TABLEX1.I0001.A001</td>
</tr>
<tr>
<td>OURTABLE</td>
<td>DSN00X.DSNDBD.A042XKJH.XSAP.I0001.A001</td>
</tr>
<tr>
<td>XTAB</td>
<td>DSN00X.DSNDBD.A240XF23.XTABX.I0001.A001</td>
</tr>
<tr>
<td>YTAB</td>
<td>DSN00X.DSNDBD.A223XFGH.XSAP.I0001.A001</td>
</tr>
<tr>
<td>TADIR</td>
<td>DSN00X.DSNDBD.DI31X23J.XSAP.I0001.A001</td>
</tr>
<tr>
<td>/IBM/OURTAB</td>
<td>DSN00X.DSNDBD.U140XPOI.OURTAB.I0001.A001</td>
</tr>
</tbody>
</table>

Structure of Database Layout
50.2.3 Compatibility

Upgraded Systems

The database layout described here differs considerably from the layout implemented for SAP R/3 releases earlier than 4.5A. If your system is upgraded from SAP R/3 Release 4.0B or lower, the new layout is only applied to:

- New tables
- Existing tables in multi-table tablespaces that need to be recreated due to structural changes

If an already isolated table and its tablespace have to be recreated, it is kept isolated even if the table is SAP buffered or the tablespace and database name are different from the 4.5A naming convention. All storage parameters of the original tablespace are reused. The new database and tablespace names are formed by adding or substituting "x" as the last character. Thus, all administrative efforts (extending primary and secondary quantities, compressing, tuning, and so on) are preserved.

Name Range Occupied by SAP Systems

If there are various applications running in the same DB2 subsystem, it is important to know the name range occupied by an SAP System:

- Creator
  The creator of all DB2 objects created and used by the SAP System is the same.
- Stogroups
  In CREATE TABLESPACE and CREATE INDEX statements, the SAP System only uses stogroups that are listed in tables TADB2 and IADB2.
  When changing storage parameters of tablespaces in transaction SE14 you may also employ self-defined stogroups.
- Databases
  The first two characters of a new database name are always taken from column STORAGEID in table TADB2. Tables and indexes are placed exclusively into databases and tablespaces that were created by the SAP Systems dedicated creator.

Customers can add new data classes, storage IDs, and stogroups to the SAP tables TADB2 and IADB2. For details, see SAP Notes 46272 and 163449 and Rules for Self-Defined Objects [Page 300].

Storage Parameter Define No

The installation and upgrade processes create tablespaces and indexes with option DEFINE NO. That means that the underlying data sets are not created until the first row is inserted into the corresponding table. For most SAP system installations sites a large number of tables remain empty, which means that a significant number of data sets will not be created. This is beneficial for many database administration tasks as well as for the DASD space utilization.

After installation or upgrade, the DB2-specific mass processing tool RSDB2MAS allows you to recreate tablespaces and indexes with storage option DEFINE NO. For more information, see Mass Processing [Page 314].

⚠️

The objects created with DEFINE NO are fully supported by all the functions within the SAP System.
Potential problems may arise if you use third party tools that do no have support for this object. Check with your tool provider if the tools implement the necessary support.
50.3 Mapping Between the Java Dictionary and DB2

SAP J2EE uses Unicode. Therefore, all database objects are created with CCSID UNICODE. UTF 16 is used to store Unicode data, in other words, all character data is stored as (VAR)GRAPHIC.

Note that with DB2 for z/OS, Unicode and non-Unicode tables can be mixed within one subsystem. Therefore, tables for SAP Web AS Java persistence can be created in the subsystem used to store the non-Unicode (or Unicode) objects of the traditional SAP Web AS ABAP part of the SAP system.

Stogroups

One stogroup is used per SAP instance and created during the installation. The name of this stogroup schema is identical to the schema used for Java persistence. The naming convention for this is SAPDB<SID>.

Database and Tablespaces

The SAP Java Dictionary uses a 1-to-1 relation between tables and tablespaces and a 100-to-1 relation between tablespaces and databases. In other words, up to 100 tablespaces are created in one database, and all tablespaces are single-table tablespaces.

With the SAP J2EE Engine, all objects are created within a dedicated namespace. To keep the relation between SAP namespaces and database objects transparent, the naming convention for databases is:

<NAMESP>XX<RANDOM>

where <NAMESP> is an abbreviation for the first three letters of the SAP namespace (for example "J2E") and <RANDOM> is an abbreviation for a three character random string.

