



PUBLIC

Document Version: 2.4 – 2024-03-01

# Database Administration Guide for SAP on IBM Db2 for z/OS

Valid for Version 12 and Version 13 of the IBM Db2 for z/OS Database for SAP Systems Based on SAP NetWeaver 7.0 and Higher

# Content

- 1 Introduction. . . . . 7**
- 1.1 New Database Features. . . . . 8
- 1.2 Document History. . . . . 10
- 1.3 Naming Conventions. . . . . 12
- 2 Architecture Overview. . . . . 14**
- 2.1 SAP Application Server for ABAP. . . . . 14
- 2.2 Implicit Database Layout for AS ABAP. . . . . 15
  - Implicit Object Creation. . . . . 15
  - Compatibility. . . . . 16
- 2.3 SAP Application Server for Java. . . . . 16
- 2.4 Mapping Between the Java Dictionary and Db2. . . . . 17
  - Db2 Data Types Used by the JAVA Dictionary Only. . . . . 18
- 2.5 Components in an SAP on Db2 for z/OS System. . . . . 18
- 2.6 Db2 Connect. . . . . 20
- 2.7 Other Operations and Considerations. . . . . 20
  - Database Access. . . . . 20
  - Hardware Failure. . . . . 21
  - Db2 Schema. . . . . 21
- 3 Db2 Setup. . . . . 22**
- 3.1 Db2 and z/OS Prerequisites. . . . . 22
- 3.2 Creating the Db2 Subsystem. . . . . 23
- 3.3 Db2 System Parameters. . . . . 24
  - Db2 System Parameters with Required Settings. . . . . 26
  - Db2 System Parameters with Highly Recommended Settings. . . . . 27
  - Db2 System Parameters with Recommended Settings. . . . . 30
  - Db2 Buffer Pool-Related Parameters with Recommended Settings. . . . . 34
  - Db2 Installation Panel DSNTIPD: Recommended Values. . . . . 35
- 3.4 Db2 Continuous Delivery. . . . . 36
- 3.5 Db2 Data Sharing Design Options for SAP. . . . . 37
  - Active/Passive Data Sharing Configurations . . . . . 38
  - Active/Active Data Sharing Configurations . . . . . 39
  - Number of Data Sharing Groups. . . . . 41
  - Number of Sysplexes. . . . . 42
  - Number of Data Sharing Members. . . . . 42
- 3.6 Configuring Db2 Logging. . . . . 43

3.7	Configuring the Default Sign-On Exit Routine. . . . .	43
3.8	Reduction of the Number of Data Sets. . . . .	44
3.9	Separating Db2 System Data Sets from Data Sets Used for SAP Tables and Indexes. . . . .	44
3.10	SAP Exploitation of Db2–Supplied Stored Procedures for Database Administration. . . . .	44
	Stored Procedures Used by SAP. . . . .	45
	Setting Up WLM Application Environments for Stored Procedures. . . . .	45
	Exception Table. . . . .	46
	Testing and Troubleshooting Stored Procedures. . . . .	46
3.11	Checking the Installation of the Db2 Subsystem. . . . .	49
3.12	Ensure Optimal Db2 Settings. . . . .	49
3.13	Setting Up the Db2 Distributed Data Facility (DDF). . . . .	50
3.14	Data Stream Encryption. . . . .	51
<b>4</b>	<b>IBM Data Server Drivers for CLI and JDBC. . . . .</b>	<b>53</b>
4.1	Directory Structure of the Db2 Client Connectivity. . . . .	54
4.2	Obtaining Licenses for the CLI and JDBC Drivers. . . . .	56
4.3	Bind Packages with CLI and JDBC Drivers . . . . .	58
4.4	Controlling Db2 Failover Using db2dsdriver.cfg . . . . .	59
<b>5</b>	<b>Db2 Connectivity. . . . .</b>	<b>61</b>
5.1	User and Password Management. . . . .	61
	Password Service - dbdb2pwd. . . . .	62
	Secure Storage in the File System (SSFS). . . . .	63
	Migration to Secure Storage in the File System from Password Service. . . . .	67
	Migration to TLS Certificate Authentication . . . . .	69
5.2	Setup of CLI Failover with the SAP Failover Configuration Tool. . . . .	70
5.3	CLI Failover - Planned or Unplanned Outage of a Db2 Member. . . . .	74
5.4	Db2 Failover for SAP Java Applications. . . . .	76
<b>6</b>	<b>The DBA Cockpit. . . . .</b>	<b>81</b>
6.1	Introduction to the DBA Cockpit. . . . .	81
	The DBA Cockpit on a Local System and SAP Solution Manager. . . . .	82
	Entry Screen of the DBA Cockpit. . . . .	84
	Screen Layout and Navigation in the DBA Cockpit. . . . .	85
6.2	System Landscape. . . . .	87
	Db2 for z/OS Landscape Tools. . . . .	87
	Adding a Remote SAP System Database Connection (optional). . . . .	88
6.3	Performance. . . . .	91
	Performance Warehouse. . . . .	93
	Storage. . . . .	97
	SQL. . . . .	98
	Db2 Subsystem Summary. . . . .	102

	Db2 Subsystem Activity. . . . .	103
	Thread Activity. . . . .	105
	Global Times. . . . .	107
	Data Sets Statistics. . . . .	108
	Installation Parameters. . . . .	108
6.4	Space. . . . .	110
	Key Performance Indicators. . . . .	110
	Unencrypted Objects. . . . .	112
	Tables and Indexes Monitor. . . . .	113
	Space. . . . .	113
	Tables. . . . .	115
	Indexes. . . . .	116
	Tablespaces. . . . .	117
6.5	Backup and Recovery. . . . .	118
	Backup. . . . .	118
	Recovery. . . . .	121
	Backup Monitor for Image Copies. . . . .	122
6.6	Configuration. . . . .	122
	Failover Configuration. . . . .	122
	Setup of Database Performance Monitoring with Stored Procedure SAPCL. . . . .	125
	Seamless Failover of SAP Application Servers to Other Db2 Members. . . . .	138
	Homogeneous System Copy: Jobhead Correction. . . . .	140
6.7	Jobs. . . . .	141
	Executing a DBA Action. . . . .	141
	JCL Job Submission. . . . .	143
	Job Descriptions. . . . .	151
	Security Considerations for Stored Procedures. . . . .	157
	Restarting Jobs. . . . .	158
	Adjusting the Jobs. . . . .	159
	Reorganization. . . . .	160
	Rebuild Index. . . . .	161
	Update Statistics (RUNSTATS). . . . .	161
	When to Exclude Tablespaces from RUNSTATS, REORG, and COPY. . . . .	163
	Central DBA Planning Calendar. . . . .	164
	DBA Planning Calendar. . . . .	165
	SAP Db2 Conversion Tool. . . . .	167
6.8	Alerts. . . . .	182
	Alerts. . . . .	182
6.9	Diagnostics. . . . .	184
	Storage Management. . . . .	184
	Traces. . . . .	215

	Automatic Start of DB Traces. . . . .	218
	Db2 Connect Diagnostics. . . . .	218
	Data Sharing Topology. . . . .	220
	Web Dynpro EXPLAIN (Extended Version). . . . .	220
	Catalog Browser (7.02 and Higher). . . . .	222
	Dependency Browser. . . . .	223
	Sysplex Monitor. . . . .	225
	Db2 Commands. . . . .	228
	EXPLAIN. . . . .	229
	PTF Status. . . . .	231
	DB2 Message Log. . . . .	237
	z/OS System Log. . . . .	237
<b>7</b>	<b>Backup and Recovery Options. . . . .</b>	<b>239</b>
7.1	Backup. . . . .	240
7.2	Online Backup. . . . .	241
7.3	Backup Procedure Recommendations. . . . .	244
7.4	Combining DB2 BACKUP SYSTEM, DS8000 FlashCopy, DS8000 Metro Mirror, and GDPS. . . . .	247
7.5	Recovery. . . . .	248
7.6	Recovery to the Current State. . . . .	249
7.7	Recovery to a Prior Point in Time. . . . .	250
7.8	Federated Recovery to a Prior Point in Time. . . . .	254
<b>8</b>	<b>Performance Tuning Considerations. . . . .</b>	<b>256</b>
8.1	Setting Optimal SAP Profile Values. . . . .	256
8.2	Monitoring and Tuning of SAP CDS Queries. . . . .	257
8.3	Periodically Recycling SAP Work Processes. . . . .	258
8.4	Customizing the SAP Objects Topology. . . . .	259
8.5	Optimal Access Paths Assurance with RUNSTATS. . . . .	259
	When RUNSTATS Is Due. . . . .	260
	RUNSTATS Options to be Used. . . . .	262
	Access Path Considerations for Volatile Tables. . . . .	264
	Access Path Considerations for Special SAP Tables. . . . .	266
	Access Optimization to VBHDR, VBMOD, and VBDATA. . . . .	267
8.6	Clustering Index. . . . .	271
8.7	Partitioning Key. . . . .	272
	Partitioning SAP Banking Tables. . . . .	274
	Partitioning Financial Tables. . . . .	275
8.8	Locking Considerations. . . . .	275
8.9	Buffer Pool Tuning Considerations. . . . .	280
8.10	Dynamic Statement Caching Considerations. . . . .	282
8.11	Data Sharing Optimization for Different SAP Business Applications. . . . .	282

8.12	Transaction-Level Db2 Accounting and Workload Management. . . . .	282
	Db2 Client Identifiers and WLM Qualifiers. . . . .	283
	Static Properties Passed to Db2. . . . .	284
	Dynamic SAP Properties Passed to Db2. . . . .	285
	Analysis of Accounting Records. . . . .	289
<b>9</b>	<b>Additional Information. . . . .</b>	<b>290</b>
9.1	Transaction Codes. . . . .	290
9.2	Environment Variables and Profile Parameters. . . . .	291
9.3	Update Task Troubleshooting. . . . .	294
9.4	SAP MDM on Db2 for z/OS. . . . .	295
<b>A</b>	<b>Appendix. . . . .</b>	<b>298</b>
A.1	References. . . . .	298
A.2	Disclaimer. . . . .	301

# 1 Introduction

This guide provides detailed information about the administration of IBM Db2 for z/OS in an SAP environment. It's primarily intended for database administrators and SAP system administrators who need to plan, install and maintain an SAP system on IBM Db2 for z/OS. A basic understanding of the fundamental database concepts and an elementary knowledge of SAP system administration are required.

This guide refers to versions 12 and higher of IBM Db2 for z/OS. To avoid double naming, we mostly speak of the "the Db2 database" or just "Db2" database when all database versions are addressed. For more information, see our [Naming Conventions \[page 12\]](#).

## Database Administration Essentials

### Architectural Overview



Find out how the SAP application server for ABAP/Java connects to the database and the required Db2 components to connect to the database

### Db2 Setup



Read about how to set up the DB2 system and parameters settings

### Db2 Connectivity



Find out about variables, profile registry, and parameter settings

### The DBA Cockpit



Read about how you can run the DBA Cockpit as part of your system administration activities in SAP Solution Manager

### Backup and Recovery



Rollforward recovery, backup methods, log file management, and consistency checks

### Performance



Performance monitoring and tuning you should consider

- [SAP Application Server for ABAP \[page 14\]](#)
- [Db2 Setup \[page 22\]](#)
- [Db2 Connectivity \[page 61\]](#)
- [The DBA Cockpit \[page 81\]](#)

- [Backup and Recovery Options \[page 239\]](#)
- [Performance Tuning Considerations \[page 256\]](#)

## 1.1 New Database Features

This section highlights new database features that are beneficial for SAP systems. The use of most of these features is described in more detail in the respective documentation.

The following are the new features:

### Db2 12

#### Function Levels

- Specify a compression algorithm at the table, table space, or partition level (for Huffman Compression)  
Available with Db2 Function level 509. For detailed information about specifying a compression algorithm at the table, table space, or partition level implicit exploited by SAP, see [FTB support for non-unique indexes](#) in the IBM documentation.
- Migration of multi-table table spaces to partition-by-growth universal table spaces(UTS)  
Available with Db2 Function level 508. For detailed information about the migration of multi-table table spaces to partition-by-growth universal table spaces (UTS) explicitly exploited by SAP, see [Migration of multi-table table spaces to partition-by-growth universal table spaces \(UTS\)](#) in the IBM documentation.
- FTB support for non-unique indexes  
Available with Db2 Function level 508. For detailed information about FTB support for non-unique indexes implicit exploited by SAP, see [FTB support for non-unique indexes](#) in the IBM documentation.
- Deletion of old statistics when using profiles  
Available with Db2 Function level 507. For detailed information about the deletion of old statistics when using profiles implicit exploited by SAP, see [Deletion of old statistics when using profiles](#) in the IBM documentation.
- Support for implicitly dropping, explicitly created universal and LOB table spaces  
Available with Db2 Function level 506. For detailed information about the support for implicitly dropping, explicitly created universal, and LOB table spaces implicit exploited by SAP, see [Support for implicitly dropping, explicitly created universal and LOB table spaces](#) in the IBM documentation.
- Improved RUNSTATS performance with automatic page sampling by default  
Available with Db2 Function level 505. For detailed information about the improved RUNSTATS performance with automatic page sampling by default implicit exploited by SAP, see [Improved RUNSTATS performance with automatic page sampling by default](#) in the IBM documentation.
- The explicit casting of numeric values to GRAPHIC or VARGRAPHIC  
Available with Db2 Function level 502. For detailed information about the explicit casting of numeric values to GRAPHIC or VARGRAPHIC explicit exploited by SAP, see [Explicit casting of numeric values to GRAPHIC or VARGRAPHIC](#) in the IBM documentation.
- Key label management for z/OS DFSMS data set encryption



Available with Db2 Function level 502. For detailed information about the key label management for z/OS DFSMS data set encryption explicit exploited by SAP, and implemented into CCMS, see [Key label management for z/OS DFSMS data set encryption](#) in the IBM documentation.

- The LISTAGG built-in function  
Available with Db2 Function level 501. For detailed information about the LISTAGG built-in function explicit exploited by SAP, see [The LISTAGG built-in function](#) in the IBM documentation.

## New Function APARs

- Enhanced IRLM deadlock processing  
Available with APAR PH08708 and APAR PH08431. For detailed information about enhanced IRLM deadlock processing implicit exploited by SAP, see [APAR PH08708](#) and [APAR PH08431](#) in the IBM documentation.
- Asynchronous XI (cross-invalidation)  
Available with APAR PH05193. For detailed information about the asynchronous cross-invalidation (XI) feature implicit exploited by SAP, see [APAR PH05193](#) in the IBM documentation.
- Sort Lists  
Available with APAR PH31684 and Z15. For detailed information about the SORT LISTS in Db2 RDS sort operations implicit exploited by SAP, see [APAR PH31684](#) in the IBM documentation.
- zHyperLink read support  
Available with APAR PI82575. For detailed information about the zHyperLink database synchronous read I/O exploitation option for SAP workload, see [APAR PI82575](#) in the IBM documentation.
- zHyperLink log write support  
Available with APAR PH05030. For detailed information about the zHyperLink log write support option for SAP workload, see [APAR PH05030](#) in the IBM documentation.
- REORG option to minimize the duration of the last log iteration <https://www.ibm.com/support/pages/apar/PH33864>
- Support for UNI\_90 locale for the LOWER, TRANSLATE, and UPPER built-in functions <https://www.ibm.com/support/pages/apar/PH47187>

## Db2 13

### Function Levels

- Utility object-level history [https://www.ibm.com/docs/en/db2-for-zos/13?topic=levels-function-level-504-apar-ph54919-october-2023#db2z\\_fl\\_v13r1m504\\_\\_e18665](https://www.ibm.com/docs/en/db2-for-zos/13?topic=levels-function-level-504-apar-ph54919-october-2023#db2z_fl_v13r1m504__e18665)

### New Function APARs

- Generating tailored JCL jobs in batch mode for migrating to Db2 13 or activating function levels <https://www.ibm.com/support/pages/apar/PH52482>
- Improved index traverse count information in DISPLAY STATS output <https://www.ibm.com/support/pages/apar/PH51565>
- Removed stacking limitations for PBG-to-PBR conversion <https://www.ibm.com/support/pages/apar/PH51359>
- Product identifier (PRDID) values for specific DRDA levels <https://www.ibm.com/support/pages/apar/PH48184>

- Improved PLAN\_TABLE.PARENT\_PLANNO column data for views and table expressions <https://www.ibm.com/support/pages/apar/PH49972> ➔
- List prefetch for MERGE statements <https://www.ibm.com/support/pages/apar/PH47581> ➔

## Z15 Hardware

IBM Z Instant Recovery

For detailed information about IBM Z instant recovery option for SAP workload, see [APAR PH05030](#) ➔ in the IBM documentation.

## More Information

- SAP Note [81737](#) ➔: Db2-z/OS: APAR List
- SAP Note [3152911](#) ➔: Release of Db2 13 for SAP Components
- SAP Note [2302997](#) ➔: Release of Db2 12 for SAP Components

## 1.2 Document History

### ⚠ Caution

Make sure you have the latest version of this document that you can find at <https://help.sap.com/docs/db2-for-zos/database-administration> on SAP Help Portal.

The following table provides an overview of the essential document changes:

Document History

Version	Date	Description
2.4	2024-03-01	<p>The following topics are updated or newly added:</p> <ul style="list-style-type: none"> <li>• <a href="#">Explain Dependency [page 223]</a></li> <li>• <a href="#">Explore Dependencies [page 224]</a></li> <li>• <a href="#">New Database Features [page 8]</a></li> <li>• <a href="#">Static Statement Cache [page 100]</a></li> </ul>

Version	Date	Description
2.3	2023-08-01	<p>The following topics are updated:</p> <ul style="list-style-type: none"> <li>• <a href="#">Adjusting the Jobs [page 159]</a></li> <li>• <a href="#">Adding Volume Space [page 213]</a></li> <li>• <a href="#">Checking the Installation of the Db2 Subsystem [page 49]</a></li> <li>• <a href="#">Db2 and z/OS Prerequisites [page 22]</a></li> <li>• <a href="#">Db2 Continuous Delivery [page 36]</a></li> <li>• <a href="#">Db2 System Parameters with Recommended Settings [page 30]</a></li> <li>• <a href="#">Db2 System Parameters with Required Settings [page 26]</a></li> <li>• <a href="#">Exception Table [page 46]</a></li> <li>• <a href="#">Migration to Secure Storage in the File System from Password Service [page 67]</a></li> <li>• <a href="#">New Database Features [page 8]</a></li> <li>• <a href="#">Password Rules [page 64]</a></li> <li>• <a href="#">Periodically Recycling SAP Work Processes [page 258]</a></li> <li>• <a href="#">References [page 298]</a></li> <li>• <a href="#">RUNSTATS Options to be Used [page 262]</a></li> <li>• <a href="#">Setup of CLI Failover with the SAP Failover Configuration Tool [page 70]</a></li> <li>• <a href="#">Step 4: Refresh the WLM Application Environment [page 133]</a></li> <li>• <a href="#">Thread Activity [page 105]</a></li> <li>• <a href="#">When RUNSTATS Is Due [page 260]</a></li> </ul>
2.2	2023-03-01	<p>The following topics are updated or newly added:</p> <ul style="list-style-type: none"> <li>• <a href="#">IBM Data Server Drivers for CLI and JDBC [page 53]</a></li> <li>• <a href="#">Directory Structure of the Db2 Client Connectivity [page 54]</a></li> <li>• <a href="#">Obtaining Licenses for the CLI and JDBC Drivers [page 56]</a></li> <li>• <a href="#">Bind Packages with CLI and JDBC Drivers [page 58]</a></li> <li>• <a href="#">Controlling Db2 Failover Using db2dsdriver.cfg [page 59]</a></li> <li>• <a href="#">User and Password Management [page 61]</a></li> <li>• <a href="#">Password Service - dbdb2pwd [page 62]</a></li> <li>• <a href="#">Secure Storage in the File System (SSFS) [page 63]</a></li> <li>• <a href="#">Password Rules [page 64]</a></li> <li>• <a href="#">Migration to Secure Storage in the File System from Password Service [page 67]</a></li> <li>• <a href="#">Migration to TLS Certificate Authentication [page 69]</a></li> <li>• <a href="#">Transaction Codes [page 290]</a></li> <li>• <a href="#">Environment Variables and Profile Parameters [page 291]</a></li> </ul> <p>Correction of links</p>
2.1	2022-12-15	<p>General update with content and links</p> <p>Section <a href="#">Step 1: Adding a Remote Database Connection [page 88]</a> was added.</p>
2.0	2022-06-24	<p>Update due to new Db2 version 13</p> <p>Correction of links</p>

Version	Date	Description
1.9	2021-04-01	Section <a href="#">Unencrypted Objects [page 112]</a> was added.
1.8	2020-12-31	General update with content, links and language corrections Section <a href="#">Failover Configuration [page 122]</a> was added.
1.7	2020-07-01	Update Version
1.6	2020-01-31	Section <a href="#">Dependency Browser [page 223]</a> was added.
1.5	2019-07-01	Update Version
1.4	2019-01-01	Update Version
1.3	2018-01-01	Update Version
1.2	2017-09-30	Update Version Change of database name by IBM to IBM Db2 for z/OS, see <a href="#">Naming Conventions [page 12]</a>
1.1	2017-03-31	Update Version
1.0	2016-10-21	Initial Version The content of this guide refers to the database version 11 and 12 of IBM Db2 for z/OS and is valid for SAP NetWeaver 7.0 and higher and all SAP systems based on these releases.

## 1.3 Naming Conventions

### SAP Terminology

- SAP NetWeaver system is referred to as *SAP system*. Additionally, the term SAP system also refers to any application system that is based on SAP NetWeaver, for example, any product of SAP Business Suite.
- SAP NetWeaver Application Server ABAP is referred to as *AS ABAP*.
- SAP NetWeaver Application Server Java is referred to as *AS Java*.

## Database Versions of IBM Db2

In this document, the database versions are referred to as follows:

Database Name	Abbreviation
IBM® Db2® 13 for z/OS®	<i>Db2 13</i>
IBM® Db2® 12 for z/OS®	<i>Db2 12</i>
IBM® Db2® 11 for z/OS®	<i>Db2 11</i>

## Renaming: IBM DB2 for z/OS is now IBM Db2 for z/OS

Some of you might wonder why we spell Db2 with a lowercase 'b' now. IBM has changed the database name from IBM DB2 for z/OS to IBM Db2 for z/OS. You will still find the old product name in older SAP publications, but we will gradually use the new term in future documentation. You will even see the old product name in older SAP publications, but we will use the new term progressively in future documentation. IBM Clients and Drivers.

## IBM Clients and Drivers

IBM Data Server Driver for ODBC and CLI is referred to as a *CLI driver*.

IBM Data Server Driver for JDBC and SQLJ is referred to as a *JDBC driver*.