The name of the tablespace is derived from the table name by removing the namespace prefix and truncating it to 8 characters (uppercase).

By default, the following tablespace attributes are used:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUFFERPOOL</td>
<td>according to page size</td>
</tr>
<tr>
<td>FREEPAGE</td>
<td>20</td>
</tr>
<tr>
<td>PCTFREE</td>
<td>16</td>
</tr>
<tr>
<td>GBPCACHE</td>
<td>CHANGED</td>
</tr>
<tr>
<td>DEFINE</td>
<td>YES</td>
</tr>
<tr>
<td>MEMBER_CLUSTER</td>
<td>not specified</td>
</tr>
<tr>
<td>LOCKSIZE</td>
<td>ROW</td>
</tr>
<tr>
<td>LOCKMAX</td>
<td>1000000</td>
</tr>
<tr>
<td>CLOSE</td>
<td>YES</td>
</tr>
<tr>
<td>COMPRESS</td>
<td>YES</td>
</tr>
<tr>
<td>MAXROWS</td>
<td>255</td>
</tr>
</tbody>
</table>

There are no predefined table categories in the SAP J2EE engine: All tables are created with the same settings and without explicit primary and secondary quantity. Instead, the DB2 V8 Automatic Extend Management is used, that is the ZPARM MGEXTSZ must be set to YES.
Indexes

By default the following index attributes are used:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUFFERPOOL</td>
<td>BP2</td>
</tr>
<tr>
<td>FREEPAGE</td>
<td>20</td>
</tr>
<tr>
<td>PCTFREE</td>
<td>16</td>
</tr>
<tr>
<td>GBPCACHE</td>
<td>CHANGED</td>
</tr>
<tr>
<td>DEFINE</td>
<td>YES</td>
</tr>
<tr>
<td>CLOSE</td>
<td>YES</td>
</tr>
<tr>
<td>DEFER</td>
<td>NO</td>
</tr>
<tr>
<td>COPY</td>
<td>YES</td>
</tr>
<tr>
<td>PIECESIZE</td>
<td>2097152K</td>
</tr>
</tbody>
</table>

LOB Objects

No explicit ROWID is used for tables containing LOB columns. BLOB fields are always created with 1 GB, CLOB fields with 0.5 GB.

Auxiliary objects to store the LOB data are created as they are in the ABAP stack with the following differences:

- The table name is always used without a namespace prefix.
- Primary and secondary quantities are not specified (DB2 Space Extend Management is used).
- The column name is used without truncation.
## 51 Environment Variables

The following table lists important environment variables needed in an SAP system on DB2 UDB for z/OS.

For additional information, call transaction RZ11, specify the profile parameter and choose Display.