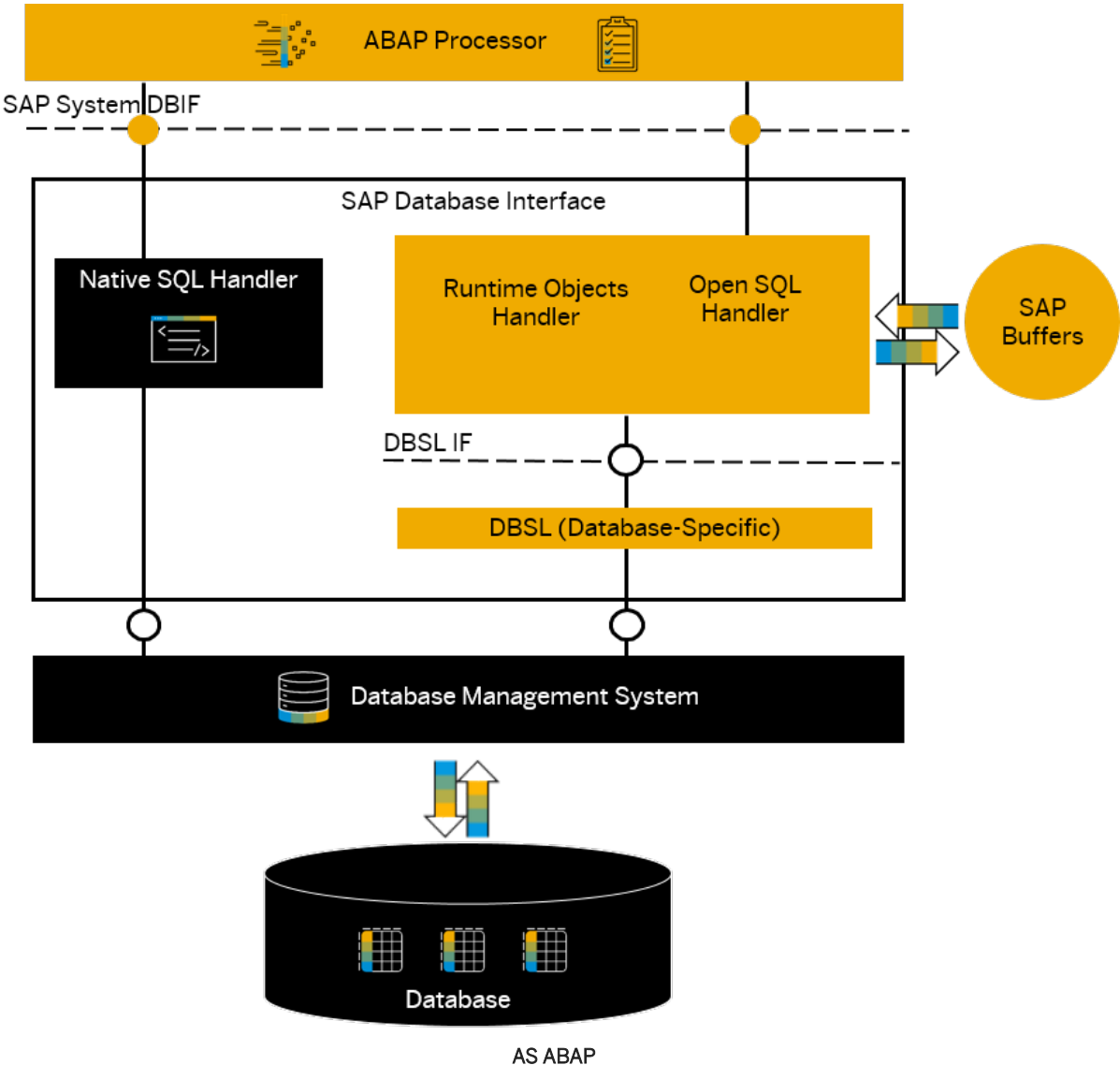
## Other Terminology

The term *Windows* refers to the Microsoft Windows operating system.

# 2 Architecture Overview

## 2.1 SAP Application Server for ABAP

The following figure provides an overview of how the SAP application server for ABAP (AS ABAP) connects to the database.



The ABAP language offers the following options to communicate with the database:

- OpenSQL for ABAP (SAP's database-independent SQL dialect used by most standard SAP applications)
- Native SQL (database-dependent)

The ABAP processor uses a database interface to connect to the database. The database interface provides a database platform abstraction layer and translates all Open SQL statements from the ABAP processor into native database-specific SQL statements.

The database interface also performs the database platform-specific type mapping between ABAP data types and database data types. Each database platform provides a platform-specific database interface library, also called *Database Shared Library* (DBSL). The DBSL is part of the SAP kernel and developed in C.

The DBSL for IBM Db2 for z/OS (dbdb21ib.\*) uses Db2's *Call Level Interface* (CLI) to communicate with the *Database Management System* (DBMS). To use CLI, the DBSL dynamically loads the Db2 client libraries.

## More Information

[Directory Structure of the Db2 Client Connectivity \[page 54\]](#)

[Introduction to Db2 Call Level Interface and ODBC](#)  in the IBM Knowledge Center.

## 2.2 Implicit Database Layout for AS ABAP

### 2.2.1 Implicit Object Creation

#### Use

The SAP installation procedure and SAP systems (ABAP and Java) fully employ Db2's implicit object creation feature. This means that the SAP applications do not create databases, tablespaces, index spaces, auxiliary tables, and indexes, but implicitly by the Db2 database. The storage attributes are governed by the Db2 system parameter settings (ZPARMs).

#### Example

When creating a table, Db2 makes or reuses databases with the name `DSN<nr>` (`<nr>=00001, 00002, . . .`) in stogroup `SYSDEFLT` (creator `SYSIBM`) and automatically generates tablespaces, auxiliary tables, and their indexes.

#### Naming Conventions

Db2 uses the `VCAT` name associated with `SYSDEFLT` as a high-level qualifier for the Db2 data sets for all objects of this storage group. The `VCAT` also controls the ICF catalog that is used for the associated data sets. Therefore, we recommend that you assign a separate `VCAT` name to `SYSDEFLT`. Also, it would help if you assigned non-SAP objects to another storage group.

Db2 does not use `SYSDEFLT` to store its objects like Db2 Catalog tables.

## More Information

The `ZPARM` settings determine the storage attributes used by Db2 when creating implicit objects. For more information about these parameters, see [Db2 System Parameters \[page 24\]](#).

### 2.2.2 Compatibility

The database layout with implicitly created databases and tablespaces differs considerably from the layout implemented with earlier SAP Releases and Db2 versions.

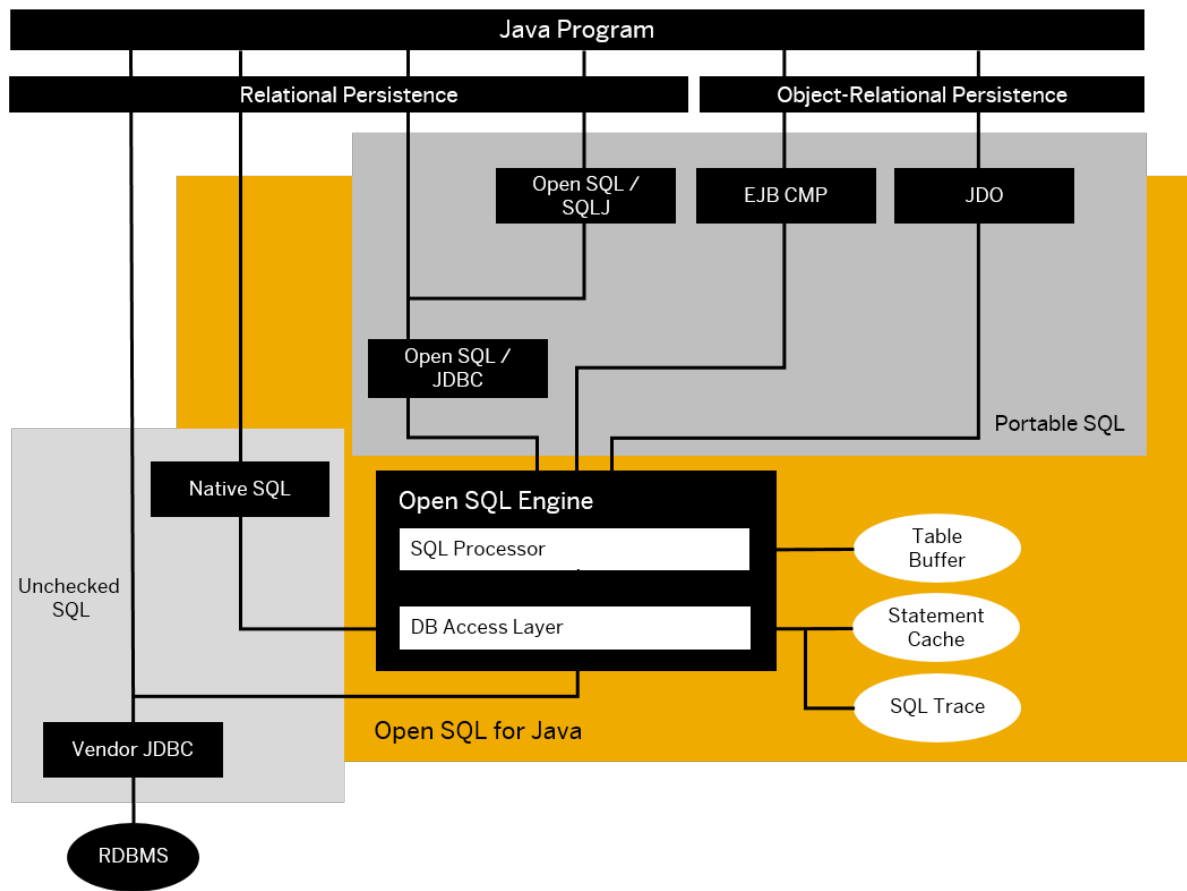
If your system is upgraded from an earlier SAP or Db2 release, the new layout only applies to the following:

- New tables
- Existing tables that need to be recreated due to structural changes

## 2.3 SAP Application Server for Java

The following figure provides an overview of how the SAP application server for Java (AS Java) connects to the database.





AS Java

Java programs that run inside the SAP application server for Java can use various standardized APIs to access the database, such as JDO, SQLJ, or JPA.

The database interface provides Java applications with the following options to communicate with the database:

- OpenSQL for Java (SAP's database-independent SQL dialect)
- Native SQL (database-dependent)

The SAP application server for Java uses various services that assist in the communication with the DBMS (for example, the dbpool service for database connection pooling).

All communication with the DBMS is done using the Db2 JDBC driver - a pure Java Type 4 JDBC driver that uses TCP/IP as its network protocol.

## 2.4 Mapping Between the Java Dictionary and Db2

SAP Java 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

With Db2 for z/OS, you can mix Unicode and non-Unicode tables within one subsystem. Therefore, you can create tables for SAP NetWeaver Java persistence in the subsystem used to store the non-Unicode (or Unicode) objects of the SAP NetWeaver ABAP part of the SAP system.

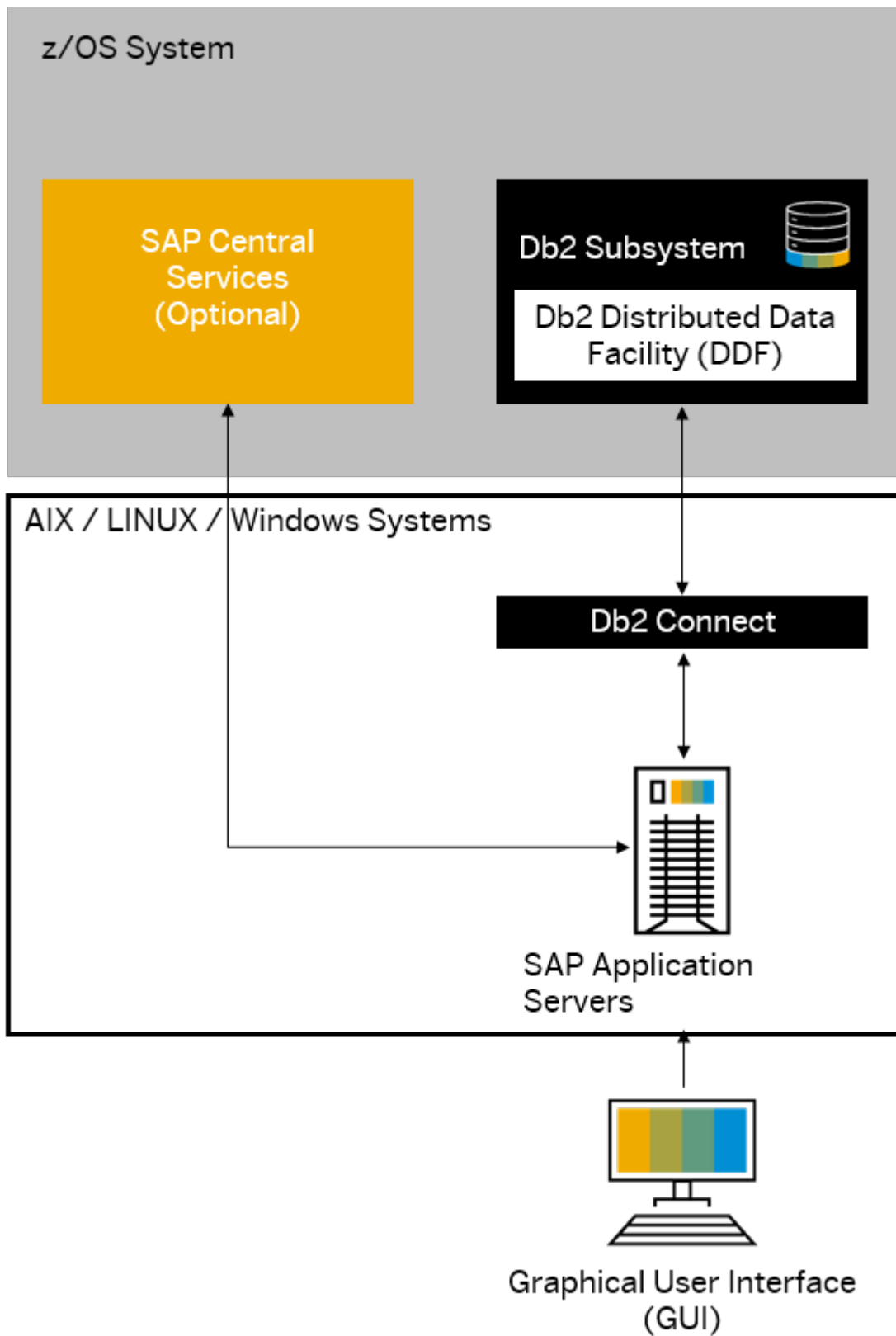
## 2.4.1 Db2 Data Types Used by the JAVA Dictionary Only

The following type used by the SAP Java dictionary is **not** used in the ABAP dictionary.

- Java types `BINARY(N)` and `VARBINARY(N)` are mapped to database types `BINARY(N)` and `VARBINARY(N)`.

## 2.5 Components in an SAP on Db2 for z/OS System

The following sections describe the components used on the application and database servers and their connecting components.



Components in an SAP on Db2 for z/OS System

## 2.6 Db2 Connect

### Use

Db2 Connect is the standard remote SQL interface from IBM that implements the Distributed Relational Database Architecture (DRDA) protocol. It consists of client components and a server component, which establish a remote SQL connection between SAP application servers and the Db2 for z/OS database server.

Db2 Connect includes two standalone drivers, the *Db2 JDBC* driver and the *Db2 CLI* driver. The SAP application server uses the Db2 JDBC driver for the Java stack and the Db2 CLI driver for the ABAP stack to establish connections to Db2 for z/OS.

### More Information

[SAP NetWeaver installation guides](#) on SAP Help Portal

## 2.7 Other Operations and Considerations

### 2.7.1 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).

#### Caution

If you intend to read data with an isolation level that causes database 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.

## 2.7.2 Hardware Failure

### Use

It would help if you implemented plans to reduce the effects of unplanned downtime due to hardware, software, or communication failure. 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

#### Caution

In the SAP system environment, planning for these failures should be similar to conventional failure recovery planning. Considering the following: Due to the highly integrated nature of the SAP data, you should perform any database recovery triggered by any failure with care to ensure that the recovered data is logically consistent. In a point-in-time recovery, you will lose any updated data after the target point-in-time.

## 2.7.3 Db2 Schema

Throughout this guide, the variable `<SCHEMA>` is used for the Db2 schema of the Db2 objects. This variable belongs to the SAP system. In older releases, you used the hard-coded name `SAPR3`.

## 3 Db2 Setup

Setting up the Db2 subsystem is an essential step in the preparation of an SAP system installation. A large number of parameters 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. The following sections only cover those parameters with special relevance to the SAP system.

### 3.1 Db2 and z/OS Prerequisites

#### Note

The following sections describe the required, highly recommended, and recommended settings for Db2 system parameters. The recommended values are listed in SAP Note [2239553](#) (for Db2 12) and SAP Note [3093470](#) (for Db2 13).

Make sure you fulfill the following requirements before you begin installing the Db2 subsystem:

- Db2 Utilities Suite for z/OS
- z/OS Unicode Conversion Services (UCS) and appropriate conversion definitions need to be set up for your environment. See the IBM Db2 documentation and Information APARs I113048, I113049.
- RRS needs to be installed and set up.  
For more information, see the *Planning Guide for SAP on IBM Db2 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 more information, see the following IBM documentation:
  - *Administration Guide*
  - *Application Programming and SQL Guide*
  - *Installation and Migration Guide*
  - *z/OS MVS Setting Up a Sysplex*
  - *z/OS MVS Programming: Resource Recovery*
- If you run Db2 in data sharing-mode, you need to set up a Coupling Facility.
- The Db2 supplied [stored procedures \[page 44\]](#) must be enabled.
- Make sure that there is no real memory paging to AUX storage in your LPARs. To prevent that, DFSORT steals real storage from other address spaces, set DFSORT parameter EXPOLD to 0. For fine-tuning, consider setting EXPMAP, HIPRSIZE, and MOSIZE to adequate values. z/OS 2.1 introduces the TUNE parameter that can also help. For more information, see the IBM Documentation *DFSORT Tuning Guide*.  
DFSORT Tuning Guide
- 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 more information, see SAP Note [81737](#).

### Note

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 \[page 232\]](#).

## Disk Layout for Db2

We recommend for your disk layout the following:

- Define two DFSMSHsm copy pools if you plan to use the Db2 utility BACKUP SYSTEM (see *Backup and Recovery Options* section) as described in the IBM documentation *Db2 13 for z/OS Utility Guide and Reference*.
- The amount of data that an individual volume can contain depends on the volume type; for example, a 3390 Model 27 volume can hold approximately 27 GB. Maximum size of a partition is 1TB with extended address volumes (EAV). See also the IBM documentation *Limits in Db2 for z/OS*. You should ensure that the data class associated with your Db2 objects provides can accommodate 64 GB data sets. A good starting point for Dynamic Volume Count is often 4. You should not set this parameter to a very large value like 30 or 40 since this can lead to shortages in the z/OS task input/output table (TIOT) table.
- Using RAID 5 often works best as it ensures good performance; (Disaster recovery can be addressed using Metro Mirror and Hyperswap)
- Use Hyperpav
- Use the zHPF (High-Performance FICON) protocol.

## 3.2 Creating the Db2 Subsystem

### Use

Since SAP systems are relatively complex and many objects are associated with them, the best choice is to keep non-SAP applications out of Db2 subsystems dedicated to SAP systems. This gives you the following benefits:

- You can set db2 parameters to values that ensure efficient SAP system operations
- You can monitor resource consumption more easily
- Sizing estimates are simplified
- Security handling is facilitated
- You can implement backup and recovery procedures more easily

## More Information

For information on considerations for possible system layouts, see section *Db2 subsystem and z/OS system in Planning Guide for SAP on IBM Db2 for z/OS*.

### 3.3 Db2 System Parameters



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 the proper functioning of SAP systems and must not be changed.
- **Highly recommended settings**  
Although they do not influence SAP system functions, these settings are essential for ensuring optimal performance and should not be changed. For example, the setting `NRGTHRS=10` provides the right 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.
- **Workfile Recommendations**  
Make sure to create the following types of work tablespaces:
  - 4KB Sort Work Tablespaces defined as non-UTS with zero SECQTY
  - 32KB Sort Work Tablespaces defined as non-UTS with zero SECQTY
  - 4KB DGTT Work Tablespaces defined as UTS PBG
  - 32KB DGTT Work Tablespaces defined as UTS PBG

Note that a single set of recommendations applies to all SAP systems, including SAP BW systems, facilitating the setup.

Once you have installed the SAP system, the settings can be checked using transaction [▶ DBACOCKPIT](#)  
[▶ Performance](#) [▶ Installation Parameters](#) [▶](#).

#### Note

The following sections describe the required, highly recommended, and recommended settings for Db2 system parameters. The recommended values are listed in SAP Note [1863848](#)  (for Db2 11) and SAP Note [2239553](#)  (for Db2 12).

## Implicit Object Creation

SAP exploits the Db2 feature to create objects for new tables for all SAP releases implicitly. This means that SAP only issues CREATE TABLE and CREATE INDEX statements, and all underlying objects such as database, tablespace, or LOB tablespace are automatically created by Db2. However, existing tables can remain in their



tablespaces. Implicitly created objects use the Db2 default storage group `SYSDEFLT`. Db2 uses the `VCAT` name associated with this default storage group as a high-level qualifier for the Db2 data sets for this storage group's objects. The `VCAT` also controls the `ICF` catalog that is used for the associated data sets. Therefore, we recommend that you assign a separate `VCAT` name to `SYSDEFLT`. Also, it would help if you assigned non-SAP objects to another storage group.

Suppose you want to change the high-level qualifier of Db2 system data sets, such as data sets used by the Db2 system catalog tables. In that case, you can follow the approach described in IBM documentation, *Changing the qualifier for system data sets*.

## Extended RBA and LRSN in 10-byte Format

We recommend converting Db2 objects that still use 6-byte RBA or LRSN format to extend 10-byte RBA and LRSN format. The Db2 `REORG` utility accomplishes the conversion. You can take advantage of the regular `REORG` executions that you are executing, such as part of SAP DBA Cockpit, performing the conversion for these objects. You can control the conversion via the Db2 system parameter `UTILITY_OBJECT_CONVERSION` or `REORG` parameter `RBALRSN_CONVERSION`.

## Accounting and SMF Considerations

Accounting and SMF Considerations Db2 allows you to reduce the volume of accounting records that Db2 generates in SMF by accumulating accounting records with the same work identifiers as an end-user ID. The Db2 system parameters `ACCUMACC` and `ACCUMUID` control this. Collecting accounting records can complicate the transaction-level analysis of Db2 performance, in any case. The reason is that you may accumulate accounting data of independent SAP transactions executed in different SAP work processes in a single accounting record written to SMF. Therefore, the general SAP recommendation is not to enable accounting records aggregation and set `ACCUMACC` to `NO`.

To reduce SMF data's total volume, you should compress this by setting the Db2 system parameter `SMFCOMP` to `YES`. If the goal is to minimize the amount of SMF data and if the detailed investigation of single Db2 transaction execution performance is not essential, you can also consider exploiting accounting aggregation by setting `ACCUMACC` to 5 and `ACCUMUID` to 11.

## More Information

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, see the IBM documentation *Installation and Migration Guide*.

We recommended keeping these parameters' default values for Db2 system parameters that are not listed here.

### Note

We recommend using the default values for any Db2 system parameter not explicitly mentioned in the following sections.

## 3.3.1 Db2 System Parameters with Required Settings

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

### Note

The recommended values are listed in SAP Note [2239553](#) (for Db2 12) and SAP Note [3093470](#) (for Db2 13).

Db2 System Parameters with Required Settings

Parameter	Explanation
ASCCSID	This parameter always needs to be <b>819</b>
CMTSTAT	<b>INACTIVE</b>  This setting is essential to ensure granular monitoring of SAP workload in Db2 while at the same time, Db2 threads remain active due to the KEEP_DYNAMIC(YES) bind option.
APPLCOMPAT	If you are running Db2 11, set this parameter to <b>V11R1</b> .  If you are running Db2 12, set it to <b>V12R1M500</b> .  If you are running Db2 13, set it to <b>V12R1M501</b> .
DECIMAL	Always use a period (.) and never a comma (,)
IDTHTOIN	Set IDTHTOIN (idle thread timeout) to 0 to disable timeout processing.
IDBACK	<b>50</b>
MAXDBAT	MAXDBAT should be at least 1000. 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 connected to this Db2 subsystem via DRDA.

### Note

The work processes that serve SAP BW open a secondary connection by default.

Parameter	Explanation
SCCSID	<p>Can be set to any value where Db2 supports the EBCDIC invariant character set (for example, 37 is such a CCSID) and that supports bijective translation between SCCSID and ASCCSID for the following characters:</p> <ul style="list-style-type: none"> <li>0123456789</li> <li>ABCDEFGHIJKLMNOPQRSTUVWXYZ</li> <li>_#~^</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p><b>Note</b></p> <p>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. It would help if you did not change SCCSID once the SAP database is loaded. If you need to change SCCSID, contact IBM Db2 support for assistance.</p> </div>
LIKE_BLANK_INSIGNIFICANT	This parameter specifies whether blanks are significant when applying the LIKE predicate to a string.

### 3.3.2 Db2 System Parameters with Highly Recommended Settings

The settings listed in the following table are highly recommended to ensure that SAP operates most efficiently. It would help if you did not change them.

The recommended values are listed in SAP Note [1863848](#) (for Db2 11) and SAP Note [2239553](#) (for Db2 12).

Db2 System Parameters with Highly Recommended Settings

Parameter	Explanation
NUMLKUS	<p>It 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 <code>sqlcode-904</code>. The maximum value for NUMLKUS is 100 million. 2097152 is recommended as an initial, first-cut value. Setting a lower value for NUMLKUS helps you 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 information, see <a href="#">Locking Considerations in Performance Tuning Considerations</a>.</p>
NUMLKTS	<p>Sets a limit on the number of locks held concurrently by a single Db2 thread on a single tablespace.</p>

Parameter	Explanation
IRLMRWT	SAP applications are written to minimize database lock contentions. However, even when such a contention happens, SAP prefers a long wait rather than a quick conflict resolution resulting in a potentially lengthy rollback.
DEADLOK	<p>Belongs to the IRLM startup procedure. Lower values increase the likelihood of IRLM latch contentions.</p> <p>This parameter specifies two values: WAIT TIME FOR LOCAL DEADLOCK and LOCAL CYCLES PER GLOBAL CYCLE.</p> <p>The LOCKING section of the Db2 accounting report provides the average, and a maximum number of MAX PG/ROW LOCKS HELD. If these values are in the range of thousands or less, which indicates that there are relatively few entries in the IRLM lock table, you can also set this parameter 1 . 1.</p>
EVALUNC	Setting this parameter to YES reduces lock contention caused by the locking of non-qualifying rows.
MONSIZE	This parameter sets the OP buffer size (in bytes). The statistics trace uses the buffer started for the SAPCL alert router. By tuning this parameter, you ensure that you can hold all exception events in the OP buffer until the SAPCL alert router reads the buffer the next time. If the specified buffer size is not large enough, an exception event loss occurs when the buffer fills before the SAPCL alert router can obtain the exceptions.
NPGTHRSH	<p>This parameter is inputted into 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.</p> <p>As of Db2 12, a value of -1 in NPAGES is considered to be less than 10.</p>
LOBVALA	<p>The size of the user storage for LOB values (in KB). The recommended value is 1 GB.</p> <p>This parameter was removed in Db2 12.</p>
LOBVALS	<p>The size of the system storage for LOB values (in MB). The recommended value is 50 GB.</p> <p>We removed this parameter in Db2 12.</p>
STARJOIN	Enables a join type called star join for accessing data in a star schema.
SJTABLES	When star join is generally enabled using the STARJOIN parameter, it will be considered only if the number of tables is greater than or equal to the value specified in SJTABLES.

Parameter	Explanation
MAX_OPT_CPU	Controls statement preparation resources in terms of CPU consumption (specified in seconds).  This is a hidden keyword ZPARM.
MXDTCACH	Size of storage pool used for data caching during query execution-only, for example, for star join access paths.
IMPDSDEF	Do not define datasets at creation time for implicitly created objects.
IMPTSCMP	Use data compression for implicitly defined tablespaces.
IMPDSIZE	Specifies the maximum data set size (DSSIZE) in gigabytes that Db2 should use for creating an implicit base table space.
IMPTKMOD	Specifies whether Db2 is to track modifications to the pages of implicitly created tablespaces. This not beneficial if the Db2 BACKUP SYSTEM utility or FlashCopy image copies are used to create backups.
TBSBPOOL	Default buffer pool for 4K tablespaces
IDXBPOOL	Default buffer pool for indexes
TBSBLOB	Default buffer pool for implicitly created LOB tablespaces
TSBP8K	Default buffer pool for 8K tablespaces
TSBP16K	Default buffer pool for 16K tablespaces
TSBP32K	Default buffer pool for 32K tablespaces
TSQTY	Default primary quantity for tablespaces.  <div style="border: 1px solid #0070C0; padding: 5px; background-color: #E6F2FF;"> <p><b>Note</b></p> <p>A value of 288KB ensures that the tablespace's compression dictionary also fits into the primary quantity..</p> </div>
IXQTY	Default primary quantity for indexes.
DSNSEQ_IMPLICITDB	For new Db2 subsystems, this catalog sequence defaults to the value 10000, useful in SAP environments. This sequence limits the number of implicit databases that Db2 uses for implicitly created objects. You created the Db2 subsystem in an earlier Db2 release. This value might still be higher. If this is the case, issue the SQL statement <code>ALTER SEQUENCE</code> to set the value of <code>DSNSEQ_IMPLICITDB</code> to 10000.
LOB_INLINE_LENGTH	Specify 2000 as the default inline length of LOBs. SAP Java relies on this default value. For SAP ABAP applications, the inline size of LOBs is explicitly specified as part of DDL.



Parameter	Explanation
PAGESET_PAGENUM	New with Db2 12. It controls whether UTS PBR tablespaces are created with relative page numbering.
PCTFREE_UPD	This parameter controls whether the SQL INSERT algorithm reserves free spaces on pages for future SQL UPDATE statements that extend the data records' length.
ALLOW_DEP_VIEW_SQLTUDF	This parameter is essential for the Zero Downtime Option of SUM (SAP Software Update Manager).
MAX_CONCURRENT_PKG_OPS	This parameter impacts the number of concurrent bind requests.

We recommend accepting the values that the installation CLIST proposes (based on the number of threads) on panel DSNTIPC for the Db2 system parameters `EDMPOOL` and `EDMDBDC`. In any case, `EDMPOOL` must not be smaller than 64000 KB, and `EDMDBDC` must not be shorter than 150000 KB.


### 3.3.3 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 resulting from performance monitoring and tuning.

#### Note

The recommended values are listed in SAP Note [2239553](#)  (for Db2 12) and SAP Note [3093470](#)  (for Db2 13).

Db2 System Parameters with Recommended Settings

Parameter	Explanation
ACCUMUID	Client identifiers that control the accumulation of accounting records.
	<div data-bbox="608 1518 713 1554" data-label="Section-Header"> <h4> Note</h4> </div> <div data-bbox="608 1572 1334 1608" data-label="Text"> <p>This recommendation only applies if you accumulate accounting records.</p> </div>
CHKFREQ	A value of 2 means that Db2 takes a checkpoint every 2 minutes.
CHECK_FASTREPLICATION	Always use FlashCopy for the CHECK utility.
DDF	Automatically starts the DDF address space.
EDMSTMTC	Specifies the size (in KB) of the EDM statement cache.
EN_PJSJ	Enables dynamic index ANDing, which is also called pair-wise join processing, when star join processing is enabled on Db2.

Parameter	Explanation
FLASHCOPY_REBUILD_INDEX	<p>Specifies the default behavior for the REBUILD INDEX utility.</p> <p>If it is set to <b>YES</b>, the utility will use data set FlashCopy to create the inline image copy. Set the parameter only to use if you have FlashCopy.</p>
FLASHCOPY_REORG_INDEX	<p>Specifies the default behavior for the REORG INDEX utility.</p> <p>If it is set to <b>YES</b>, the utility will use data set FlashCopy to create the inline image copy. Set the parameter only to use if you have FlashCopy.</p>
FLASHCOPY_LOAD	<p>Specifies the default behavior for the LOAD utility.</p> <p>If it is set to <b>YES</b>, the utility will use data set FlashCopy to create the inline image copy. Set the parameter only to use if you have FlashCopy.</p>
FLASHCOPY_PPRC	<p>This setting ensures that data set-level FlashCopy is only invoked with mirrored volumes using Metro Mirror (a.k. PPRC) when it is ensured that the volumes do not go into DUPLEX PENDING state, which may be disruptive.</p>
DSMAX	<p>Determines the maximum number of data sets that are to be allowed open at one time.</p>
LBACKOUT	<p>Disables postponing backouts for long running units of recovery at restart.</p> <p>During the SAP application server startup, the application server accesses a large number of tables. Therefore, the overall Db2 subsystem should be consistent once the application servers have been started. Setting LBACKOUT to NO ensures this because it causes Db2 to process the backward log during Db2 restart completely.</p> <p>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 makes those Db2 objects available earlier that do not need to be recovered.</p>
LRDRTHLD	<p>Threshold that controls the identification of long-running readers.</p>
MAXCONQW	
MAXKEEPD	<p>Depending on the number of concurrent Db2 threads, you should adjust this parameter to reduce CPU utilization. For example, if a single Db2 subsystem serves 1000 SAP work processes, MAXKEEPD increase value in the range of 50000.</p> <p>Monitor the Db2 local statement cache hit ratio near to determine the best value of MAXKEEPD. If it is below 80%, increment MAXKEEPD in steps of 5000 until the hit ratio is satisfactory.</p>
MAXRBLK	<p>The maximum size for RID List processing should be 100 MB. This value can be increased if there is enough real storage available. This size is in kilobytes.</p>

Parameter	Explanation
MLMT	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.
OBJECT_CREATE_FORMAT	This parameter controls whether newly created tablespaces use 10-byte RBA/LRSN format.
PARAMDEG	Sets the limit to the maximum degree of parallelism with which you can execute a query. SAP systems can use query parallelism in a very controlled manner (for some components and some selected statements only) and can explicitly turn it on (on a statement level). We recommend limiting the maximum degree of parallelism to the number of available CPUs.
PCLOSET	<p>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 a read-only state.</p> <p>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 a 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.</p>
PCLOSEN	<p>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 a read-only state.</p> <p>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 a read-only state. Having Db2 switch an infrequently updated page set from read-write to a read-only state can result in performance benefits for recovery, logging, and data sharing processing.</p>
REC_FASTREPLICATION	Use FlashCopy for Db2 RECOVER utility whenever possible. If the disk subsystem does not support cascaded FlashCopy, BACKUP SYSTEM and RECOVER can not exploit FlashCopy at the same time.
RETLWAIT	Applies to data sharing only. We recommend that you wait for retained locks rather than receive an unavailable resource message immediately.
SMFACCT	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 activated to facilitate efficient tracking.
SMFCOMP	We recommend compressing SMF records rather than accumulating accounting records to allow fine-grained performance analyses at the Db2 transaction-level.



Parameter	Explanation
SRTPOOL	Amount of storage (in KB) needed for the sort pool. This value can be increased if there is enough real storage available.
STATROLL	This setting enables the aggregation of partition-level statistics and helps the optimizer to choose a better access path.
SYNCVAL	Controls the synchronization of statistics recording across a data sharing group.
SYSTEM_LEVEL_BACKUPS	Let the RECOVER utility use system-level backups as a recovery base.
URCHKTH	Some SAP system processes commit very seldom, which can cause many problems. There is not much that can be done about it because changing the application logic is too tricky. However, in 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 CHKREQ value, adjust URCHKTH to avoid frequent occurrences.
URLGWTH	Additional threshold for identifying long-running, non-committing transactions and reports.
UTIMOUT	The value is reduced from its default (6) due to the relatively large timeout (IRLMRWT) value.
WFDBSEP	Controls maximum workfile separation
WFSTGUSE_AGENT_THRESHOLD	Controls maximum workfile usage
XLKUPDLT	This parameter slightly reduces the overhead of acquiring locks for some statements.
REORG_IGNORE_FREESPACE	When a subset of partitioned table space is reorganized, this setting ensures that the data records still fit into this subset of partitions.  We removed this parameter in Db2 12.
INLISTP	The Db2 query engine optimizes for IN-list predicate up to the number of specified values in this parameter.

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.

### Note

We strongly recommend using 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:

- CDSRDEF = 1

- RETVLCFK = NO
- CTHREAD >= 200
- IDBACK >= 50
- MXQBCE = 1023
- MGEXTSZ = YES
- INDEX\_MEMORY\_CONTROL = AUTO (new with Db2 12)

### 3.3.4 Db2 Buffer Pool-Related Parameters with Recommended Settings

The following table gives recommended settings for Db2 buffer pool-related parameters.

#### ⓘ Note

Do not overcommit real storage. Ensure that your buffer pool's allocated size is compatible with the amount of real storage available.

We recommended using PGFIX and 1 MB frames. To enable 1 MB frames for Db2 buffer pools, set the LFAREA parameter in z/OS adequately.

The buffer pool attribute PGFIX, which enables long-term fixing of pages in real storage, may only be set to **YES** if sufficient real storage is available. In this case, it is recommended to take advantage of page fixing.

This does not change the tablespaces or indices' page size backed by such a buffer pool, which is still 4KB, 8KB, 16 KB, or 32KB.

### Larger Prefetch and Deferred Write Quantity

Db2 uses larger prefetch and deferred write quantity to read from disk into a buffer pool and write from a buffer pool to disk, respectively – should the buffer pool exceed a certain size. The maximum quantities are 256 MB for sequential prefetch and deferred write for SQL processing and 512 MB for utilities. The larger quantities apply to buffer pools with at least the following sizes:

#### Sequential Prefetch

**SQL:** VPSEQT\*VPSIZE > 160 MB

**Utilities:** VPSEQT \* VPSIZE > 320 MB

#### Deferred Write

**SQL:** VPSIZE > 160 MB

**Utilities:** VPSIZE > 320 MB

#### Sizing Recommendation

We recommend taking advantage of these larger quantities to size buffer pools such that they are not slightly below the thresholds but above them.

## Db2 Buffer Pool Parameters

The following are buffer pool recommendations. It would help if you took these as a starting point. Depending on the individual workload, the buffer pool configuration can be adapted and optimized.

For small test and development systems, you can set VPSIZE to smaller values. For large SAP production systems, VPSIZE can be set to values larger than 1 million for essential buffer pools like BP2 and BP3.

For more information about the results of some representative measurements (that can serve you as guidance), see the IBM publication *IBM Z: Performance Report on Exploiting Large Memory for Db2 Buffer Pools with SAP*.

Db2 Buffer Pool-Related Parameters with Recommended Settings for SAP Systems

Buffer Pool	Assign to	VPSIZE	VPSEQT	DWQT	VDWQT
BP0	Db2 cat 4 KB	5000	50	50	10
BP8K0	Db2 cat 8 KB	3000	50	50	10
BP16K0	Db2 cat 16 KB	1000	50	50	10
BP32K	Db2 cat 32KB	1000	50	50	10
BP1	4K work files	25000	90	50	10
BP32K3	32K work files	25000	90	50	10
BP2	4 KB TS	150000	50	30	5
BP8K1	8 KB TS	50000	50	30	5
BP16K1	16 KB TS	10000	50	30	5
BP32K1	32 KB TS	10000	50	30	5
BP3	4 KB indices	200000	40	30	5
BP4	VB protocol	1000	10	70	50
BP40	LOB TS	5000	50	50	10
BP8K8	8 KB indices	20000	40	30	5

\* Optional, in case index compression is used.

For the recommended buffer pool settings in an SAP BW environment, see SAP Note [536074](#).

### 3.3.5 Db2 Installation Panel DSNTIPD: Recommended Values

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

Field	Recommended Value
DATABASES	2500
TABLES	40
COLUMNS	20
VIEWS	1
TABLESPACES	20
PLANS	100
PLAN STATEMENTS	30
PACKAGES	200
PACKAGE STATEMENTS	30
PACKAGE LISTS	2
EXECUTED STATEMENTS	30
TABLES IN STATEMENTS	2
TEMP 4KB SPACES	200M
TEMP 4KB DATA SETS	4
TEMP 32KB SPACES	200M
TEMP 32KB DATA SETS	4

It would help if you considered the storage size-related values given in these tables 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 \[page 256\]](#).

## 3.4 Db2 Continuous Delivery

Db2 12 introduces the concept of *Continuous Delivery*. This means that new functionality is not delivered in releases but rather in a continuous manner in the Db2 12 service stream. As before, Db2 APARs, PTFs, and PUT levels continue to be the primary mechanism. They contain both fixes and new features. To properly control when new functionality is deployed, Db2 introduces the following new levels:

- Db2 maintenance level (ML) contains the Db2 code changes and introduces fixes. It is also known as code level.

- Db2 function level (FL) introduces new Db2 features and functionality
- Db2 catalog level (CL) contains Db2 Catalog changes that are required for specific function levels.
- APPLCOMPAT bind option determines the SQL level of applications. It activates new SQL syntax and freezes the SQL syntax even if an FL is later moved back to an earlier level.

The significant levels are the Db2 maintenance levels, which contain the code changes, and the Db2 function levels, which activate new functionality. Similarly, as before, SAP validates new Db2 maintenance levels based on PUT levels periodically. Also, SAP certifies new Db2 function levels. It is up to the customers to decide when to apply a new maintenance or function level. You can also skip maintenance and function levels.

The required APPLCOMPAT bind option depends on the SQL syntax that is used by an application. Therefore, APPLCOMPAT is controlled by SAP and depends on the SAP NetWeaver release and Db2 version. By default, SAP uses `APPLCOMPAT = V12R1M500`. A new control file `db2applcompat.ini` is introduced that is located in the SAP global directory. It contains an entry `APPLCOMPAT=V12R1M<number>` for Db2 12 and it contains an entry `APPLCOMPAT=V13R1M<number>` for Db2 13.

For the ABAP stack, SUM and SWPM generate the `db2applcompat.ini` file and populate it with an APPLCOMPAT value that is well suited for the particular NetWeaver release. The following function levels and APPLCOMPAT values are required (as in SAP Note [2902423](#) without Db2 Function Level 506):

- In the context of a release change, SWPM and SUM update APPLCOMPAT in an existing `db2applcompat.ini` file.
- SAP DbSI levels from March 2017 and later read the `db2applcompat.ini` file and use the specified collection name to connect to Db2. You should set the profile parameter `db2/db2/autobind` to 1 and follow the instructions in SAP Note [2101963](#). If you cannot use the collection, the SAP DbSI performs an automated bind with the specified APPLCOMPAT level. The naming convention of the collection ID is `SAP<DB2 Connect Release><APPLCOMPAT level><Unicode Indicator>`.

#### ❖ Example

If Db2 Connect V11.1 is used and APPLCOMPAT is V12R1M501, then the collection ID is `SAP1101V12R1M501U`. To trigger a new bind, you can run `R3trans -x`.

For more information about the Db2 12 Continuous Delivery Levels, see SAP Note [2461282](#).

## 3.5 Db2 Data Sharing Design Options for SAP

There are two primary data sharing options for providing a highly available environment for an SAP database using Db2 on z/OS. In either case, to prevent a single point of failure at the hardware level, you can use two or more computer systems (known as Central Electronic Complex or CEC).

These basic data sharing options represent the two fundamental approaches: active/passive and active/active. Different optimizations, such as adding inactive Db2 members on a CEC primarily used for disaster recovery purposes, can be implemented.

The options are:

- **Active/Passive:** Single active Db2 member with passive Db2 standby member (formerly known as data sharing option 0).

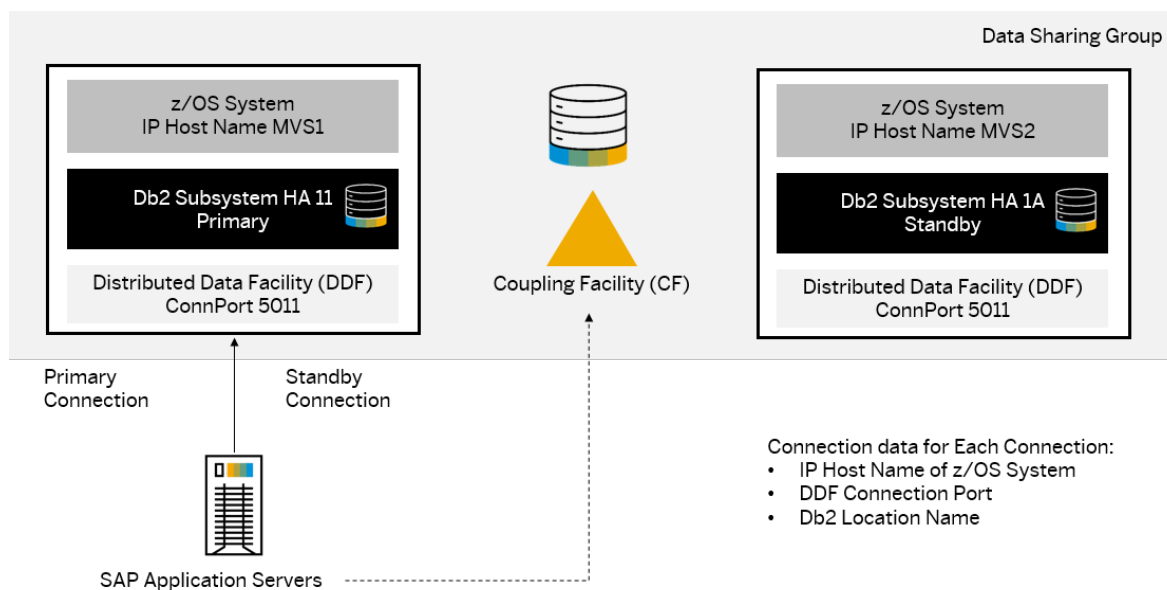
- **Active/Active:** Two or more active Db2 members (formerly known as data sharing option 2).

Since the active/passive configuration virtually eliminates any data sharing overhead (except for the period during which connections are failed over to another member), it provides for the best performance and response times. The fast failover and least downtime are enabled by an active/active configuration, in any case.

## 3.5.1 Active/Passive Data Sharing Configurations

### 3.5.1.1 Single-Active Db2 Member with Passive Db2 Standby Member

This option is chosen most often when high availability is the primary concern. The current hardware is sufficient to handle all database SAP requirements identified from an SAP sizing or a report generated by the **IBM Insight tool for SAP tool**. In other words, your SAP database workload can be handled by one Db2 data sharing member in one CEC.



Active/Passive Data Sharing Configuration: Single Active Db2 Member with Passive Db2 Standby Member

#### ⚠ Caution

The fastest failover performance when a Db2 member goes down is achieved with *two or more active members*. This is because the physical locks (P-locks) held by the different members are usually more granular with two or more active members.

Under normal conditions (with every component working correctly), the passive Db2 member should not use any system resources except as needed to start each component. Even though the idea of high availability is to eliminate human intervention, system programmers (both z/OS and SAP) should check the status of their systems periodically.

Because the active/passive configuration virtually eliminates any data sharing overhead (except for the period during which connections are failed over to another member). It provides for the best performance and response times.

### **3.5.1.2 Cascaded Active/Passive Configuration**

An extension of this configuration is adding a second Db2 standby member running on another CEC. This extension to run both the Db2 primary member and the first standby member on one CEC and the second standby member on another CEC can optimize the CEC configuration used for disaster recovery purposes. This extended configuration provides a cost-effective design for planned and unplanned outages that recognizes the failover types and frequencies but still provides the same availability level as the original design.

The cascading Db2 failover approach allows you to reuse the MIPS on the primary CEC and use CECs of different capacities in the same Sysplex.

The planned outage of a Db2 subsystem or z/OS Logical Partition (LPAR) to apply maintenance is typically most frequent, while an unplanned outage of a CEC is least likely with mean times between failures of more than 40 years.