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Profile Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DB2CODEPAGE</code></td>
<td><code>-</code></td>
<td>Must be set to code page 819. Only necessary in a non-Unicode system.</td>
</tr>
<tr>
<td><code>DB2DB6EKEY</code></td>
<td><code>-</code></td>
<td>Must be set so <code>dbdb2pwd</code> can set your key. To ensure that your settings are secure, you should change the value after installation. Encryption key length = 16 characters</td>
</tr>
<tr>
<td><code>dbms_type</code></td>
<td><code>dbms/type</code></td>
<td>Specifies database type (for example: DB2)</td>
</tr>
<tr>
<td><code>dbs/db2/con_profile</code></td>
<td><code>-</code></td>
<td>Connection profile for DB2 z/OS</td>
</tr>
<tr>
<td><code>dbs/db2/planname</code></td>
<td><code>-</code></td>
<td>Plan name for dynamic SAP database interface (only for application server on z/OS)</td>
</tr>
<tr>
<td><code>dbs/db2_ps</code></td>
<td><code>-</code></td>
<td>The DB2 Connect Collection ID should only be set in special cases.</td>
</tr>
<tr>
<td><code>dbs/db2_pw</code></td>
<td><code>-</code></td>
<td>The DB2 Connect password variable should only be set in special cases. The SAP system will normally generate a Collection ID. If you use <code>db2radm</code> to bind, the tool will automatically use the generated collection id.</td>
</tr>
<tr>
<td><code>dbs/db2_schema</code></td>
<td><code>-</code></td>
<td>Owner of database objects for this component</td>
</tr>
<tr>
<td><code>dbs/db2_ssid</code></td>
<td><code>-</code></td>
<td>You can either set one or the other. Alternatively, if you are working with the failover function, the parameters will be set according to the <code>connect.ini</code>.</td>
</tr>
<tr>
<td><code>dbs/db2_use_accounting</code></td>
<td><code>-</code></td>
<td>Must be set to 1 if you want the SAP application server to pass SAP attributes like ABAP program name and end user ID to DB2 at the granularity of transactions. DB2 feeds this information into DB2 accounting records and exploits it when creating WLM enclaves for application servers that connect via DRDA. Default value is 0.</td>
</tr>
<tr>
<td><code>dbs/db2_use_copy_yes</code></td>
<td><code>-</code></td>
<td>If set to 1, all indexes are created with storage attribute COPY YES. Default value is 0.</td>
</tr>
<tr>
<td>Environment Variable</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td><strong>dbs_db2_use_define_no</strong></td>
<td>If set to 1, all tablespaces and indexes are created with storage attribute DEFINE NO. Default value is 0.</td>
<td></td>
</tr>
<tr>
<td><strong>dbs/db2/use_define_no</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>dbs_db2_use_drda</strong></td>
<td>Must be set to 1 to use DB2 Connect. Default value is 0.</td>
<td></td>
</tr>
<tr>
<td><strong>dbs/db2/use_drda</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>dbs_db2_use_lockmax_system</strong></td>
<td>If set to 1, all tablespaces are created with storage attribute LOCKMAX SYSTEM, excluding those with LOCKSIZE TABLE or LOCKSIZE TABLESPACE. Default value is 0.</td>
<td></td>
</tr>
<tr>
<td><strong>dbs/db2/use_lockmax_system</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>dbs_db2_use_wlm</strong></td>
<td>Must be set to 1 if you want to let the SAP application server on z/OS create WLM enclaves at the granularity of transactions. Default value is 0.</td>
<td></td>
</tr>
<tr>
<td><strong>dbs/db2/use_wlm</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>dbs_db2_user</strong></td>
<td>Must be set.</td>
<td></td>
</tr>
<tr>
<td><strong>dbs/db2/user</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DIR_LIBRARY</strong></td>
<td>Full path name of Database Interface shared library</td>
<td></td>
</tr>
<tr>
<td><strong>DIR_LIBRARY</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SAPDBHOST</strong></td>
<td>TCP/IP address of database host</td>
<td></td>
</tr>
<tr>
<td><strong>SAPDBHOST</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SAPSYSTEM</strong></td>
<td>SAP system name</td>
<td></td>
</tr>
<tr>
<td><strong>SAPSYSTEMNAME</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STEPLIB</strong></td>
<td>STEPLIB (used only by application server on z/OS)</td>
<td></td>
</tr>
<tr>
<td><strong>STEPLIB</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>RSDB_RECO_SYNC_ALL_SERVER</strong></td>
<td>Activates synchronization of all application servers during database reconnect</td>
<td></td>
</tr>
<tr>
<td><strong>rsdb/reco_sync_all_server</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SQL_TRACE</strong></td>
<td>Switch for SQL trace Sets SQL trace</td>
<td></td>
</tr>
<tr>
<td><strong>dbs/db2/sql_trace</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Environment Variables (for Standby System)**

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RSDB_DB2HOSTSTANDBY</strong></td>
<td>Database host</td>
</tr>
<tr>
<td><strong>rsdb/db2_host_standby</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DBS_DB2_SSID_STANDBY</strong></td>
<td>Database Attach Name</td>
</tr>
<tr>
<td><strong>dbs/db2/ssid_standby</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DBS_DB2_PLANNAME_STANDBY</strong></td>
<td>Plan name (only for application server on z/OS)</td>
</tr>
<tr>
<td><strong>dbs/db 2/planname_standby</strong></td>
<td></td>
</tr>
</tbody>
</table>
If you need to bind and grant jobs, (for example, to change the plan name) adapt the following job templates to suit your needs:

The following bind template is for z/OS only. However, the grant template is valid for z/OS and other systems.