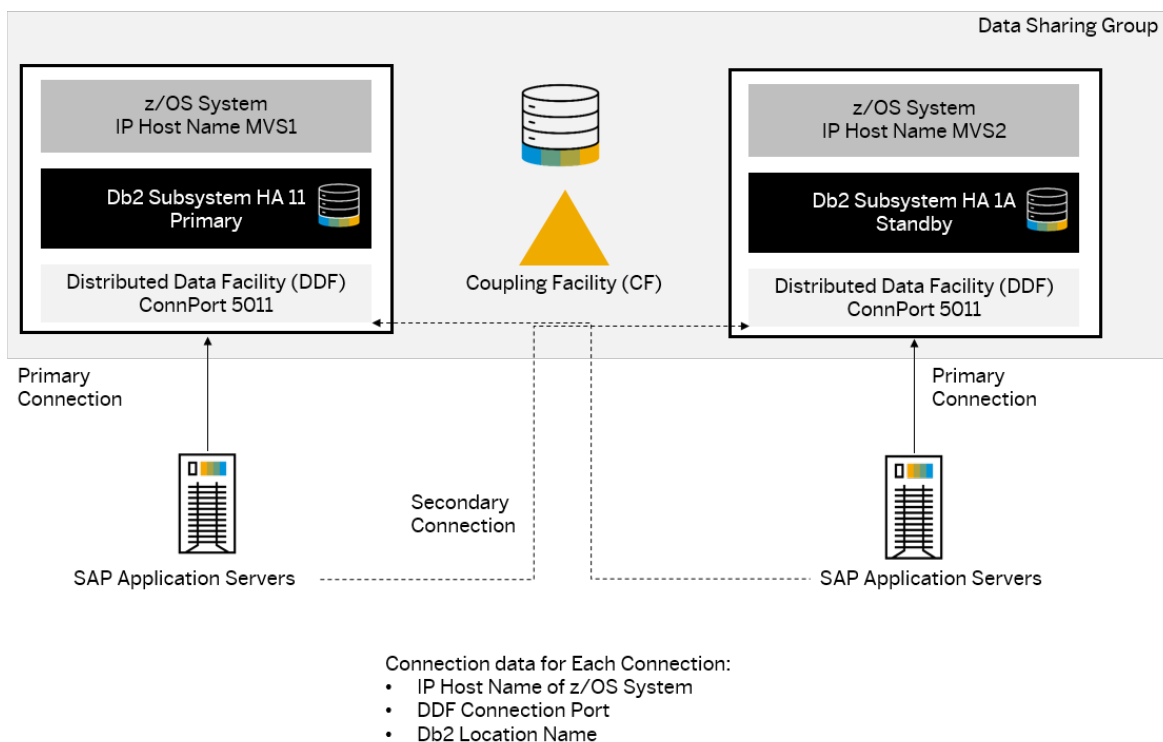
The cascaded failover capabilities for SAP ABAP and SAP Java workload can consider and use the standby members only as a last resort.

## **3.5.2 Active/Active Data Sharing Configurations**

### **3.5.2.1 Two or More Active Db2 Members**

You can select this option to achieve the highest level of availability. In the case of an unplanned outage of a Db2 member, the failover should usually be faster than the active/passive configuration. The retained locks are not at the tablespace or partition level but are more granular.

This option is also typically considered when one Db2 member running in one IBM Z or LPAR cannot handle the SAP database workload from a capacity standpoint. This option also provides the most considerable flexibility as it allows the workload to be spread evenly among multiple Db2 members, reducing the impact of a single member's loss.



#### Active/Active Data Sharing Configuration: Two or More Active Db2 Members

When you consider how to configure Db2 to support more workload, this is often the first thought that comes to mind. In this configuration, the *Db2 connection failover* is set up so that application servers move (connect) from the failing active Db2 member to one of the other active Db2 members.

Suppose the respective host supporting these Db2 members is sized just to fit the expected workload. In that case, one should expect degraded performance when all of the SAP database workloads are concentrated on the surviving DB. This degraded performance could come about due to a lack of CPU capacity or lack of memory. Consider using the IBM Z capacity-on-demand options such as Capacity Backup Upgrade (CBU).

If a single host cannot handle the workload well, you may also consider running four Db2 members with two members on each host. That way, if a Db2 member is stopped, you can still have Db2 members on all hosts being active processing the workload. This configuration also allows you to limit a failing member's scope to a fourth of the entire workload.

It is also possible to dedicate a Db2 member to a specific subset of the overall workload. For example, the Warranty functionality of ERP runs on one Db2 member while the other ERP functionality runs on another Db2 member.

If you decide to run active/active, give careful consideration to sizing the hardware correctly and configuring Workload Manager (WLM) if you run multiple Db2 subsystems in the same LPAR. Suppose you require the same level of performance no matter what state the system is in. Each system should have enough CPU and memory capacity reserved to handle the maximum additional workload on each system.

One of the strengths of z/OS on IBM Z is simultaneously supporting multiple workloads. This is where WLM is essential because it enables you to assign importance to each workload. So in the event of a failover of workload to one surviving Db2 member, you can configure WLM to ensure that the SAP workload receives priority over the other workload, even if it is a non-production SAP workload.



If it is a non-production SAP workload, extra definitions in WLM are required for WLM to distinguish between the SAP systems. The sample WLM definitions in the *Planning Guide for SAP on IBM Db2 for z/OS* assume that you are running one SAP system per LPAR. All of the service classes begin with the prefix SAP. If you want to mix production and non-production workload or run multiple production workloads in the same LPAR, you must extend the sample definitions to control these workloads. One way is to create service classes for each SAP system.

#### ❖ Example

You could create PR1HIGH, PR1MED, and PR1LOW for SAP production system **PR1** and DR1HIGH, DR1MED, and DR1LOW for the SAP development system **DR1**.

A more flexible naming strategy is to put the SAP system name in the service classes. The total number of WLM service classes should not be too high.

## 3.5.2.2 Cascaded Active/Active Configuration

Like the active/passive approach, you can extend the active/active configuration by adding passive Db2 members on another CEC to recognize the failover types and frequencies while providing the same level of availability as the original design. The cascaded failover capabilities for SAP ABAP and SAP Java workload consider and use the standby members only as a last resort.

To ensure that after a planned failover, there is no impact on the SAP workloads that are served by another Db2 member, you may vary the active/active data sharing configuration by dedicating a passive Db2 member for each active Db2 member. If an active Db2 member is stopped, the SAP work processes failover to the assigned passive Db2 member. Other active Db2 members are not impacted by other Db2 threads that compete for resources in the member, such as buffer pools or global statement cache. A further advantage of this approach is that you can easily reuse the CPU resources of the z/OS LPAR of an active Db2 member for planned events such as applying Db2 maintenance. You must prepare extra memory for the period during which the SAP work processes failover, which results in Db2 activity on both members.

## 3.5.3 Number of Data Sharing Groups

A typical SAP core landscape consists of a development system, a quality control system, a stress test system, and a production system. Optionally, one might decide to have a training system, a technical sandbox, or a production support system.

It is common these days for businesses to roll out a larger number of SAP business and technology systems such as SAP Business Warehouse, SAP CRM, or SAP industry solutions. Each SAP system usually requires a separate SAP landscape.

Whatever SAP applications or solutions you are implementing, the total number of SAP systems to build and maintain can add up quickly. It would help if you decided which of these need to be configured for high availability.

You already may have decided that you must configure your SAP production system to be highly available. Therefore, you must configure the production at the very least to run in Db2 data sharing mode.

What about the non-production SAP systems?

It is challenging to answer this question. It depends on your service level agreement (SLA). Some SLAs require that even the development system be highly available. It is very costly to have developers that cannot work because the system is unavailable. Whatever your SLA, we recommend that you configure a minimum of one other SAP system for Db2 data sharing in your promote-to-production landscape. This system allows you to test your code or configuration changes to verify that there are no problems in running them in your data sharing setup. Your production system is not; you want to learn that your changes do not work with Db2 data sharing.

Which non-production system should be configured for data sharing? It depends on how sooner or later, you want to test your changes with data sharing. Applications will run just fine with data sharing when doing single-user or component-level tests. The story might be quite different for stress tests. As the number of users running against different systems increases, you might have a bigger potential for resource contention. So we recommend that your other data-sharing system is either your quality control system or your stress test system if you have one.

We recommend that you consider having a minimum of one additional data-sharing system in each of your SAP landscapes where your business needs require that you have high availability for your production system. Each SAP component shares standard technology, but there is also non-common functionality. A more important thing to keep in mind is the specific landscape configuration work. So it is recommended that you have one other DB data sharing system per SAP Landscape.

One approach can be to consider, to reduce the overall number of Db2 subsystems in your environment: use the SAP Multiple Components in One Database (MCOB) technology to combine specific SAP systems in the same Db2 subsystem or data sharing group. MCOB is often more attractive for non-production systems. Keep in mind that it is often a business requirement in SAP shops to clone, backup, and recover individual SAP systems. You can more easily accomplish this if a Db2 subsystem or data sharing group serves a single SAP system.

## 3.5.4 Number of Sysplexes

So far, you have concentrated on figuring the number of data-sharing systems to ensure that the application changes and SAP basis changes do not cause any problems with data sharing. What about the infrastructure changes, such as coupling facility changes? What system should the infrastructure group use to test their changes? The infrastructure group should consider building a Parallel Sysplex with a data-sharing system independent of the production and non-production SAP systems. It is sufficient to have one Parallel Sysplex for the infrastructure group. There is no need nor benefit to having one technical sandbox system per the SAP landscape. This approach, while consistent, would be cost-prohibitive. Some large customers run the production and non-production SAP systems in separate Sysplexes. Such a configuration allows separating the shared file systems and the scope of System Automation.

## 3.5.5 Number of Data Sharing Members

After you have decided that you need Db2 data sharing, the next question is how many data sharing members are required for each highly available system. The answer to this question depends on the data sharing option you are implementing. The choice you choose depends on the sizing estimate for your proposed production system or systems.

For an active/passive production system, you need only two data sharing members per data-sharing group. The primary data sharing member does all of the work, and the secondary data sharing member is only a standby. This is called *passive data sharing*. This option is valid as long as your workload does not exceed the capacity of the CEC or production LPAR for SAP.

For an active/active production system, you have an active workload distributed between two or more members of a data-sharing group. Assume that you are configuring a two-way data-sharing system. If one system fails or becomes unreachable, you will move the workload to the surviving data sharing member. Suppose you want the system to perform in failover mode with the same throughput level as in non-failover. In that case, you must ensure that there be sufficient extra CPU capacity, memory capacity, and I/O bandwidth to handle the failed overwork in one CEC or LPAR. You must double the capacity of each CEC. A CEC fails so rarely that it may not be so important to have all of that extra capacity ready for a failover situation that will rarely happen.

Another possible option one production system could have three data sharing members in a group. Suppose one data-sharing system fails with this version of option 1. In that case, you can redistribute the workload to one of the surviving members or redistribute the workload evenly among the surviving members. The main choices are to minimize Db2 inter-systems interest or not overallocated Db2 DBM1 virtual storage when making this decision.

We recommend that you configure at least one of your non-production data sharing systems the same way your production system. On the one hand, this makes the management of your system landscape easier. On the other hand, you might detect potential bottlenecks or setup problems within your test environment if it is an exact copy.

## 3.6 Configuring Db2 Logging

The required number and size of the Db2 log data mainly depend on the data change volume of the SAP system that uses a Db2 subsystem or data sharing group. As general guidance, the Db2 log environment should follow the following best practices:

- Use dual logging and dual archiving
  - Keep a minimum of 6 hours of recovery log data in the active log pairs at any time. The objective is to provide some reaction time in case of an archiving problem
  - Always write archive log `COPY1` and `COPY2` to DASD and let DFSMSHsm (or equivalent) migrate them away to tape or VTS. The objective is to avoid contention during concurrent archive log reads
  - Keep at least 48 hours of recovery log data on DASD for fast recovery
- You have the following options:
1. Over-configure the active log pairs (number/size). Write archive log `COPY1` and `COPY2` to DASD, but you can migrate them to tape/VTS at any time
  2. Keep archive log `COPY1` on DASD for 48h before migrating it to tape/VTS

## 3.7 Configuring the Default Sign-On Exit Routine

Since SAP systems use Db2 secondary authorization routines, you must replace the default sign-on exit routine with IBM's sample routine. The installation job used to perform this task is described in the IBM

documentation *Installation and Migration Guide* under *Installation Step 7: Define user authorization exit routines: DSNTIJEX (optional)* section.

## 3.8 Reduction of 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 you will not create a very significant number of data sets. This is beneficial for many database administration tasks as well as for DASD space utilization. The objects created with `DEFINE=NO` are fully supported by all of the functions within the SAP system.

## 3.9 Separating Db2 System Data Sets from Data Sets Used for SAP Tables and Indexes

It is best practice to separate the Db2 system data sets from the data sets used for SAP tables and indexes by assigning different high-level qualifiers (HLQ). This allows you to identify objects quickly and to treat them differently.

All new SAP tables and indexes are assigned to Db2 stogroup `SYSDEFLT` because SAP exploits Db2's implicit object creation feature and Db2 uses `SYSDEFLT` for these objects. You can specify an HLQ for `SYSDEFLT` when it is created as part of the Db2 installation.

The HLQ that Db2 assigns to the data sets of the Db2 catalog and directory is specified via the Db2 system parameter `CATALOG`. These mechanisms provide you the option to configure different HLQs for these two sets of data sets.

Db2 does not support altering the HLQ that is associated with `SYSDEFLT`. Therefore, the Db2 system data sets can be renamed so that they use a different HLQ.

## 3.10 SAP Exploitation of Db2–Supplied Stored Procedures for Database Administration

A stored procedure is a user-written program called by an application with an SQL `CALL` statement. It is a compiled program that is stored on a Db2 server and can execute SQL statements.

Different components of the SAP infrastructure such as CCMS, SAP BW, the SAP upgrade tool, and the SAP installation tool internally call Db2–supplied stored procedures.

## 3.10.1 Stored Procedures Used by SAP

SAP uses a multitude of stored procedures. The following stored procedures are shipped as part of Db2:

- Stored procedures to return information on Db2 and its environment that follow the naming scheme ADMIN\_INFO\_\*, for example, ADMIN\_INFO\_SYSPARM
- Stored procedures to execute Db2 utilities that follow the naming scheme ADMIN\_UTL\_\*, for example, ADMIN\_UTIL\_SCHEDULE
- Stored procedures to execute z/OS JCL jobs that follow the naming scheme ADMIN\_JOB\_\*, for example, ADMIN\_JOB\_SUBMIT
- Stored procedures to manipulate z/OS data sets that follow the naming scheme ADMIN\_DS\_\*, for example, ADMIN\_DS\_WRITE
- Stored procedures to execute Db2, z/OS Unix System Services, or specific z/OS console commands that follow the naming scheme ADMIN\_COMMAND\_\*, for example, ADMIN\_COMMAND\_DB2
- Stored procedure DSNACCOX uses Db2 real-time statistics values to automatically determine tables and indices that benefit from utilities like REORG and RUNSTATS
- Stored procedure GET\_SYSTEM\_INFO returns system information about Db2
- Stored procedure WLM\_REFRESH to refresh a z/OS WLM environment used for Db2 stored procedures
- Stored procedures used by the IBM Data Server Driver for JDBC to retrieve Db2 metadata such as the stored procedures SQLCOLUMNS and SQLTABLES

Also, the stored procedure SAPCL is shipped by SAP. Its primary purpose is to act as a data collector for Db2 performance data. It retrieves this data from the Db2 IFI interface.

## 3.10.2 Setting Up WLM Application Environments for Stored Procedures

Db2 stored procedures, except native SQL stored procedures, run in z/OS WLM application environments. These application environments have different properties, such as the number of TCBS (NUMTCB) or whether they are program-controlled. This allows you to optimize them according to their individual needs. Db2 10 introduces the automatic creation and configuration of the WLM application environments as part of Db2 installation or Db2 release migration. The Db2 job DSNTIJRW creates WLM environments. The Db2 job DSNTIJRT then creates the Db2-supplied stored procedures in these environments. There is also a job to verify the stored procedures (DSNTIJRV) and a job to reflect the stored procedures in SYS1.PROCLIB.

For more information about how to configure Db2 and z/OS for running these stored procedures, see section *Installation step 20: Configure Db2 for running stored procedures and used defined functions* in the IBM documentation.

The WLM environments created by DSNTIJRW follow the naming scheme DSNWLM\_\*, for example, DSNWLM\_UTILS for stored procedure DSNUTILS. It is recommended that you adopt this naming scheme by replacing DSNWLM with the Db2 SSID or group name so that you create dedicated WLM environments for each Db2 subsystem or Db2 data sharing group, respectively. That way, you ensure that there are no dependencies between different Db2 subsystems. For example, if two Db2 subsystems shared a WLM environment, both Db2 subsystems are affected by stopping the WLM environment.

If you would like to limit the number of WLM environments created, you can adapt `DSNTIJRW` not to create the following environments for stored procedures that are not exploited by SAP. These environments, which you may decide not to create, are:

`DSNWLM_DEBUGGER`

`DSNWLM_MQSERIES`

`DSNWLM_WEBSERVICES`

`DSNWLM_XML`

The stored procedure `SAPCL`, which is shipped by SAP, should be assigned to the WLM environment `DSNWLM_GENERAL`.

### 3.10.3 Exception Table

The SAP system will use the `DSNACCOX` exception table.

The SAP recommendations for installing that exception table differ from the official IBM documentation of stored procedure `DSNACCOX`, changing the `CREATE TABLE DSNACC.EXCEPT_TBL` statement into the following:

#### Sample Code

```
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;  
COMMIT;
```

### 3.10.4 Testing and Troubleshooting Stored Procedures

1. SAP produces small trace files for all stored procedures in SAP directory `DIR_HOME` (see transaction `AL11`). These trace files can come in very handy if an error occurs. They are reset each time you invoke a stored procedure. Therefore, they show only the last execution of the stored procedure. If you encountered an error that you want to analyze later, save this file in a different place to prevent it from being overwritten.
2. You can check if stored procedures work correctly by using:

#### Sample Code

```
/D WLM,APPLENV=*
```

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

```
D WLM,APPLENV=*
IWM029I 16.12.02 WLM DISPLAY 540
APPLICATION ENVIRONMENT NAME      STATE      STATE DATA
S23_GENERAL                       AVAILABLE
S23_NUMTCB1                       AVAILABLE
S23_PGM_CONTROL                   AVAILABLE
S23_UTILS                         AVAILABLE
```

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

### Sample Code

```
/V WLM,APPLENV=S23_GENERAL,RESUME
/V WLM,APPLENV=S23_UTILS,RESUME
```

You can also stop the stored procedure itself due to an error. You can check this with the Db2 command:

### Sample Code

```
/-S231 DISPLAY PROC(*)
```

This results in the following output:

```
-S231 DIS PROC(*)
DSNX940I -S231 DSNX9DIS DISPLAY PROCEDURE REPORT FOLLOWS - 871
----- SCHEMA=SYSPROC
PROCEDURE      STATUS ACTIVE  QUED  MAXQ  TIMEOUT  FAIL  WLM_ENV
DSNACCOX
DSNUTILS       STARTED    0    0    1      0    0  S23_GENERAL
DSNUTILU       STARTED    0    0    1      0    0  S23_UTILS
ADMIN_INFO_SQL STARTED    0    0    5      0    0  S23_UTILS
ADMIN_UTL_SORT STARTED    0    0    1      0    0  S23_GENERAL
ADMIN_INFO_HOST STARTED    0    0    1      0    0  S23_GENERAL
ADMIN_INFO_SSID STARTED    0    0    1      0    0  S23_GENERAL
ADMIN_COMMAND_DB2 STARTED    0    0    1      0    0  S23_GENERAL
ADMIN_INFO_SYSPARM STARTED    0    0    1      0    0  S23_NUMTCB1
ADMIN_UTL_SCHEDULE STARTED    0    0    1      0    0  S23_GENERAL
ADMIN_EXPLAIN_MAINT STARTED    0    0    1      0    0  S23_GENERAL
DSNX9DIS DISPLAY PROCEDURE REPORT COMPLETE
DSN9022I -S231 DSNX9COM '-DISPLAY PROC' NORMAL COMPLETION
```

If the status of a stored procedure is other than STARTED, restart the stored procedure to continue:

### Sample Code

```
/-S231 START PROC(DSNACCOX)
```

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

#### Sample Code

```
ICH420I PROGRAM ... FROM LIBRARY ... CAUSED THE
ENVIRONMENT TO BECOME UNCONTROLLED.
BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR DAEMON (BPX.DAEMON)
PROCESSING.
```

you either:

- Did not define a separate WLM environment for the stored procedures DSNACCJS, JQ, JF, JP, and DSNACCUC
- Did not assign one of these stored procedures to this separate WLM environment
- Missed one of the members of SDSNLOAD that needs to be program-controlled

To solve the first two problems, 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 third problem, let your system programmer execute the following RACF commands:

#### Sample Code

```
RDEFINE PROGRAM member ADDMEM('library'//NOPADCHK) UACC(READ)
SETROPTS WHEN(PROGRAM) REFRESH
```

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

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

#### Sample Code

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

where? can be Q or P.

As already described above, DSNACCJQ and JP use the EMCS console. SDSF uses the same console, and if the user submitting the JCL job is at the same time in SDSF, DSNACCJQ and JP cannot execute. You must leave SDSF while submitting JCL jobs from SAP.

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

#### Sample Code

```
ALTUSER userID OPERPARM(ROUTCODE(ALL) AUTH(INFO))
```



## 3.11 Checking the Installation of the Db2 Subsystem

To 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:

### Sample Code

```
dsn system(<DB_ATTACH_NAME>)
```

DB\_ATTACH\_NAME is the Database Attach Name, either the Db2 subsystem name or the Db2 group attachment name, depending on which character 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 the IBM Publication *Db2 for z/OS Installation Guide*.

## 3.12 Ensure Optimal Db2 Settings

### Exception Events

There are different types of events:

- [Lock Escalations](#)
- Long-running transactions ([Long Running URs](#))
- [Deadlocks](#)
- [Timeouts](#)
- [Active Log Shortage](#)

You can display the alerts of these events in their respective analysis monitors. For more information, see the [DB Alert Router \[page 182\]](#).

### Long-Running Transactions

Long-running transactions are either long-running recovery (UR), which changes at least one row, or long-running read-only transactions.

For the long-running URs, the amount of time required to roll back Db2 work depends on the number of Db2 logs written. That implies that long-running work units that accomplish many UPDATE, INSERT, or DELETE activities are rolled back slowly. For example, transactions are rolled back if the application explicitly issues the SQL ROLLBACK statement or at the restart of Db2 when Db2 previously abended.

The negative effects of long-running URs restrict access to the affected resources during rollback and prolong Db2 restart times. This kind of alert is raised when Db2 detects long-running URs depending on the ZPARMS URCHKTH and URLGWTM. For more information, see the IBM 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 and other transactions queue up behind the REORG utility acquires' claim. 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 an alert.

## Using the Correct Db2 System Parameters

Ensure that all of the Db2 system parameters are set as listed in this documentation in the section [Db2 Setup \[page 22\]](#).

## Switching off Unnecessary Db2 Traces

The only trace that should be active during the SAP system installation is 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 you should switch on 2 and 3. 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 significant when heavy DML (INSERT, UPDATE, and DELETE) activity. At SAP system installation time, many Db2 tables are created, and many 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.

## 3.13 Setting Up the Db2 Distributed Data Facility (DDF)

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. It would help if you did not change the name of the procedure.

### Sample Code

```
//*****  
//* JCL PROCEDURE FOR THE STARTUP OF THE  
//* DISTRIBUTED DATA FACILITY ADDRESS SPACE
```

```

// *
// *****
//D620DIST PROC RGN=17M,
// LIB= 'D640.V110.SDSNEXIT'
//IEFPROC EXEC PGM=DSNYASCP,REGION=&RGN
//STEPLIB DD DISP=SHR,DSN=&LIB
// DD DISP=SHR,DSN=SYS1.SCEERUN
// DD DISP=SHR,DSN=D640.V110.SDSNLOAD

```

2. Define the DDF startup mode in the Db2 initialization parameters (DSNTIJUZ). You can define `DDF=NO`, `AUTO`, or `COMMAND`. We recommend you 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 that 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 identify logical units of work within Db2 trace records uniquely. 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 from the value specified for the DRDA PORT.

#### Sample Code

```

// *MODLOG DDF DEFINITIONS / DST
// *JOBPARM SYSAFF=SAPE
//MAKELU EXEC PGM=DSNJU003
//STEPLIB DD DISP=SHR,DSN=D640.V110.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
// *

```

It is possible to create dynamic DDF location aliases to better control when the SAP application servers connect to a specific Db2 member. This can be accomplished using the Db2 command.

**-MODIFY DDF ALIAS(...)** **ADD**. Such an alias can be stopped and started individually. This gives you more granular control if you run both the ABAP and Java stack on the same Db2 system. It also helps if you want to prevent that SAP establishes a database connection to a Db2 member when its DDF component has been started.

## 3.14 Data Stream Encryption

In case the SAP database server and the SAP application servers do not reside in a secure data center with a dedicated network for the SAP data traffic, security requirements may require to encrypt the data stream

between the database server and the application servers. With Db2 for z/OS, this can be accomplished using DRDA data stream encryption. The IBM Data Server Driver for CLI and JDBC support this on the client-side.

Db2 for z/OS takes advantage of the integrated hardware cryptographic co-processors on IBM Z and the z/OS integrated cryptographic service facility (ICSF) to encrypt and decrypt the data stream by hardware means. Therefore, the performance impact of encrypting the data stream is relatively low on the Db2 for z/OS side. Be aware, though, that the performance impact on the client-side is more considerable.

To enable encryption of the data stream, set the SAP profile parameter `db2/db2/authentication` to `DATA_ENCRYPT`. If it is set to `SERVER_ENCRYPT`, only user IDs and passwords sent over the network are encrypted. If you set `db2/db2/authentication` to `SERVER_ENCRYPT_AES`, user IDs and passwords are encrypted using AES. Suppose `db2/db2/authentication` is set to `DATA_ENCRYPT`. In that case, all security-sensitive data is encrypted when sent over the network – for example, SQL statements, query answer sets, or the sets of values that are assigned to the parameter markers of SQL statements during query execution.

For more information about the DRDA data stream encryption, see the IBM documentation for *Db2 for z/OS Administration Guide* and *Db2 Connect User's Guide*.

To ensure that the data stream is encrypted in the network's formatted output, you also need to enable this on the Db2 for z/OS server side. More specifically, you need to activate ICSF in the z/OS LPAR on which the Db2 server is running. When ICSF is active, you can see the following message in the z/OS system log:

#### Sample Code

```
o ...CSFM001I ICSF INITIALIZATION COMPLETE
o ...CSFM400I CRYPTOGRAPHY - SERVICES ARE NOW AVAILABLE
```

The data stream between CLI driver and Db2 for z/OS is encrypted in the formatted output. You can get a `db2trc trace` collected on the client-side (by issuing the following commands: `db2trc on -f <trace_file><run you activities>db2trc offdb2trc fmt -c <trace_file> <formatted trace output file>` `db2trc executable` resides in `adm` subdirectory of the CLI driver).

The following messages indicate that the data stream is encrypted:

#### Sample Code

```
SEND(AR) RQSDSS - Request Data Stream Structure
LL: 74 CORR: 0002 CHAINED: n CONT ON ERR: n
SAME CORR FOR NEXT DSS: n NM: ACCSEC - Access Security
LL: 68 CP: 106D
NM: SECMEC - Security Mechanism
LL: 6 CP: 11A2
Encrypted Userid, Password and Security Sensitive Data (13)
NM: RDBNAM - Relational Database Name
LL: 22 CP: 2110
ASCII: ....@@@@@@@@@@@@
EBCDIC: DSGG
NM: SECTKN - Security Token
LL: 36 CP: 11DC
DATA: 8AFB9F4780801F61 2BA0B264067CF67E
0D04573AC1489BDD 7B1A29A114359D17
```

## 4 IBM Data Server Drivers for CLI and JDBC

The SAP prerequisite for Db2 12 is to use the IBM Data Server Drivers for CLI and JDBC of Db2 Connect 11.1 (out of maintenance since April 2022) or later. You can update CLI and JDBC drivers independently.

### → Recommendation

Use the latest driver version and fixpack certified by SAP (see SAP Note [81737](#)).

### Restrictions

The CLI and JDBC drivers run standalone. Other connectivity setups are not supported. See the IBM documentation for more information about the CLI and JDBC drivers.

SAP Java applications rely on processing LOBs in a transaction until the commit point is reached. With Db2 progressive streaming protocol, LOBs are freed when cursors are closed. Therefore, it is required to disable progressive streaming for JDBC connections. The JDBC connection property `progressiveStreaming` must be set to **no**. You need to specify the parameter in the URL that the JDBC driver uses to connect to the Db2 for z/OS server.

You can modify the URL in the SAP ConfigTool:

- Choose the *Secure Store*.
- Select key `jdbc/pool/<SID>/Url`.
- Add `progressiveStreaming=NO`. Separate the entry with semicolons.
- Choose *Add* and *Save*. SAP installations are setting this value by default.

## 4.1 Directory Structure of the Db2 Client Connectivity

With the CLI and JDBC drivers, SAP uses a directory structure as shown in the following figure:

Directory Structure of the Database Client for SAP NetWeaver  
(including the JDBC Driver)



```
where <OS> is AIX_64, WINDOWS_AMD64, LINUX_S390X_64, LINUX_X86_64, LINUXPPC64_LE (as of 11.5)
```

### Note

In the following, the paths mentioned have been written in UNIX style. For Windows, the same paths are used, however, with backslashes.

## Installation of the CLI and JDBC Drivers

- While you are running SWPM, the CLI and JDBC drivers are installed in the global directory of your SAP system:  
global/db2/<OS>/db2\_clidriver/ (CLI driver)  
global/db2/jdbc (JDBC driver)  
SWPM installs the correct version of the CLI and JDBC drivers that corresponds to your application server (either AIX, Linux, Linux on IBM Z, or Windows).
- If you install an application server on an operating system that is **new** to your SAP system landscape, and **additional** CLI driver for this **new operating system** is installed in the global directory.  
For example: global/db2/<OS>/db2\_clidriver.
- Each time you start the application server, the CLI and JDBC drivers are copied from the global/db2/<OS>/db2\_clidriver and global/db2/jdbc directories to the local exe directory. The active SAP system uses these copies of the CLI and JDBC drivers in the local exe directory.

### Caution

If you start SAP standalone tools (for example, tp or R3trans) from the command line, these tools use the CLI and JDBC drivers in the global directory and **not** the local directory.

## Updating the Global Directory with a New Fix Pack

IBM provides fix packs release dependent regularly. You can update the CLI and JDBC drivers while the SAP system runs. IBM provides so-called *special builds* if an issue needs to be fixed in between such a cycle. These special builds contain the fix for a particular issue in addition to the official fix pack. If such an issue affects the SAP environment, the special build will be provided on the SAP Support Portal and can be downloaded by all Db2 customers. For more information on special builds delivered by SAP and their content, see SAP Note [1927404](#).

Directory Structure of Client Driver provided by SAP

```
|<OS>
| |<clidriver> CLI driver in file clidriver.SAR
| |<Gskit> Global Security Kit, required to setup TLS Certificate Authentication
| |<SAPCAR> SAPCAR executable to extract the CLI driver
| ..
|<jdbc> JDBC driver .jar file
| |<license>
| | |<UNIX>
| | |<Windows>
|<ServerLicensing> Scripts to activate the license when using the Server Licensing Model
```

To install the Fix Pack, do the following:

1. Download and unpack the driver fix pack:
  - All customers should download the Fix Pack from SAP:
    - Backup the directory `global/db2/<OS>/db2_clidriver`
    - You download the CD from the SAP Support Portal and extract it.
    - Use SAPCAR to unpack the `clidriver/clidriver.SAR` file corresponding to your operating system into an empty directory `global/db2/<OS>/db2_clidriver`. There is no top directory in the archive.
    - Copy the content of the extracted `<CD>/jdbc` directory to `global/db2/jdbc`.
  - If there is a solid reason to get the Fix Pack from IBM:
    - You obtain the file from IBM. Make sure to get the Fix Pack certified by SAP.
    - Use `gunzip` and `tar` to unpack the driver fix pack to `global/db2/<OS>`. The top directory in the archive is `odbc_cli`. Beneath the directory, a `clidriver` is created.
    - Rename the directory `odbc_cli/clidriver` to `odbc_cli/db2_clidriver` and move directory `db2_clidriver` one level up to `global/db2/<OS>`.
    - Extract the JDBC driver tar file and the `db2_db2driver_for_jdbc_sqlj.zip` to any temp directory and copy the `db2jcc4.jar` file to the directory `global/db2/jdbc`.
2. You install the permanent license file for the CLI and JDBC drivers. For more information, see [Obtaining a Db2 Driver License \[page 56\]](#).
3. Restart the application server to activate the changes.

### ⚠ Caution

Suppose you have updated the CLI and JDBC drivers in the global directory but still need to restart the application server. In that case, the versions of the CLI and JDBC drivers globally and in the local directory can differ.

## 4.2 Obtaining Licenses for the CLI and JDBC Drivers

### Prerequisites

Regarding licensing, both the CLI driver and the JDBC driver are part of the Db2 Connect product, which requires a proper license. The CLI driver provided by SAP via SAP Support Portal is equipped with a **try-and-buy license** to facilitate the deployment of an SAP system containing the ABAP stack.

Packages downloaded from IBM do not contain any license and can not be used to install an SAP system.

The JDBC driver has a temporary license mechanism as well. The required temporary license is created automatically by SWPM while installing an SAP system based on Java.

### ⚠ Caution

Both the try-and-buy license and the temporary license are valid for **90-days**. During this period, it is crucial to deploy a permanent license. Otherwise, the SAP system stops working when the 90 day period has expired.

If the SAP system runs with a try-and-buy license, warning messages are reported in the SAP system log (transaction SM21).

To run an SAP system, you should add the permanent license files for CLI and JDBC drivers.

If you are running an ABAP-only SAP system, one may assume that the CLI driver's permanent license file is sufficient. However, as soon as SAP lifecycle management tools like SWPM are used, a permanent license for the JDBC driver will be provided. For example, SWPM uses the JDBC driver to perform certain database connection checks.

### Procedure

Depending on how and where you purchase Db2 Connect, the CLI and JDBC drivers' permanent license files are available from different sources.

You need to apply for a license according to your Db2 Connect version. A Db2 Connect 11.1 license is not valid for Db2 Connect 11.5 and vice versa. This applies to CLI and JDBC drivers.

### Permissions

For both IBM and OEM customers:

In all cases, the `<sid>adm` user must have write permission for the license directory.



## License File for CLI Driver

The following describes how to obtain your CLI driver license.

### IBM Customers

If you purchase Db2 and Db2 Connect from IBM, you need to do the following:

1. Retrieve the permanent license file for the CLI from the Db2 Connect image you receive from IBM.
2. Unpack the image and navigate to `db2/license`.
3. Copy the license file to the following directory:  
`global/db2/<OS>/db2_clidriver/license`.  
Depending on your edition of Db2 Connect (Enterprise Edition, Unlimited Edition, Application Server Edition), the license file's name varies. For example, it is `db2consrv_ee.lic` with the Enterprise Edition.

### SAP OEM Customer

If you purchase Db2 and Db2 Connect from SAP as an OEM customer, you need to do the following:

1. Go to SAP Support Portal at <https://support.sap.com/swdc>.

#### Note

The site with the license file is only accessible for OEM customers.

2. Download the zip file containing the license for the CLI driver according to your release.
3. Extract the zip file. The extracted files name is `db2consrv_ee.lic`.
4. Copy the license file to the following directory:  
`global/db2/<OS>/db2_clidriver/license`.
5. You must restart the application server to apply for the new license.

## License File for JDBC Driver

Suppose your SAP system consists of a standalone Java stack. In that case, SAPinst asks you for the license file's location for the JDBC driver during SAP system installation and automatically includes it in the `CLASSPATH` variable. The name of this license file is `db2jcc_license_cisuz.jar`.

The following describes how to obtain your JDBC driver license for the standalone Java stack.

### IBM Customers

If you purchase Db2 and Db2 Connect from IBM, you need to do the following:

1. You can download the JDBC license file in the activation kit from the IBM Passport Advantage website.
2. The `db2jcc_license_cisuz.jar` file is located in the `sql1lib\java` directory for Windows systems or the `sql1lib/java` for UNIX or Linux systems.
3. Transfer the `db2jcc_license_cisuz.jar` in binary mode to `global/db2/jdbc`

### SAP OEM Customers

If you purchase Db2 and Db2 Connect from SAP as an OEM customer, you need to do the following:

1. Go to <https://support.sap.com/swdc>.

#### Note

The site with the license file is only accessible for OEM customers.

2. Download the zip file `db2jcc_license_cisuz.jar` from the JDBC driver.
3. Extract the zip file.
4. Transfer the extracted `db2jcc_license_cisuz.jar` in binary mode to `global/db2/jdbc`.

### Server-side Licensing using Db2 Connect Unlimited License

There is also an option for server-side licensing if you have the Db2 Connect Unlimited Edition. With this approach, you do not use a license file on the SAP application server. Instead, a stored procedure is used to execute this purpose on the Db2 for z/OS server. The name of this stored procedure is `SYSIBM.DB2CON_LIC`. See the IBM documentation *db2connectactivate - Server license activation utility* for more information.

If you use this option, remove the try-and-buy license file from the directory `./global/db2/<OS>/db2_clidriver/license` on the application server. The Db2 CLI driver stops searching for licenses after the first find.

### Troubleshooting

NFS

- If your global directory is shared using NFS and NFS version 3 clients, ensure that the NFS Lock Manager (NLM) is started on the z/OS NFS Server. Add the NLM attribute to the NFS server attributes file.
- If you have only the NFS version 4 clients, is no need for NLM since the NFS version 4 protocol handles locking automatically. Add the NONLM attribute to the NFS server attributes file.

## 4.3 Bind Packages with CLI and JDBC Drivers

If necessary new packages are bound automatically by default. You can set environment variable `db2_db2_autobind` to 0 to switch off autobind. A Collection ID is generated with following naming convention:  
`SAPMMmmapp1compatU`

where:

`MM` is the CLI and JDBC drivers' major version (2 digits)

`mm` is the CLI and JDBC drivers' minor version (2 digits)

`app1compat` is the DB2 12 APPLCOMPAT level like `V12R1M500` (9 digits; this only applies to Db2 12)

`U` is the Unicode label (This is left blank for non-Unicode.)

### Example

The generated Collection ID for the CLI driver in client release 11.5 with Unicode and Db2 12 for z/OS is `SAP1105V12R1M500U`.

## Grant and Specific Collection ID

To grant required Db2 authorizations or bind packages with a specific Collection ID `db2radm` can be used.

### Sample Code

```
db2radm -m db2i [-P <DDF port>] [-L <DDF location>] [-H <database host>] [-S <ssid>] [-U <grantee>] [-Q <schema>] [-C <collection ID>] [-B [yes|no|only|force]] [-G [yes|no|only]] [-u <user> -p <password>|-x <user_record_key> -y <password_record_key>]
```

`db2radm` does the following:

- Binds the CLI packages. The SAP naming convention for the Collection ID binds the packages if not overwritten by option `-C`. If the option `-B no` is specified, the bind is omitted.
- Grants required Db2 authorizations.
- The `db2radm` call grants package usage privileges to grantee user `<GRANTEE>` and schema privileges to the schema user `<SCHEMA>`. For this GRANT, you must specify the administrative user directly with `-u <user>` and `-p <password>` or via SSFS records `-x <user_record_key>` and `<password_record_key>`.

A detailed description of `db2radm` functionality is in SAP Note [843808](#).

### Caution

If you use `-B force` and a connecting thread in Db2 is currently using the Collection ID, your bind is blocked.

## 4.4 Controlling Db2 Failover Using `db2dsdriver.cfg`

Starting with Db2 12, the only supported Db2 failover solution is the native failover capabilities of the Db2 CLI and JDBC drivers, which is also called seamless ACR (automatic client reroute). The failover approach based on `connect.ini` for ABAP is no longer supported.

These two approaches follow the same philosophy, allowing you to configure failover sequences. The native Db2 CLI failover solution is controlled via the configuration file `db2dsdriver.cfg` on the app server.

[Setup of CLI Failover with the SAP Failover Configuration Tool \[page 70\]](#) section describes how to use the SAP Failover Configuration Tool to convert from `connect.ini` to `db2dsdriver.cfg`. For more information about `db2dsdriver.cfg` file, see the IBM documentation *IBM data server driver configuration file*.

### Example

The following is an example of a file used with Db2 CLI failover for SAP ABAP. The parameters `enableAcr` and `enableSeamlessAcr` are essential and need to be both set to `true`. You need to specify fully qualified domain names for parameters `host` and `hostname` in sections `<dsnrcollection>` and `<alternateserverlist>`.

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<configuration>
  <dsncollection>
    <dsn alias="RB9" name="DDFS230" host="ihsapce.wdf.sap.corp"
port="09750"/>
  </dsncollection>
  <databases>
    <database name="DDFS230" host="ihsapce.wdf.sap.corp" port="09750">
      <acr>
        <parameter name="acrRetryInterval" value="0"/>
        <parameter name="affinityFailbackInterval" value="0"/>
        <parameter name="enableAcr" value="true"/>
        <parameter name="enableSeamlessAcr" value="true"/>
        <parameter name="maxAcrRetries" value="3"/>
        <alternateserverlist>
          <server name="S231" hostname="ihsapce.wdf.sap.corp"
port="09750"/>
          <server name="S232" hostname="ihsappe.wdf.sap.corp"
port="09750"/>
        </alternateserverlist>
        <affinitylist>
          <list name="list1" serverorder="S231,S232"/>
        </affinitylist>
        <clientaffinitydefined>
          <client name="cl" hostname="isplp67" listname="list1"/>
        </clientaffinitydefined>
      </acr>
    </database>
  </databases>
</configuration>

```

# 5 Db2 Connectivity

The following sections describe how to configure your database's connectivity on z/OS and your SAP system.

For the connectivity between Db2 and your SAP system, SAP only supports the [CLI and JDBC drivers](#) [page 53].

A JDBC 4.0 compliant driver is shipped with Db2 Connect.

## 5.1 User and Password Management

### DB Connect User and Password

DDF (Distributed Data Facility) checks the user authorization at connect time using a RACF user ID (the DB Connect user ID).

The z/OS group <SCHEMA> must be defined as a Db2 secondary authorization ID of the DB Connect user.

The <SCHEMA> defines the SAP system's Db2 objects' schema as specified by the environment variable `dfs_db2_schema`.

On the client side, the DB Connect user and password are stored in one of the two options:

- Password service - `dbdb2pwd`: for kernel 7.53 and lower
- Secure storage in the file system (SSFS): supported for kernel 7.53 PL 1200 and higher and mandatory as of kernel 7.54

Since the global directory is shared by all application servers of an SAP system, the password file and SSFS are shared as well. Therefore, you only need to control the password file or SSFS in one instance. Db2 for z/OS allows RACF password as well as RACF password phrase authentication. The following table shows which combinations are supported depending on the SAP kernel release and client-side setup.

Password Store	Supported with Kernels	RACF Password (up to 8 characters)	RACF Password Phrase ( > 8 characters)
Password service - <code>dbdb2pwd</code>	7.22, 7.53	Yes	No
SSFS	7.53 PL 1200, 7.54	Yes	Yes

#### → Recommendation

- Create a DB Connect user in RACF without the TSO segment and setting the password or password phrase without an expiration date. If the password expires, new processes can't connect to the

database. The reconnect mechanism for all processes will no longer function until the password is updated.

- Use a DB Connect user for every SAP system. This ensures that if the password or password phrase is supplied incorrectly for one system and the user is revoked by RACF, the effect doesn't propagate to other SAP systems running with the same DB Connect user.
- Connect to Db2 with the same DB Connect user for all SAP instances of a particular SAP system. This ensures the best performance of the dynamic statement cache of Db2.

## 5.1.1 Password Service - dbdb2pwd

Password service `dbdb2pwd` is supported for Kernel 7.22 and 7.53. Password service `dbdb2pwd` is only supported with RACF password.

### → Recommendation

Switch from password service to SSFS or TLS Certification Authentication for a higher security level. For SAP kernel 7.54 SSFS is mandatory if you use password authentication.

## User and Password

The DB Connect user is stored by the following:

- Environment variable:  
`db_s_db2_user`
- Profile parameter:  
`db_s/db2/user`

The password is stored in the encrypted password file `dbdb2pwd` in the respective global directory:

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

## Changing Password

You can change the user's password specified by `db_s_db2_user` by calling the following command as `<sapsid>adm` to create the encrypted password file `dbdb2pwd` in the respective global directory.

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

The password length is limited to 16 characters.

## 5.1.2 Secure Storage in the File System (SSFS)

SSFS is supported for Kernel 7.53 PL 1200 and 7.54.

### Context

The SSFS is part of the SAP NetWeaver Application Server (AS) ABAP. The encrypted storage of the data prevents unauthorized persons or programs being able to access this data.

The ABAP SSFS is located in the following directory:

- UNIX:  
`/sapmnt/<SAPSID>/SYS/global/security/rsecssfs`
- Windows:  
`\\<global host>\sapmnt\<SAPSID>\SYS\global\security\rsecssfs`

The usage of SSFS is activated by the profile and environment variable:

- Environment variable: `rsdb_ssfs_connect = 1`
- Profile parameter: `rsdb/ssfs_connect =1 .`

To switch from password service to SSFS, see the section [Migration to Secure Storage in the File System from Password Service \[page 67\]](#).

### User and Password

The default DB Connect user is stored in SSFS in plain text as `DB_CONNECT/DEFAULT_DB_USER` and the corresponding password in encrypted form as `DB_CONNECT/DEFAULT_DB_PASSWORD`.

### Changing User and Password

You can change your user and password with the following commands:

```
rsecssfx put DB_CONNECT/DEFAULT_DB_USER <DB Connect user> -plain
```

```
rsecssfx put DB_CONNECT/DEFAULT_DB_PASSWORD <password>
```

The percent sign is used as an escape character when calling the `rsecssfx` tool. To insert percent sign (%) You must use the percent sign plus its UTF-8 representation instead (%25).

### More Information

[Secure Storage in the File System \(AS ABAP\)](#)

## 5.1.2.1 Password Rules

SAP on Db2 for z/OS supports passwords with special characters and a length greater than 8 for all database connect users.

### Database Connect Users

Database connect users (e.g., DB Connect user, database administration users) are used to connect to the database. These users must be defined in RACF and are used by various components such as SAP kernel, SL Toolset (SWPM, SUM), SAP standalone tools (R3trans, R3load, db2radm), and CCMS.

### Prerequisites

You must use the RACF password phrase instead of the RACF password as authentication method. Usage of SSFS must be activated for the SAP system. (SAP kernel 7.53 PL 1200 or SAP kernel 7.54 required, supported by SL Toolset SP 37). First activate SSFS and then switch to RACF password phrase. For more information about the setting up secondary database connections in the DBA cockpit see SAP Note [3304720](#).

#### Note

Password service `dbdb2pwd` is only certified with RACF password authentication. Because of RACF password you're limited to eight characters. To activate long password support, the prerequisites described below must be implemented. The rules and limitations described below don't apply for `dbdb2pwd`.

### Secure Storage in the File System

On the SAP Application Server side, you must activate the `SSFS`. The `SSFS` replaces the `dbdb2pwd` password service, used with older kernel versions.

The `SSFS` is set up using tool `rsecssfz`.

### Software Requirements for Secure Storage in the File System

You must use either SAP kernel 7.53 PL 1200 or SAP kernel 7.54 PL 100. As of SL Toolset SP37, SWPM and SUM can support `SSFS` in a transparent way. Both tools set up the SAP system to use `SSFS` when SAP kernel 7.54 is provided.

### z/OS RACF Authentication Method

Activate the z/OS RACF password phrase authentication method for all database connect users.



## Rules for Creating a Password Phrase

The password phrase rules described below differ from the password phrase rules described in the IBM z/OS RACF documentation and are limited to the SAP environment.

Valid characters:

- Alphabetic (a-z, A-Z)
- Numeric (0-9)
- Special characters

Password phrase length:

- Minimum 9 characters
- Maximum 62 characters

A password phrase must contain at least:

- Two alphabetic characters
- Two nonalphabetic characters.

Invalid password phrases:

- Containing the z/OS user ID
- Containing more than two consecutive identical characters

## Allowed List – Special Characters

Allowed List – Special Characters

Character	Sy- mb- ol	Uni- cod- e	As- cii Hex	Comment
Space		U+0 020	20	
Exclamation mark	!	U+0 021	21	
Quotation mark	“	U-0 022	22	
Number sign Hash	#	U+0 023	23	
Dollar sign	\$	U+0 024	24	

Character	Sy- mb- ol	Uni- cod- e	As- cii Hex	Comment
Percent sign	%	U+0 025	25	The percent sign is used as an escape character when calling the <code>rsecssfx</code> tool.  You must use the percent sign plus its UTF-8 representation instead (%25)
Ampersand	&	U+0 026	26	
Apostrophe single quote	'	U+0 027	27	In case you want to specify a single quotation mark within your password phrase, you must specify two single quotation marks for every single quotation mark.
Left paren- thesis	(	U+0 028	28	
Right paren- thesis	)	U+0 029	29	
Asterisk	*	U+0 02A	2A	
Plus sign	+	U+0 02B	2B	
Comma	,	U+0 02C	2C	
Hyphen-mi- nus	-	U+0 02D	2D	
Full stop Period Dot	.	U+0 02E	2E	
Solidus Slash	/	U+0 02F	2F	
Colon	:	U+0 03A	3A	
Less-than sign	<	U+0 03C	3C	
Equals sign	=	U+0 03D	3D	
Greater-than sign	>	U+0 03E	3E	
Question mark	?	U+0 03F	3F	
Commercial at	@	U+0 040	40	
Low line	_	U+0 05F	5F	

Character	Sy mb ol	Uni- cod e	As- cii Hex	Comment
Left curly bracket	{	U+0 07B	7B	
Right curly bracket	}	U+0 07D	7D	

## 5.1.3 Migration to Secure Storage in the File System from Password Service

SAP systems using the dbdb2pwd password service must be converted manually to the secure storage in the file system.

### → Recommendation

Test this procedure in a test/sandbox environment before rolling it out to production.

Do the migration only if the prerequisites are fulfilled.

## Prerequisites

For AS ABAP systems running on Db2 for z/OS, you can use SSFS to store DB Connect user and password. This feature is supported in SL Toolset (SWPM and SUM) as of SP 37. You need Kernel 7.53 PL 1200 or higher or 7.54 to use the SSFS.

## Procedure

### Case kernel 7.53:

- Preparing the SAP instance for the usage of SSFS can be done online without disruption.
- SAP instance restart is required to activate the changes.
- This can be done in a rolling manner for each application server.

### Case kernel 7.54

- Preparing the SAP instance for the usage of SSFS must be done with a kernel lower than 7.54 or during a system outage, as SSFS is mandatory in 7.54.
- The switch to kernel 7.54 can be done only after the SSFS setup is completed.
- An SAP system restart is required to activate the changes (SSFS and kernel 7.54).

Steps for the changeover:

1. Log on to the SAP application server as user <sapsid>adm
2. Exchange the SAP kernel in /usr/sap/<sapsid>/sys/exe/run with a level according to the prerequisites (7.53 or 7.54).
3. Make sure the following environment variables are set:

- SAPSYSTEMNAME = <SAPSID>

**UNIX/Linux:**

**Sample Code**

```
RSEC_SSFS_DATAPATH = /usr/sap/<SAPSID>/SYS/global/security/rsecssfs/data
RSEC_SSFS_KEYPATH = /usr/sap/<SAPSID>/SYS/global/security/rsecssfs/key
```

- **Windows:**

**Sample Code**

```
RSEC_SSFS_DATAPATH=\\<global
host>\sapmnt\<SAPSID>\SYS\global\security\rsecssfs\data
RSEC_SSFS_KEYPATH=\\<global
host>\sapmnt\<SAPSID>\SYS\global\security\rsecssfs\key
```

4. Make sure the corresponding profile parameters are set in DEFAULT.PFL

**Sample Code**

```
rsec/ssfs_datapath = $(DIR_GLOBAL)$ (DIR_SEP) security$ (DIR_SEP) rsecssfs$
(DIR_SEP) data
rsec/ssfs_keypath = $(DIR_GLOBAL)$ (DIR_SEP) security$ (DIR_SEP) rsecssfs$
(DIR_SEP) key
```

5. Create or update the two mandatory entries in the SSFS. Either using db2radm (SAP Note [3308930](#)):

**Sample Code**

```
>>db2radm -m pwdmig
```

or rsecssfx

**Sample Code**

```
>>rsecssfx put DB_CONNECT/DEFAULT_DB_USER <DB Connect user> -plain
>>rsecssfx put DB_CONNECT/DEFAULT_DB_PASSWORD <password>.
```

6. Check the content of SSFS: rsecssfx list
7. In the login environment of the <sapsid>adm user, change the value of the environment variable rsdb\_ssfs\_connect from 0 to 1:

**Sample Code**

```
rsdb_ssfs_connect = 1
```

UNIX/Linux: You can find the environment variable in the .sapenv\* login scripts.

8. Log out and log in again as user <sapsid>adm to activate the changed environment for the user.

9. Check the database connection: `R3trans -x`  
If the changeover was successful, the trace file `trans.log` must contain the following message:

#### Sample Code

```
4 ETW000 ==> connect info for default DB read from SSFS
```

10. Restart the SAP instances on this application server.
11. Repeat steps 7–10 for each application server of this SAP system.
12. Change `rsdb/ssfs_connect` from 0 to 1 in `DEFAULT.PFL`:

#### Sample Code

```
rsdb/ssfs_connect = 1
```

If set, the profile parameter setting won't overwrite the corresponding environment variable (see next step). As `rsdb/ssfs_connect` is always set, the change in `DEFAULT.PFL` can be done without impacting other instances.

13. After all, instances have been successfully converted to SSFS, check if you see the following message in the current developer trace files of all instances (e.g., checking `dev_w0` is sufficient):

#### Sample Code

```
B ==> connect info for default DB read from SSFS
```

If successfully checked, remove or rename the `dbdb2pwd` file in the global directory.

#### Note

If you start your SAP instance using the `startsap` script, the script tests the DB connection using `/usr/sap/<sapsid>/<instance>/exe/R3trans`. This is still the old kernel since `sapcpe` has yet to be executed. If `dbdb2pwd` file has already been removed and the old kernel doesn't support SSFS, the DB connection test fails, and the script stops.

Exchange the kernel in `/usr/sap/<sapsid>/<instance>/exe` manually to bypass this error.

SWPM uses SSFS for all installations with kernel 7.54.

SUM doesn't change the current setup. If the target environment uses kernel 7.54, an error message is raised if SSFS still needs to be set up.

## 5.1.4 Migration to TLS Certificate Authentication

The increase in external and insider threats and regulatory requirements now compels enterprises to implement more secure technologies. One of the most important is using X.509 standard for public key certificates with the Transport Layer Security (TLS) protocol as an addition to or a complete replacement of user and password credentials.

The setup requires deep knowledge in z/OS, RACF, Db2 for z/OS, and the setup of the database connectivity of an SAP NetWeaver system.

The Best Practices document [TLS with Client Certificate Authentication for SAP Application Server Connections to Db2 on IBM Z](#) describes all required steps in detail.

## 5.2 Setup of CLI Failover with the SAP Failover Configuration Tool

### Prepare the Technical Setup (Technical Prerequisites)

You must complete the following technical setup before using any of the features described below.

#### System Prerequisites

Before you start to use the *Failover Configuration* tool and the **CLI Failover**, make sure you have a system with at least CLI driver 10.5 FP5, CLI Failover enabled DbSI, and the monitored system fulfills the prerequisites.

We recommend using Db2 Connect v11.1 FP4 SB38143 or higher. This level of Db2 Connect is also required for Db2 12 for z/OS.

For more information, see SAP Note [1777242](#).

#### Remote Systems

To use the *Failover Configuration* tool remotely, set up the database, and establish an RFC Connection. For each remote system, call transaction Db2 and choose *Db2 Connections*.

#### Support Packages/Correction Instructions

The support packages or correction instructions in SAP Note [1522391](#) and its referenced SAP Notes (for example, SAP Note [1777242](#)) need to be installed in the local and destination systems.

To check your support package level, call the menu **System > Status**.

For JAVA based systems, see SAP Note [1426172](#).

#### Defaults

To start the *Failover Configuration* tool, use transaction Db2, and choose **Configuration > Failover Configuration**.

If all prerequisites are met, the monitored system's current failover configuration is being read to identify the current failover mechanism; the tool refers to the presence of profile parameter `db2/dsn_alias`.

You must set the profile parameter `db2/dsn_alias` to a valid value to identify the connection attributes in the generated failover configuration. The SAP recommended value is the **SAP System ID**.

To ensure that SAP tools like `R3trans` work, the environment variables `db2_db2_dsn_alias`, and `DB2SDRIVER_CFG_PATH` needs to be set to the alias name used in `db2dsdriver.cfg` and to the path where the Db2 CLI driver resides, respectively.

## Purpose

This *Failover Configuration* tool eases the configuration of the failover setup for HA environments. The tool adds transparency to the setup and shows all relevant information understandably. It relieves the user from writing the files manually, prevents spelling errors in the process, and informs about possible configuration errors.

Each application server will know a sequence of data sharing members. The application server will try to establish a connection to the first member in the order.

When the CLI driver cannot reach the server, the CLI driver tries to establish a connection to one of the other data sharing members in the sequence. An affinity list defines the order of data sharing member. One or more application servers share one affinity list.

## Recommended Usage

### Changing CLI Failover Configuration on Enabled Systems

The following steps are recommended if changes are necessary to CLI Failover enabled systems:

1. The current configuration is loaded on startup
2. Apply changes
3. React to warnings and errors
4. Perform sanity check
5. Select *Save* function
6. Pay attention to popup and confirm
7. Restart the application servers or temporarily set profile parameter `rdisp/wp_auto_restart` to a non-zero value to enforce a smooth restart of all work processes.

### Migration to CLI Failover

This tool performs the migration from the legacy `connect.ini` based failover mechanism to the new CLI based failover mechanism.

The following steps are necessary for a successful migration:

1. Prepare your environment:
  - Make sure that all Db2 data sharing members have the same DDF location alias.
2. Launch the new *Failover Configuration* tool and call the transaction `db2`. Choose **► Configuration ► Failover Configuration ►** and port.
  - The transaction automatically reads the current `connect.ini` file.
3. Review the loaded failover configuration
  - Adjust missing Db2 data sharing members
  - Define new affinity list and review those present
  - Check the application server tab and make sure each application server has the right affinity list assigned.
  - Go to the parameter tab and adjust those recommended if necessary.
  - Adjust the DDF Location Name to the value assigned in step 1.a.
  - Perform sanity check and react to errors

4. Choose ABAP – db2dsdriver.cfg from the *Output Format* list box.
5. Click on the *Save* button and confirm the writing of the new failover configuration to disk.  
The new failover configuration is now written to disk, but the SAP system is still running with the legacy failover mechanism.
6. Confirm the applied changes by testing the new configuration
7. Adjust the profile parameter `db2/dbs/db2/dsn_alias` to the SAP system ID. SAP recommends using transaction `RZ11` for this procedure.  
If SAP Note [211103](#) is installed, SAP Profile parameter `/db2/dbs/db2/dsn_alias` is no longer required.
8. Add environment variable `db2_dbs_dsn_alias` and `DB2DSDRIVER_CFG_PATH` to `<sid>adm` environment.
9. Test the new configuration. You can do this with the program `db2cli`, which can be executed from the application server's command line.

### Sample Code

```

${DB2_CLI_DRIVER_INSTALL_PATH}/bin/db2cli validate -dsn $
{db2_dbs_dsn_alias}
${DB2_CLI_DRIVER_INSTALL_PATH}/bin/db2cli validate -dsn $
{db2_dbs_dsn_alias} -connect -user <user> -passwd <password>

```

10. Restart the application server. Check work file `dev_w0` for the following message: `Db2Trc: db2/dbs/db2/dsn_alias is set (sgc) -> we use the Db2 Connect failover feature`
11. Adjust the SAP profile and configure timeouts if needed. If the IP belongs to the `hostname` in the `<alternateserverlist>`, this is not active when connecting or reconnecting, and a connect returns the error `ConnectionTimeout`. One way to check this is to run `telnet <hostname>` or `ssh <hostname>` on the application server when the IP is inactive. The Operating System-level timeouts vary from about 21 seconds under Windows, about 75 seconds under AIX up to about 125 seconds under Linux. If you consider that the CLI driver does three retries, this will result in a long time until you will try a connection to the next server.
12. 1. Set `tcpipConnectTimeout` to 5 (seconds) and set `ConnectionTimeout` to  $1 + \text{<value resulting from below formula>}$

### Sample Code

```

(<Timeout> + acrRetryInterval) * maxAcrRetries * (<# of Db2 members in
affinity list> - 1)

```

The recommendation for a 2 Db2 member setup with the default values for `acrRetryInterval` and `maxAcrRetries` is:

- Set `ConnectionTimeout` to **16**(seconds) =  $1 + (5 + 0) * 3 * (2 - 1)$
- Set `tcpipConnectTimeout` to **5**(seconds)

The recommendation for a 4 Db2 member setup is:

- Set `ConnectionTimeout` to **46**(seconds) =  $1 + (5 + 0) * 3 * (4 - 1)$
- Set `tcpipConnectTimeout` to **5**(seconds)

We are setting `ConnectionTimeout` to smaller values as recommended results in not attempting all members for failover.

Alternatively, if you already have defined in your SAP profile `db2/dbs/db2/pcon_timeout` or `scon_timeout`, then you may set `ConnectionTimeout` to the value of `pcon_timeout` or `scon_timeout` and set `tcpipConnectTimeout` to the value resulting from this formula:



### Sample Code

```
tcpipConnectTimeout<= (((ConnectionTimeout - 1) / (<# of Db2 members in affinity list> - 1)) / maxAcrRetries) - acrRetryInterval
```

- Place the two parameters into the database section:

### Sample Code

```
<databases>
  <database name="DDFSGR" host="ihsapdh.wdf.sap.corp"
port="10860">
  <parameter name="ConnectionTimeout" value="16" />
  <parameter name="tcpipConnectTimeout" value="5" />
  <acr>
```

- After migration from SAP Failover to CLI Failover, remove from SAP Profile:
  - `dbs/db2/pcon_timeout`
  - `dbs/db2/scon_timeout`
- Environment variable `SAPLOCALHOST` is not necessary. You can use virtual hostnames in client affinity definition in `db2dsdriver.cfg`.
- Check SAP Note [2082467](#) regarding TCP/IP `KEEPALIVE` parameter.

## Legacy Failover with connect.ini

This works as described in the [Changing CLI Failover Configuration on Enabled Systems](#) section, but the DbSI does not need to be instructed to reload the changes. They are applied as soon as a work process is going to establish a new connection. The recommended parameters entered automatically will be ignored.

## Troubleshooting

The following table provides some solutions to symptoms or Issues:

Symptom/Issue	Solution
Database connection missing	Set up a database connection using transaction <i>Db2</i> and choose <i>DB Connections</i> . To specify DB values, see section <a href="#">Prepare the Technical Setup (Technical Prerequisites)</a> .
Exceptions in the message window and missing RFC Connection	Define RFC Connection to the monitored system and make it available in the system configuration.
Exceptions in the message window and file not parsed with RFC Connection present	Function modules for reading and writing configuration files must be present in the monitored system. Upgrade system to appropriate Support Package.

Symptom/Issue	Solution
SAP profile parameter <code>Conflicting definitions in the SAP profile and <code>db2dsdriver.cfg</code> may disconnect the failover capability.</code>	<p><b>Example:</b></p> <p>SAP profile parameter:</p> <ul style="list-style-type: none"> <li><code>db2/db2/pcon_timeout = 16 sec</code> (Maximum overall time to establish a SAP primary connection, corresponds to CLI parameter <code>ConnectionTimeout</code>)</li> <li><code>db2/db2/scon_timeout = 16 sec</code> (Maximum overall time to establish a SAP secondary connection, corresponds to CLI parameter <code>ConnectionTimeout</code>)</li> </ul> <p><code>db2dsdriver.cfg</code> parameters:</p> <ul style="list-style-type: none"> <li><code>tcpipConnectTimeout = 6 sec</code> (Individual time per connection - open a socket - attempt)</li> <li><code>maxAcrrtries = 3</code> Result: <math>6 \text{ sec} \times 3 = 18 \text{ sec} &gt; 16 \text{ sec}</math></li> </ul> <p>Overall connect time exceeded, and the SAP work process stops further connection attempts. The SAP work process is not connected to any Db2 member and not operational. For more information, see SAP Note <a href="#">1465252</a></p>

For more information about [Failover Configuration](#), see the following SAP Notes:

- SAP Note [1522391](#) New Failover Configuration
- SAP Note [1777242](#) Db2 z/OS: Cli client based seamless failover
- SAP Note [1426172](#) Java Failover

## 5.3 CLI Failover - Planned or Unplanned Outage of a Db2 Member

This section describes the behavior of database connections when CLI Failover is configured, and a Db2 member is either stopped intentionally or if the member fails. The sample uses a simple 2-way data-sharing system to describe the mechanism but works similarly for more members or more application servers.

### Context

#### Sample system configuration:

Assume you have a 2-way Db2 data-sharing system with data sharing members A and B.

The SAP AppServer S1 uses CLI Failover with the following parameter definitions in `db2dsdriver.cfg`:

S1 uses member **A** as **primary** and member **B** as **secondary** connection

The following parameters are set:

```
affinityFailbackInterval=300
enableAcr=true
enableSeamlessAcr=true
acrRetryInterval=0
maxAcrRetries=3
```

Scenarios 1a) and 1b) describe a planned maintenance scenario in which the Db2 operator wants to stop Db2 member A, for example for maintenance purposes.

### Szenario 1a: Start planned maintenance for member A

Use `STOP DDF MODE ( QUIESCE )` on member A to stop database connections from S1 to A

- **Expected behavior:**
  - Idle connections from S1 to member A (which are on a COMMIT point) are terminated immediately.
  - Active connections from S1 to member A are terminated as soon as they reach a commit point

#### Note

Long-running SAP connections, like, e.g., SAP Batch, might take a while to reach a commit point.

- Active connections from S1 reconnect to member B
- **Expected timeframe:**  
For the connection switch and successful stop of DDF: **immediately** (assuming all connections are at a commit point).

You can now stop db2 member A completely, and you can apply maintenance.

### Szenario 1b: Planned maintenance finished on member A

Restart member A, restarting DDF as well.

- **Expected behavior:**
  - Active connections from S1 to member B switch back within 300 seconds (`affinityFailbackInterval`) when they reach a commit point.

#### Note

Long-running SAP connections, like, e.g. SAP Batch, might take a while to reach a commit point.

- Idle connections from S1 to member B are terminated by Db2 (server) after a maximum of 20 minutes - which is the Db2 server idle thread timeout. For more information, see the IBM documentation *PK69339: AUTOMATIC KEEP DYNAMIC DISTRIBUTED THREAD REFRESH*.
- **Expected timeframe:**  
You need 0-20 minutes for the switchback, assuming all connections are either idle or at a commit point.

#### Note

If SAP work processes have no connection to Db2 DDF, the Db2 thread recycle had dropped the connections, and afterward, you dispatched no SAP work to that process.

If such "connectionless" SAP work processes get work dispatched, they will connect to either A or B, whichever is active at that time. Therefore it may even happen that they get work dispatched after A is active again and connect to A, and then they had never done a failover and failback.

Scenario 2a) and 2b) describe an unplanned failure (outage) of Db2 member A.

### Scenario 2a: Unplanned failure of member A

Db2 member A fails

- **Expected behavior:**
  - All connections from S1 to member A are terminated immediately
  - Connections that are at a COMMIT point reconnect seamlessly to member B
  - READ-ONLY connections that are not at a COMMIT point reconnect seamlessly to member B
  - For connections that are not at a COMMIT point and have UPDATES pending, the SAP work process will see a ROLLBACK and then reconnect to member B

### Scenario 2b: Member A available again

Restart member A, restarting DDF as well

- **Expected behavior and timeframe:**
  - Same as in scenario 1b above

#### 📌 Note

For scenarios 1b and 2b:

- If you set `affinityFailbackInterval = 0`, then the timeframe is 0-60 minutes. This is because the Db2 server thread recycle timeout is 60 minutes. For more information, see the IBM documentation *PK69339: AUTOMATIC KEEP DYNAMIC DISTRIBUTED THREAD REFRESH*. The thread recycles mechanism terminates active connections to member B, which are at a commit point after a maximum of 60 minutes. These connections will then reconnect to member A.
- Any SAP work process that restarts will immediately connect to member A. Reasons for such a restart may be the canceling of the individual SAP work processor because the complete SAP instance is restarted.

## 5.4 Db2 Failover for SAP Java Applications

The Db2 JDBC Driver provides native support for database failover that can be exploited with SAP Java applications.

The following section describes how you enable high availability for your Application Server Java in the case of planned and unplanned outages of a Db2 for z/OS subsystem that runs in a data-sharing environment. High availability enablement is based on the seamless and cascaded failover feature of the IBM Data Server Driver for JDBC (JCC driver). The IBM Data Server Driver for JDBC reroutes connections to a standby member.

The section below describes the steps that are necessary for enabling planned and unplanned failovers for Java instances.

Proceed as follows:

1. Stop your Java application server.
2. Provide your connection profile in the global directory, for example, `/usr/sap/JHA/SYS/global`.
3. Create the config file specified in the URL, which controls the Db2 data sharing failover

#### Sample Code

```
<?xml version="1.0"?>
<configuration>
  <dsn_collection>
    <dsn port="8162" host="halhost.sample" name="DDFDHA" alias="JHA"/>
  </dsn_collection>
  <database>
    <database port="8162" host="halhost.sample" name="DDFDHA">
      <ACR>
        <parameter name="enableAcr" value="true"/>
        <parameter name="enableSeamlessAcr" value="true"/>
        <parameter name="enableAlternateServerListFirstConnect"
value="false"/>
        <alternate_server_list>
          <server port="8162" name="server1" hostname="halhost"/>
          <server port="8162" name="server2" hostname="ha2host"/>
          <server port="8162" name="server3" hostname="ha3host"/>
        </alternate_server_list>
        <affinity_list>
          <list name="list1"
serverorder="server1,server2,server3"/>
          <list name="list2"
serverorder="server2,server3,server1"/>
          <list name="list3"
serverorder="server3,server1,server2"/>
        </affinity_list>
        <client_affinity_defined>
          <client name="client1" hostname="appserver1"
listname="list1"/>
          <client name="client2" hostname="appserver2"
listname="list2"/>
          <client name="client3" hostname="appserver3"
listname="list3"/>
        </client_affinity_defined>
      </ACR>
    </database>
  </configuration>
```

The DDF ports, which are identical in a data sharing group, need to be listed for all members.

If you want to activate automatic failback after 5 minutes, for example you may add to the ACR section the following parameter:

```
<parameter name="affinityFailbackInterval" value="300"/>
```

There must be no blanks in the URL; parameters starting with a blank are ignored.

By default, the JDBC driver tries to connect to a Db2 member three times in succession. Between the connection attempts, it waits 0 seconds. If you would like to adjust this, you can set in the ACR section the parameters `<parameter name="maxAcrRetries" value="3"/>` and `<parameter name="acrRetryInterval" value="0"/>` to different values.

4. Change the URL of the database connection.

## ❖ Example

In the following example, we assume these parameters:

Parameter:	Value:
SAPSID	JHA
DB2 SSIDs	DHA1, DHA2, DHA3
LOCATION	DDFDHA
DDF PORTs	8162,8162,8162
Collection ID (*)	SAPJJHADDFDHA
SCHEMA	SAPJAVA
HOSTNAME of DHA1	ha1host
HOSTNAME of DHA2	ha2host
HOSTNAME of DHA3	ha3host

- Start the *ConfigTool*
- Choose *Secure Store*
- Choose key `jdbc/pool/<SID>/Url`
- Change the URL to use the connection profile by specifying the config file's file name via parameter `dsdriverConfigFile`.  
For AIX and Linux the URL is: `jdbc:db2:///JHA:dsdriverConfigFile=/usr/sap/JHA/SYS/global/config.xml;`  
`keepDynamic=yes;currentPackageSet=SAPJJHADDFDHA;currentSQLID=SAPJ1;queryCloseImplicit=2; <possible more parameters>`  
For Windows the URL is: `jdbc:db2:///JHA:dsdriverConfigFile=\sapglobalhost\sapmnt\JHA\SYS\global\config.xml;`  
`keepDynamic=yes;currentPackageSet=SAPJJHADDFDHA;currentSQLID=SAPJ1;queryCloseImplicit=2; <possible more parameters>`  
Omit any blanks in the URL; parameters starting with a blank are ignored.  
There is no need to change the values for `currentPackageSet` or `currentSQLID` to enable high availability for the Java Stack. Keep the setting `keepDynamic=yes`.  
Remove the following parameters from your URL if the configuration profile parameters replace them. If you do not remove them, the corresponding values defined in the configuration profile are not used:  
`enableSeamlessFailover`  
`enableClientAffinitiesList`  
`clientRerouteAlternateServerName`  
`clientRerouteAlternatePortNumber`  
Keep the other parameters.
- Choose *Add*
- *Save* your changes in the *ConfigTool*.

## 5. How to Handle Connection Timeouts

Configure a connection timeout if needed. If the IP belongs to the `hostname` in the `<alternate_server_list>`, this is not active when work processes connect or reconnect, and when a connect returns the error, `ConnectionTimeout`. One way to check this is to run `telnet <hostname>` or `ssh <hostname>` on the application server when the IP is inactive. The operating system level timeouts vary from about 21 seconds under Windows, about 75 seconds under AIX and about 125 seconds under Linux. If you consider that the JDBC driver performs three retries, this will result in a long time until you will attempt a connection to the next server. For Linux, this would amount to:

```
(125 + 0) * 3 * (2 - 1) = 375 sec
```

Therefore, it is recommended to set `acrRetryInterval` to 5, consistent with the recommendation for CLI. Then set `connectionTimeout` to `1+ <value resulting from below formula>`. Note the lower case 'c' at the beginning of the parameter name.

```
(<Timeout> + acrRetryInterval) * maxAcrRetries * (<# of Db2 members in affinity list> - 1)
```

The recommendation for a 2 Db2 member setup with the default values for `acrRetryInterval` and `maxAcrRetries` is:

```
Set connectionTimeout to 16(seconds) = 1 + (0 + 5) * 3 * (2 - 1)
```

The recommendation for a 4 Db2 member setup is:

```
Set connectionTimeout to 46(seconds) = 1 + (0 + 5) * 3 * (4 - 1)
```

If you set `connectionTimeout` to less than the recommended value, then the result will be that you will use not all Db2 members for failover.

### Note

You must specify `connectionTimeout` as a URL parameter.

As noted above `connectionTimeout` must be specified in the URL (without blanks). Here a sample URL part: `queryCloseImplicit=2;connectionTimeout=16;<possible more parameters>`  
The parameters' values are set according to the (known) formula: `maxAcrRetries * acrRetryInterval < connectionTimeout` and should result in a reconnect after 16 seconds. And should result in a reconnect after 16 seconds. Following sample shows you in which XML section you need to add the parameters in the `config.xml` file:

### Sample Code