### Bind Template

```sql
//OCOP2   EXEC PGM=IKJEFT01,REGION=128M
//INHFS2   DD PATH='<DBRMPATH>/DBRM.db2sqlv6', PATHOPTS=(ORDONLY)
//OUTMVS2  DD DSN=&&DBRMDAT2(DB2SQLV6),UNIT=SYSDA,DISP=(NEW,PASS),
           RECFM=FB,LRECL=80,BLKSIZE=3120,SPACE=(3120,(10,10,5))
//SYSTSPRT DD SYSOUT=* 
//SYSTSIN  DD *
OCOPY INDD(INHFS2) OUTDD(OUTMVS2) BINARY PATHOPTS(USE) /*

//DBPAC2   EXEC PGM=IKJEFT01
//DBRMLIB  DD DISP=SHR,DSN=&&DBRMDAT2
//STEPLIB  DD DISP=SHR,DSN=<DB2_SDSNLOAD>
//SYSPRINT DD SYSOUT=* 
//SYSTSPRT DD SYSOUT=* 
//SYSTSIN  DD *
DSN SYSTEM(<SUBSYSTEM_NAME>)
   BIND PACKAGE(<PACKNAM6>) OWNER(<SCHEMA>) MEMBER(DB2SQLV6) -
       ACTION(REPLACE) -
       DYNAMICRULES(RUN) ISOLATION(UR) -
       CURRENTDATA(YES) RELEASE(COMMIT) KEEPDYNAMIC(YES)
END /*

//OCOP3   EXEC PGM=IKJEFT01,REGION=128M
//INHFS3   DD PATH='<DBRMPATH>/DBRM.db2sqlv8',
//          PATHOPTS=(ORDONLY)
//OUTMVS3  DD DSN=&&DBRMDAT3(DB2SQLV8),UNIT=SYSDA,DISP=(NEW,PASS),
           RECFM=FB,LRECL=80,BLKSIZE=3120,SPACE=(3120,(10,10,5))
//SYSTSPRT DD SYSOUT=* 
//SYSTSIN  DD *
OCOPY INDD(INHFS3) OUTDD(OUTMVS3) BINARY PATHOPTS(USE) /*
```
Grant Template

/****BEGIN OF BIND PLAN

//DBBIND EXEC PGM=IKJEFT01
//STEPLIB DD DISP=SHR,DSN=<DB2_SDSNLOAD>
//SYSPRINT DD SYSOUT=* 
//SYSTSIN DD *
DSN SYSTEM(<SUBSYSTEM_NAME>)
BIND PLAN(<PLANNAME>) OWNER(<SCHEMA>) -
PKLIST(*,<PACKNAM6>.*<*<PACKNAM8>.*<DSNUTILS>.*,<PACKNAM8>.*DSNACC.*) -
ACTION(REPLACE) RETAIN ISOLATION(UR) DYNAMICRULES(RUN) -
ACQUIRE(USE) RELEASE(COMMIT) -
CURRENTDATA(YES) KEEPDYNAMIC(YES)
END
*/

/**END OF BIND PLAN

//DBGRANT EXEC PGM=IKJEFT01
//STEPLIB DD DISP=SHR,DSN=<DB2_SDSNLOAD>
//SYSPRINT DD SYSOUT=* 
//SYSTSIN DD *
//**DRDA only
GRANT EXECUTE ON PACKAGE "<COLLID>".* TO <CONNECTOR>;
//**End DRDA only
//**z/OS only
GRANT EXECUTE ON PLAN <PLANNAME> TO <SIDADM>;

//DBPAC3 EXEC PGM=IKJEFT01
//DBRMLIB DD DISP=SHR,DSN=&&DBRMDAT3
//STEPLIB DD DISP=SHR,DSN=<DB2_SDSNLOAD>
//SYSPRINT DD SYSOUT=* 
//SYSTSIN DD *
DSN SYSTEM(<SUBSYSTEM_NAME>)
BIND PACKAGE(<PACKNAM8>) OWNER(<SCHEMA>) MEMBER(DB2SQLV8) -
ACTION(REPLACE) -
DYNAMICRULES(RUN) ISOLATION(UR) -
CURRENTDATA(YES) RELEASE(COMMIT) KEEPDYNAMIC(YES)
END
*/

//*BEGIN OF BIND PLAN

//DBBIND EXEC PGM=IKJEFT01
//STEPLIB DD DISP=SHR,DSN=<DB2_SDSNLOAD>
//SYSPRINT DD SYSOUT=* 
//SYSTSIN DD *
DSN SYSTEM(<SUBSYSTEM_NAME>)
BIND PACKAGE(<PACKNAM8>) OWNER(<SCHEMA>) MEMBER(DB2SQLV8) -
ACTION(REPLACE) -
DYNAMICRULES(RUN) ISOLATION(UR) -
CURRENTDATA(YES) RELEASE(COMMIT) KEEPDYNAMIC(YES)
END
*/

//*END OF BIND PLAN

SAP DBA Guide for z/OS
GRANT TRACE, MONITOR1, MONITOR2 TO <SIDADM>;
GRANT EXECUTE ON PACKAGE <PACKNAM6>.* TO <SCHEMA>;
GRANT EXECUTE ON PACKAGE <PACKNAM8>.* TO <SCHEMA>;

//*End z/OS only
GRANT EXECUTE ON PACKAGE DSNUTILS.* TO <SCHEMA>;
GRANT EXECUTE ON PACKAGE DSNACC.* TO <SCHEMA>;
GRANT SYSCTRL TO <SCHEMA>;
GRANT ALL ON TABLE DSNACC.TSTEMP_TBL TO <SCHEMA>;
GRANT ALL ON TABLE DSNACC.IXTEMP_TBL TO <SCHEMA>;
GRANT CREATESG TO <SCHEMA>;
GRANT CREATEDBA TO <SCHEMA>;
GRANT DISPLAY TO <SCHEMA>;
GRANT PACKADM ON COLLECTION <SCHEMA> TO <SCHEMA>;
GRANT ALTERIN, CREATEIN, DROPIN ON SCHEMA <SCHEMA> TO <SCHEMA>;
GRANT USE OF ALL BUFFERPOOLS TO <SCHEMA>;
GRANT INDEX ON TABLE SYSIBM.SYSTABLES TO <SCHEMA>;
GRANT INDEX ON TABLE SYSIBM.SYSTABLESPACE TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSCOPY TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSCOLAUTH TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSCOLUMNS TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSFOREIGNKEYS TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSINDEXES TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSINDEXPART TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSTABLEAUTH TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSTABLESPACE TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSFIELDS TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSDATABASE TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSDBAUTH TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.MODESELECT TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.LUMODES TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.LULIST TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.USERNAMES TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.LUNAMES TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.LOCATIONS TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.IPNAMES TO <SCHEMA>;

SAP DBA Guide for z/OS
GRANT SELECT ON TABLE SYSIBM.SYSRESAUTH TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSSTOGROUP TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSVOLUMES TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSPKSYSTEM TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSPLSYSTEM TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSPACKAUTH TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSPACKDEP TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSPACKSTMT TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSPACKAGE TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSPACKLIST TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSDBRM TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSPLAN TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSPLANAUTH TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSPLANDEP TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSTMT TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSCOLDIST TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSCOLDISTSTATS TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSINDEXSTATS TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSTABSTATS TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSDUMMY1 TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSCHECKDEP TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSCHECKS TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSSCHEMAAUTH TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSTABLES TO <SCHEMA>;
GRANT SELECT ON TABLE SYSIBM.SYSTABLESPACE TO <SCHEMA>;
GRANT UPDATE ON TABLE SYSIBM.SYSTABLESPACE TO <SCHEMA>;
GRANT UPDATE ON TABLE SYSIBM.SYSTABLES TO <SCHEMA>;
GRANT UPDATE ON TABLE SYSIBM.SYSTABLESPACE TO <SCHEMA>;
GRANT UPDATE ON TABLE SYSIBM.SYSTABSTATS TO <SCHEMA>;
GRANT UPDATE ON TABLE SYSIBM.SYSINDEXES TO <SCHEMA>;
GRANT UPDATE ON TABLE SYSIBM.SYSCOLUMNS TO <SCHEMA>;
GRANT DELETE ON TABLE SYSIBM.SYSCOLDIST TO <SCHEMA>;
GRANT ALL ON TABLE SYSIBM.SYSPRINT TO <SCHEMA>;
GRANT SELECT, DELETE ON TABLE SYSIBM.TABLESPACESTATS TO <SCHEMA>;
GRANT SELECT, DELETE ON TABLE SYSIBM.INDEXSPACESTATS TO <SCHEMA>;
/*
//SYSTSIN DD *
DSN SYSTEM(<SUBSYSTEM_NAME> )
  RUN PROGRAM(DSNTIAD) PLAN(<DB2_DSNTIAD_PLAN>) -
   LIB('DB2_RUNLIB_LOAD')
END
/*