```
<database port="8162" host="halhost.sample" name="DDFDHA">
  <ACR>
    <parameter name="enableAcr" value="true"/>
    <parameter name="enableSeamlessAcr" value="true"/>
    <parameter name="enableAlternateServerListFirstConnect"
value="false"/>
    <parameter name="affinityFailbackInterval" value="300"/>
    <parameter name="maxAcrRetries" value="3"/>
    <parameter name="acrRetryInterval" value="5"/>
  </alternate_server_list>
  ...
</database>
```

## 6. Restart your Application Server Java.

- To trigger planned failover, stop the DDF on one member gracefully: "STOP DDF MODE(QUIESCE)" - default mode. At the end of a transaction, a thread will reconnect to the next member if no resources are held any longer.

### Note

Setting `enableClientAffinitiesList` to true in the URL cause that only the members specified in the URL -and in that order- will be considered. If you do not remove it from the URL, then the server orders of the `config.xml` are not used.

If you run two or more Db2 members of a data-sharing group on the same z/OS LPAR, you can proceed as described in SAP Note [1398993](#) to ensure that you can identify a member uniquely.

## Additional Information

### Note

There is an internal mapping of the six allowed parameters according to JCC properties.

<code>config.xml</code> parameter	JCC property
<code>&lt;parameter name="enableAcr" value="true" /&gt;</code>	<code>enableClientAffinitiesList</code>
<code>&lt;parameter name="enableSeamlessAcr" value="true" /&gt;</code>	<code>enableSeamlessFailover</code>
<code>&lt;parameter name="enableAlternateServerListFirstConnect" value="false" /&gt;</code>	<code>enableAlternateServerListFirstConnect</code>
<code>&lt;parameter name="affinityFailbackInterval" value="300" /&gt;</code>	<code>affinityFailbackInterval</code>
<code>&lt;parameter name="maxAcrRetries" value="3" /&gt;</code>	<code>maxRetriesForClientReroute</code>
<code>&lt;parameter name="acrRetryInterval" value="0" /&gt;</code>	<code>retryIntervalForClientReroute</code>

Because of this mapping and the URL parameter takes precedence, you need to remove the JCC properties from the URL to activate the settings in the `config.xml`.



# 6 The DBA Cockpit

Your local SAP system is connected, when you call transaction `DBACOCKPIT` for the first time. However, you must configure `SAPCL`, as described in the following sections.

If you want to monitor remote SAP systems, you must add them to the list of systems in your DBA Cockpit **before** you configure `SAPCL`. For more information, see [Adding a Remote SAP System Database Connection \(optional\) \[page 88\]](#).

## ⓘ Note

You cannot carry out some monitoring tasks offered in the DBA Cockpit in a remote environment.

## 6.1 Introduction to the DBA Cockpit

The DBA Cockpit is a platform-independent tool that you can use to monitor and administer your database. The advantage of using the DBA Cockpit is that it has been specifically designed for the administration and monitoring of databases in an SAP system landscape. Therefore, the DBA Cockpit functions primarily support database administrators in adapting their databases for the workload of SAP systems. The DBA Cockpit eases database administrators' work because all-important performance tuning, monitoring, and administration tasks are available in a single transaction.

### Central Monitoring of the Databases in an SAP System Landscape

The DBA Cockpit is part of SAP NetWeaver ABAP systems and integrated into the SAP Solution Manager. You can run the DBA Cockpit as part of your system administration activities in SAP Solution Manager. The DBA Cockpit is optimized to handle administration and monitor your entire system landscape databases from a central system. In particular, you can use the DBA Cockpit to manage the configuration of databases centrally. You can administer and monitor remote databases from the DBA Cockpit using remote database connections. For more information, see [The DBA Cockpit, on a Local System, and SAP Solution Manager \[page 82\]](#) section.

### Administration and Monitoring Functions for IBM Db2 for z/OS

If you administer and monitor IBM Db2 for z/OS databases with the DBA Cockpit, the following functions are available:

- **Performance monitoring**

For example, you can display performance and workload statistics, analyze SQL statements, perform time spent analyses, or display performance information about database objects.

- **Space monitoring and administration**  
The DBA Cockpit allows you to watch your database's space consumption, including database objects such as tables, indexes, tablespaces, or containers. You can analyze space allocation and perform administration activities to change the storage layout of your database.
- **Backup and recovery overview**  
You get an overview of all performed database backups.
- **Database configuration**  
In this area, you get an overview of your database configuration. You can also change the database configuration, including configuration settings.
- **Job scheduling**  
Direct access to the DBA Planning Calendar, the DBA log, scheduled Db2 tasks, and other tools allows you to plan and keep track of all critical jobs for the databases.
- **Alert monitoring**  
You can view alerts in the alert monitor and enable notifications for violations of database alert thresholds.
- **Diagnostics**  
Various diagnostic functions allow you to identify critical situations in your database, such as missing tables and indexes. The diagnostics tools also include, for example, the audit log, the Catalog Browser, the Db2 message log, and the trace status.

## Availability of Functions in the DBA Cockpit

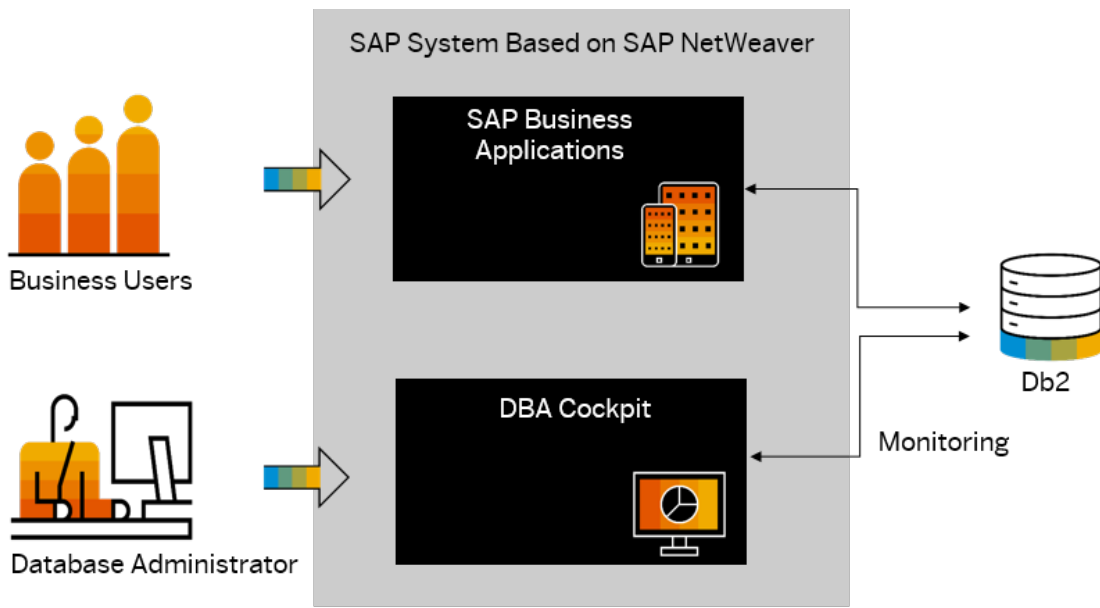
The availability of functions in the DBA Cockpit depends on various conditions, such as the following:

- The connections that are assigned to the system, such as the availability of RFC connections
- The user permissions, which are SAP-specific and also database-specific
- The configuration of the database, such as monitoring settings or automatic maintenance settings
- The configuration of the back end

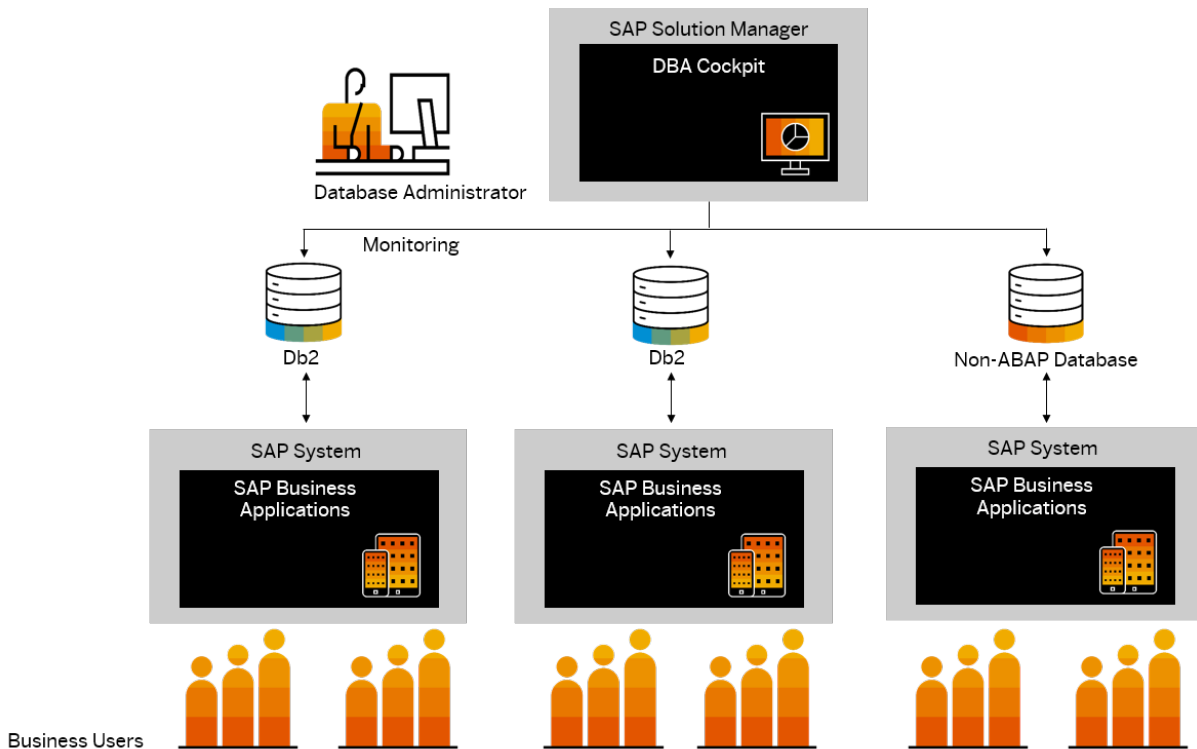
Depending on the conditions that the currently selected system fulfills, functions might either be disabled or not visible in the DBA Cockpit. Some screens might also show a reduced amount of information, and specific functions might be disabled (for example, administrative functions are unavailable if you do not have the required permissions).

### 6.1.1 The DBA Cockpit on a Local System and SAP Solution Manager

The DBA Cockpit is part of every SAP NetWeaver-based ABAP system. You can run the DBA Cockpit locally on an SAP NetWeaver-based system by calling the `DBACOCKPIT` transaction. Alternatively, you can run the DBA Cockpit on your SAP Solution Manager system, where you can access all databases in your system landscape using remote connections. Suppose you use the DBA Cockpit as part of the SAP Solution Manager system. In that case, this allows you to update and administrate all databases from a central system to each system separately. Using the DBA Cockpit centrally will enable you to administrate non-ABAP databases and ABAP-based systems with older versions of the DBA Cockpit.



DBA Cockpit on a Local SAP System



DBA Cockpit on an SAP Solution Manager System

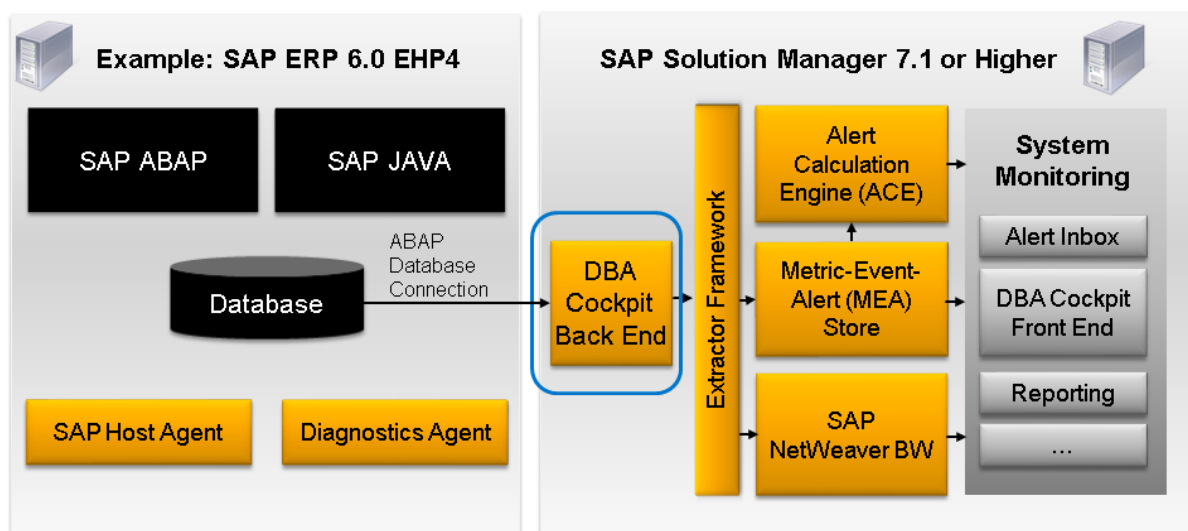
SAP Solution Manager is the free SAP toolset in your system landscape to monitor the full stack of an SAP system: from the operating system up to the business process. SAP Solution Manager is typically installed on a separate system. It provides centralized access to tools, methods, and preconfigured contents that you can use during your systems' evaluation, implementation, and operation. For database administrators, SAP Solution Manager offers a range of root cause analysis tools, alerting, and reporting.

The DBA Cockpit tools complement the available SAP Solution Manager tools, which provide high-level overviews of possible database issues, including reporting and alerting functions. In addition to the high-level overviews of SAP Solution Manager, the DBA Cockpit serves as an in-depth analysis tool for database-related issues.



Available Database Monitoring in SAP Solution Manager and the DBA Cockpit

The DBA Cockpit not only provides more tools for the expert database administrator, but it also plays an essential part in the SAP Solution Manager infrastructure. The remote database monitoring infrastructure of the DBA Cockpit is used by SAP Solution Manager to extract metrics from remote databases. These metrics are then passed on to the different applications in the SAP Solution Manager.



The DBA Cockpit as Metrics Provider for SAP Solution Manager

## 6.1.2 Entry Screen of the DBA Cockpit

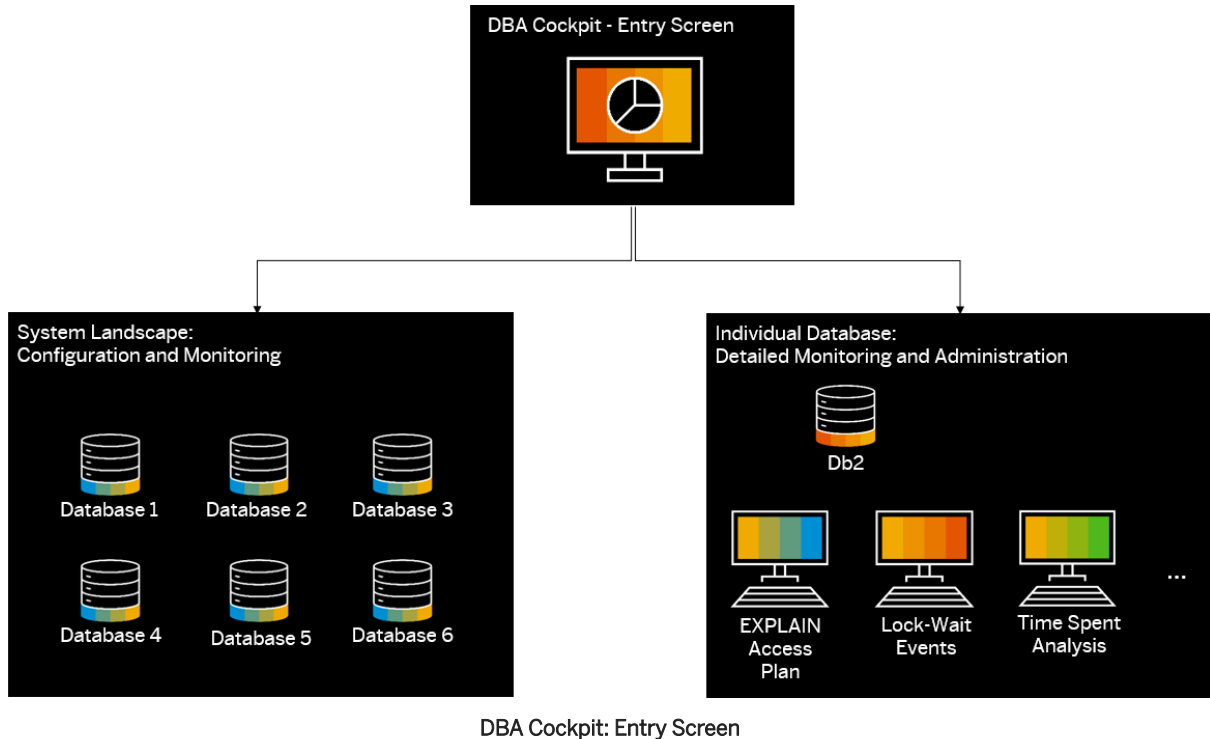
In the DBA Cockpit, you can monitor the databases of your entire system landscape. The DBA Cockpit provides the following functions:

- Overview screens that allow you to check the status of your system landscape at one glance
- Database-specific screens that will enable you to identify and analyze issues of individual databases in detail

When you start the DBA Cockpit, the system displays the *System Configuration* screen. On this screen, you can decide to proceed as follows:

- Configure and monitor your entire system landscape. To do this, you stay on the entry screen of the DBA Cockpit (this screen is the *System Landscape* tab page in the Web browser-based user interface).

- Monitor and administer individual databases in detail in your system landscape. To do this, you choose the *Database <Your Database>* tab page (in the Web browser-based user interface) or the *System...* pushbutton (in the SAP GUI-based user interface).



## 6.1.3 Screen Layout and Navigation in the DBA Cockpit

### Use

The DBA Cockpit works with the following user interfaces:

- SAP GUI
- A Web browser-based user interface (using the Web Dynpro UI technology)

By default, transaction `DBACOCKPIT` starts SAP GUI. You can switch to the Web browser-based user interface from SAP GUI or set the Web browser-based UI as default.

#### **Note**

The functions of the DBA Cockpit are available for both the Web browser-based and the SAP GUI-based user interface unless stated otherwise at the beginning of a section. If there are differences between the Web browser-based user interface and the SAP GUI, this is indicated in the documentation. Most areas in this documentation refer to the Web browser-based user interface. The relevant procedures describe how you navigate the Web browser-based UI, which is slightly different from the SAP GUI navigation.

## More Information

[SAP GUI-Based User Interface \[page 86\]](#)

### 6.1.3.1 SAP GUI-Based User Interface

The screens of the DBA Cockpit using the SAP GUI are divided into the following areas:

#### Application Toolbar

For example, it provides basic functions, displaying or hiding the areas on the left side using the *Full Screen On/Off* pushbutton.

#### System Landscape Selector

Let you choose the system to monitor. Additional information about a distributed database system is displayed if available.

#### Navigation Frame

Displays the main task areas of database administration in a tree structure, for example, performance monitoring, space management, and job scheduling. Within each task area, there is a set of related screens that you can choose from, and that correspond to the screens of the task areas on the *Database* tab page of the Web Dynpro user interface.

Also, *System Landscape* task area provides access to all system landscape-relevant actions such as system configuration, database connection, or self-monitoring. This task area corresponds to the *System Landscape* tab page on the Web Dynpro UI.

#### System Data

This area is common to most actions. For example, it provides the last refresh, the database startup time, and the name of the database server.

## Errors and Warnings from Self-Monitoring

If there are any issues with the infrastructure of the DBA Cockpit itself, the relevant message is displayed.

### Action Area

Displays the details of the currently selected action.

### Action Message Window

It only appears with specific actions and displays additional information that is related to the selected action.

### Framework Message Window

Displays the message window provided by the DBA Cockpit. It contains a complete history of all messages sent during the session.

Also, you can:

- Clean up the window by choosing the *Clear Message Window* pushbutton.
- Collapse or expand the window by choosing either the *Minimize Message Window* or the *Show Message Window* pushbutton.
- Check if a long text for a message is available by double-clicking the message or by choosing *Show Long Text*.
- Display previously shown messages by choosing the *Show History* pushbutton.
- Print the message text by choosing the *Print Version* pushbutton.

#### Note

Changes to the screen area sizes are user-specific and are restored when you next start the DBA Cockpit.

## 6.2 System Landscape

### 6.2.1 Db2 for z/OS Landscape Tools

The *Db2 for z/OS Landscape Tools* sections is intended to monitor/maintain all SAP systems in the customer's landscape from a single point – at a glance!

The following actions are now available:

## Landscape Configuration

*Landscape Configuration* shows the landscape actions and available systems. Here you can specify which landscape action should affect which system by de-/selecting the respective system's checkboxes.

## Component Check

*Component Check* shows an overview of the platform- relevant Db2 and application components installed for the respective systems

### Note

To display information on kernel components of remote systems, you must maintain an RFC connection.

## Parameter Check

*Parameter Check* shows an overview of the number and maximum deviations of the Installation parameters. That means, after having checked the parameter values against the respective SAP Notes, the number of red, yellow, and green traffic lights are listed. Also, it shows the highest alert in each category ("required", "highly recommended", "recommended", and "custom"). To look at the details (for example, which parameters are not set correctly in the respective system), click the link in the overview table. Thus you can navigate to the corresponding DBA Cockpit view and system: Installation Parameters Check (7.00 and above) / the Installation Parameters tab *Status* (7.02 and above)

## 6.2.2 Adding a Remote SAP System Database Connection (optional)

### 6.2.2.1 Step 1: Adding a Remote Database Connection

#### Procedure

1. Call transaction DBACOCKPIT.
2. In the system landscape toolbar, choose the *DB Connections* icon.
3. Choose the *Add* button.  
The screen *DB Connections: Add Connection Entry* appears.



- Enter the name of the connection in the *Connection Name* field.

**Note**

This name is a unique ID that you can choose freely, except for names reserved by SAP for generated connections. These can be, for example, administrator connections or connections that are used by systems from the system landscape directory (SLD).

- Specify the database connection attributes as follows:
  - In the *Database System* field, choose the name of the database platform.
  - In the *Connection Maximum* field, enter an appropriate value. This value limits the number of database connections that are concurrently held by the SAP system. The SAP system does not let you exceed this limit.
  - In the *Connection Optimum* field, enter an appropriate value. This value is more flexible and can be exceeded.
  - If you want the connection to be mandatory for the SAP system, select the *Permanent Connection* indicator. This parameter defines the availability of the database connection. When this parameter is selected, the database connection is regarded as being the local default connection. If this database connection is not available for a work process, the work process of the SAP system does not run.

**Caution**

You should set this parameter only if this connection is required to run your SAP system.

- In the *User Name* field, enter the name of the connected user. Make sure that you choose a user with the appropriate authorizations.
  - In the *Password* field, enter a password for the connected user.
- In the *Connection Parameters* table, specify the following additional database-specific attributes:

Connection Parameters

Attribute	Description
<i>Collection ID</i>	Identifier of the collection that is used for your database connection
<i>Database Host</i>	Name of your Db2 database host.
<i>Database Name</i>	Name of your Db2 system.
<i>Port Number</i>	This is the port number of your database. Contact your database administrator for details.
<i>Schema Name</i>	Name of the database schema. If you are using an MCODE system, the database schema also indicates the database system in which you are currently working.
<i>SSL Client Label</i>	SSLClientLabel where SSLClientLabel is the label for the SSLClientKeystore as configured in db2dsdriver.cfg of the local SAP System.

Attribute	Description
<a href="#">Authentication</a>	CERTIFICATE for TLS Certificate Authentication; otherwise, blank.
<a href="#">SSL Security</a>	Use SSL if your TLS connection should be encrypted, otherwise blank.

7. To confirm your entries, choose [Save](#).
8. Now you must catalog your database.

You can only set up and use [SSL Client Label](#), [Authentication](#), or [SSL Security](#) fields for DB2 subsystems, which are enabled for TLS Certificate Authentication as described in the [TLS Certificate Authentication for SAP with Db2 for z/OS](#) documentation.

For TLS Certificate Authentication, you have to set `Authentication=CERTIFICATE` and `SSL Client Label=<SSLClientLabel>` where `SSLClientLabel` is the label for the `SSLClientKeystore.db` as configured in `db2dsdriver.cfg` of the local SAP System. For more information see [TLS Certificate Authentication for SAP with Db2 for z/OS](#) documentation).

Before defining the connection, you have to copy the `SSLClientKeystore.db` and `SSLClientKeystash` files from the application server of the remote SAP/DB2 system to the application server of your local SAP system and then edit and configure your local `db2dsdriver.cfg` file.

If your TLS connection should be encrypted, you have to set `SSL Security=SSL`.

With that configuration, you don't have to define a DSN Alias for this remote connection in your `db2dsdriver.cfg` file. Still, you need this file for defining the TLS Certificate Authentication parameters `SSLClientKeystore.db` and `SSLClientKeystash`.

For more information see the [TLS Certificate Authentication for SAP with Db2 for z/OS](#) documentation and SAP Note [3231515](#).

## Result

As soon as the connection has been specified, the DBA Cockpit connects automatically to the newly added database system and displays data on the [System Data](#) tab page.

## How to handle connection errors like error 16 at the connection test?

Get the work process number from transaction `sm21` (system log), where you will detect the error at the end. Then search the corresponding more detailed SQL error message in the dev-trace file for this work process in transaction `sm50`.

## 6.2.2.2 Step 2: Adding a Remote SAP System

### Procedure

1. Call transaction `DBACOCKPIT`.
2. In the system landscape toolbar, choose [System Configuration](#)

#### ⓘ Note

For every remote system activated as described above, the DBA Cockpit Framework schedules some automatic tasks and maintains tables in the remote schema. The number and frequency of these tasks vary depending on the type of system (for example, ABAP or Java stack, etc.) and the checkboxes setting in the [Collector Option](#) section of the system configuration.

Suppose you temporarily do not want the remote Db2 subsystem to be accessed, such as maintenance, new installation, and system copy. In that case, you can disable these activities by switching the status (column [Configuration Status](#) on the system configuration panel) to [Inactive](#).

3. Select [Add](#)
4. Fill in the required system name
5. Choose [Database Connection](#) and select a previously configured database (as described in [Step 1: Adding a Remote Database Connection \[page 88\]](#))

### Result

As soon as the connection has been specified, the DBA Cockpit connects automatically to the newly added database system and displays data on the [System Data](#) tab page.

#### ⓘ Note

For every remote system activated as described above, the DBA Cockpit Framework schedules some automatic tasks and maintains tables in the remote schema. The number and frequency of these tasks vary depending on the type of system (for example, ABAP or Java stack, etc.) and the checkboxes setting in the system configuration's [Collector Options](#) section. If you temporarily do not want the remote Db2 subsystem to be accessed (for example, in case of maintenance, new installation, system copy), you can disable these activities by switching the status (column [Configuration Status](#) on the system configuration panel) to "INACTIVE".

## 6.3 Performance

The Db2 performance indicators are made available at different levels of reporting by the stored procedure `SAPCL`:

- Subsystem-wide scope: This includes performance data for the Db2 resources that are shared by all of the 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 many performance indicators. This data presents a prime source of performance monitoring and tuning indicators.
- Data set scope: Data set monitoring allows us to detect I/O performance problems of individual data sets.

Moreover, the Database Performance Monitor, otherwise known as the *DBA Cockpit*, includes 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 perfect vehicle for monitoring Db2 in SAP system environments, it lacks some of the stand-alone Db2 monitor features. 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.

SAPCL connects to the Db2 subsystem, for which you should gather the performance data. Some of the collected data is only available if Db2 monitor trace class 1 is active. For that reason, SAPCL 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 \[page 218\]](#). As soon as a request for performance data comes in, SAPCL calls Db2 via the IFI interface. The collected data is examined and written into regular or temporary tables.

Then the calling transaction reads these data from the corresponding tables. The tables' content is automatically deleted to keep the regular SAP monitor tables from growing excessively; Regular tables mainly hold database alerts. SAPCL inserts performance indicators into temporary tables to satisfy synchronous requests for performance data.

For more information about the Db2 trace classes and the IFI interface, see IBM Documentation *Db2 for z/OS Administration Guide*.

#### Note

The *DBA Cockpit* also supports the monitoring of remote Db2 subsystems.

## Accessing the DBA Cockpit

To access the DBA Cockpit, call transaction DBACOCKPIT.

## Monitoring a Remote Db2 Subsystem

To monitor a remote Db2 subsystem, select a system in the main screen of transaction DBACOCKPIT by choosing *System* <your systems name>.

## Adding a New Connection

To add a new connection, choose [Add](#) on the main screen and specify the new system.

### 6.3.1 Performance Warehouse

You can analyze the performance data of your database system using the Performance Warehouse.

To access the Performance Warehouse, call transaction `DBACOCKPIT`, and choose [► Performance](#)  
[► Performance Warehouse](#) [▶](#).

The following areas are available in the Performance Warehouse:

- [Reporting \[page 94\]](#)  
This content is displayed by default.
- [Configuration \[page 95\]](#)

If you are using the SAP GUI-based user interface, the application starts in a different Web browser.

## Prerequisites

To use the Performance Warehouse, you must have an SAP Solution Manager system with Solution Manager Diagnostics (SMD) enabled in your system.

To configure the extraction of data from the SAP BI system into the Solution Manager Diagnostics BI (SMD BI), you use the SMD Setup Wizard.

## Features

In the Performance Warehouse, some of the performance indicators collected by the DBA Cockpit are stored in an SAP Business Intelligence (BI) system. This SAP BI system is used by the Solution Manager Diagnostics (SMD) back-end of an SAP Solution Manager system. SMD uses SAP BI to store workload data of SAP applications.

Based on this architecture, the DBA Cockpit uses SAP BI technology to provide reports for performance analysis, which you can customize according to your needs. All collected data has a time dimension to analyze the database performance for any point in time or over a specified time frame.

Reports are displayed as a chart to visualize key performance indicators (KPIs). Also, for all reports, there is a detailed table view. To navigate within these reports, you use the SAP BI drill-down feature.

By default, the Performance Warehouse is delivered with predefined content that you can use to create your reports according to your needs.

You can access many of the benefits the Performance Warehouse offers and see more details about the local system via the Technical Monitoring Cockpit via the Solution Manager or in the connected system itself, which is available starting with Release 7.40. For more information, see <https://help.sap.com/viewer/tmc>.

## 6.3.1.1 Reporting

You use the *Reporting* screen data to analyze database performance problems in the present or the past.

To access the Reporting screen of the Performance Warehouse, call transaction `DBACOCKPIT`, and choose **► Performance ► Performance Warehouse ► Reporting ▾**.

### Note

If you are using the SAP GUI, a separate Web browser opens for this application.

## Specifying the Time Frame

To display detailed reports, you first have to specify the time frame for which you want to analyze data by defining the following:

- *Granularity*  
You can choose between *Minute*, *Hour*, *Day*, or *Month*. Depending on your selection, the values for your time frame might change.

### Note

Data is currently collected only on a daily or monthly basis.

- *Time Frame*  
If you choose *Custom Selection* from the drop-down list, you can manually enter your analysis's starting and ending time.  
To activate your custom selection, choose *Apply Filter*. For any other selection from the drop-down list, the reports are automatically refreshed.

The reports are categorized. There is one tab page for each category that you create. You find a button row on every tab page for the reports. Every pushbutton in the button row represents a specific view of database performance.

### Reset Report

By choosing *Reset Report*, you only reset the characteristics that you set in *Detail: Navigation*. The time frame is not reset.

## Displaying a Report

To display a report, choose the appropriate view pushbutton on the respective tab page.

The reports consist of up to two sections:

- In the upper section, a chart is displayed to visualize the key performance indicators (KPIs). The chart provides a subset of the key columns from the detail table view.

#### Note

The chart display is optional and not available for all available views.

- In the lower section, a detailed table view is available.

You can drill down your reports using the context menu of a column header in the *Detail: <Category – View>* screen area or by specifying the respective value using the pushbuttons in the *Detail: Navigation* screen area. In *Detail: Navigation*, you can also add and remove columns or key figures, or you can set filters on columns.

You can also create your exceptions (*Chart: Exceptions* or *Details: Exceptions*) for almost all reports on key performance indicators.

## 6.3.1.2 Configuration

You set all parameters that are related to the Performance Warehouse on the *Configuration* screen.

For example, you can configure the framework, the templates used for the reports, and the report categories.

The DBA Cockpit uses BI Business Explorer (BEx) Web templates to analyze the Solution Manager Diagnostics (SMD) BI's performance data. You can create your own BI BEx Web templates based on this data and integrate new BI BEx Web templates into the performance warehouse.

You can access the *Configuration* screen of the performance warehouse by calling transaction `DBACOCKPIT` and choosing **► Performance ► Performance Warehouse ► Configuration ►**.

#### Note

If you are using the SAP GUI, a separate Web browser opens for this application.

On the *Configuration* screen, the following tab pages are available:

- *Configuration*
- *Web Reports*
- *Report Categories*

## Configuration

You display and modify the performance warehouse's configuration parameters for the monitored system on this tab page.

To modify the parameters, choose *Edit*, *Save*, or *Cancel*.

Depending on your database platform, the displayed selection of values varies. The following parameters are displayed for all database platforms:

- [BI Server](#)
- [Managing DBA Cockpit](#)
- [Reporting Time Zone](#)

### Note

The *Default* checkbox is selected if your complete landscape's default value is the same as the one specified for your system.

## Web Reports

On this tab page, you configure the display on the *Reporting* screen.

You can view and modify the integrated BI BEx Web templates in the expandable tree table. To change the parameters, choose *Edit*, *Add* or *Delete*.

The main report categories appear, and for each report category, you can view or modify the views by expanding the appropriate report category. These views appear as pushbuttons on the respective category tab page on the *Reporting* screen. To change the sequence within a category, choose *Up* or *Down*.

To display details about a view, select it in the table.

The following parameters are displayed in the *Details for Web Reports* area below the table view:

Details for Web Reports: Parameter List

Parameter	Description
<i>Report</i>	Specifies the name of the report. This text appears on the <i>Reporting</i> screen on the view pushbutton on a category tab page.
<i>Description</i>	A detailed description of the report. This text appears on the <i>Reporting</i> as a tooltip for the key indicator's pushbutton on a category tab page.
<i>Category</i>	Specifies the report category. Each category is represented on the <i>Reporting</i> on a separate tab page.
<i>Web Report (Default)</i>	The technical name of the BI BEx Web templates
<i>Web Report (Day)</i>	The technical name of the BI BEx Web templates for granularity Day
<i>Web Report (Month)</i>	The technical name of the BI BEx Web templates for granularity Month
<i>Date Providers (Time)</i>	Specifies the data provider of the BI BEx Web templates with a time dimension. You can change the time dimension by choosing <i>Granularity</i> on the <i>Reporting</i> screen.



Parameter	Description
<a href="#">Active</a>	If selected, the report is available for performance analysis.
<a href="#">Default</a>	If selected, the report is executed as soon as the tab page is selected.
<a href="#">Database</a>	Name of database platform
<a href="#">Release (min)</a>	Minimum database release for this report
<a href="#">Release (max)</a>	Maximum database release for this report

## Report Categories

You display and modify the categories for BI BEx Web templates of the reports shown on the [Reporting](#) screen on this tab page.

To modify the parameters, choose [Edit](#), [Add](#) or [Delete](#). To change the sequence of the categories on the [Reporting](#) screen, choose [Up](#) or [Down](#).

## 6.3.2 Storage

### 6.3.2.1 Virtual and Real Storage

To monitor DBM1 virtual storage usage, proceed as follows:

1. Call transaction `DBACOCKPIT`
2. Choose [► Performance ► Db2 Subsystem Activity ►](#)
3. In the drop-down list box, choose [DBM1 virtual storage](#)

The information provided on this screen allows you to determine the primary 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 periodic, not synchronously, determined in the Db2 statistics interval controlled by the Db2 system parameter `STATIME`.

You can change the settings for history collection as follows:

1. Call transaction `DBACOCKPIT`
2. Choose [► Configuration ► SAP Collector Profile ►](#)

You can set the parameters in the DBM1 Storage History Panel:

- The data collection interval
- The retention time

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.

Improved navigation and display:

In newer releases / Support Packages, the DBM1 virtual storage display has been moved to a new [Storage](#) folder and renamed to [Virtual and Real Storage](#). Db2 10 real storage counters have been made available, and with Db2 11, we provided some more information concerning IRLM pool statistics and log manager frames.

Choose [Performance](#) > [Storage](#) > [Virtual and Real Storage](#) >

#### Note

The DBM1 virtual storage entry is removed from the [Db2 Subsystem Activity](#) drop-down as soon as it is available as a DBA Cockpit navigation tree entry in the storage folder.

### 6.3.2.2 Workfiles

The workfile database is used as storage for Db2® work files for processing SQL statements that require working space (such as the space that is necessary for a sort) and as storage for created global temporary tables and declared global temporary tables.

The following information is provided on the [Workfiles](#) monitor:

- Number and size of in-memory work files Temporary storage used (4K and 32 K)
- Overflow events, alert threshold, and DGTT information (up to Db2 11)

### 6.3.2.3 Environmental Descriptor Manager (EDM) Pool

The EDM Pool workspace provides an overview of the Environmental Descriptor Manager (EDM) pool activity connected with Db2. You can view this information on the EDM Pool screen:

- EDM statistics lists total counts, counts during the last sample period, and the “per second” count for various tasks associated with the EDM pool.
- [DBD Pool](#) lists information about database descriptors (DBDs).
- [SKEL Pool](#) utilization shows information about the EDM skeleton pool and lists information about the package search.
- [Statement Pool](#) lists information about the statements in the global cache and EDM pool.

For more information, see the IBM documentation [Calculating EDM pool sizes](#).

### 6.3.3 SQL

The SQL Cache screen displays information about SQL statements that are executed very often and stored in your system's SQL cache. This information helps you identify those SQL statements that consume

many resources. You can also determine whether fine-tuning those statements is necessary to improve the performance of the database.

### 6.3.3.1 Dynamic Statement Cache

This function provides monitoring capability for the prepared SQL statements in the dynamic statement cache. The information about the cached statements is gathered during their preparation and execution.

To access this function, proceed as follows:

1. Call transaction `DBACOCKPIT`
2. Choose [► Performance](#) [► SQL](#) [► Dynamic Statement Cache](#) [►](#)

### 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, this trace is automatically turned on at the SAP system startup. Subsequently, stopping and restarting IFCID 318 will clear any previously collected statistics for existing statements in the cache. Suppose a Db2 system executes dynamic SQL statements from the dynamic SQL statement cache while the IFCID 318 is off. In that case, 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 statistics fields for filtering. Db2 will then return information about the statements with the highest values in the selected field.

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.

### Db2 Data Sharing Environment

If Db2 data sharing is employed, you can access cached statement statistics for any Db2 member of the data sharing group by selecting that member from the dropdown list. Select **ALL** from the dropdown list box for a consolidated view of the SQL statements in the dynamic statements cache or all members in the data sharing group.

## 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 the section [EXPLAIN \[page 229\]](#).

## Correlation of SQL Statement and ABAP Program Code

Every SQL statement 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 one to navigate directly from an SQL statement to the ABAP source code that has initiated the statement's dynamic preparation.

The SQL statement tag includes the STATUID value for OLAP queries in SAP BI systems. This allows you to inspect the BI statistics of OLAP queries.

For MCOB landscapes, it is recommended to exploit the navigation feature only for the statements belonging to the SAP system in which you are working.

## Display Historical Statements

It is possible to view Db2 statement cache statistics for prior points in time.

### Background

The `SAPCL` alert router regularly retrieves the statement cache statistics from Db2. It persistently stores the 50 statements with the largest accumulated elapsed time in the interval since the last time historical statement cache statistics have been retrieved. This interval is controlled by the Db2 statistics interval ZPARAM STATIME.

### 6.3.3.2 Static Statement Cache

This function provides monitoring capability for the prepared SQL statements in the static statement cache. The information about the cached statements is gathered during their preparation and execution.

To access this function, proceed as follows:

1. Call transaction `DBACOCKPIT`
2. Choose [► Performance](#) [► SQL](#) [► Static Statement Cache](#) [►](#)

## Trace IFCID 0401

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 401 is turned on.

Therefore, this trace is automatically turned on at the SAP system startup. Subsequently, stopping and restarting IFCID 401 will clear any previously collected statistics for existing statements in the cache. Suppose a Db2 system executes static SQL statements from the static SQL statement cache while the IFCID 401 is off. In that case, Db2 will return a record for any qualifying statement in the cache, but most of the statistics fields for those records will show all zeros.

## Filtering

You can filter the content of the static statement cache by:

- Package Name
- Collection ID

And you can select whether you want to see only explainable `stmts` or all `stmts`, including the not explainable ones.

Additionally, you can select any of the following statistics fields for filtering.

- Number of executions
- Number of synchronous reads
- Number of getpages
- Number of rows examined
- Number of rows processed
- Number of sorts
- Number of index scans
- Number of tablespace scans
- Number of parallel groups
- Number of synchronous writes
- Number of RID list failures (DB2 limits)
- Number of RID list failures (no storage)

After providing a threshold value, Db2 will return information about all statements that exceed the given value for the selected statistics field.

You can see how many rows were fetched from the DB after filtering the results in the filter bar.

## Db2 Data Sharing Environment

If Db2 data sharing is employed, you can access cached statement statistics for any Db2 member of the data sharing group by selecting that member from the dropdown list. Select **ALL** from the dropdown list box for a

consolidated view of the SQL statements in the static statements cache or all members in the data sharing group.

## SQL Statement Details

You can select one entry of the static statement cache and click on the [Details](#) button in the top region of the screen or double-click on the row in the ALV table to navigate to the details screen. The details screen provides further information about the selected statement. The details screen consists of four tabs:

- Statement
- Identification
- Elapsed Wait Time
- Statistics

In the Statement Tab, you can see the complete SQL statement. The Identification tab provides general information about the statement, like the host language or when it was initially bound. This information can help to identify the statement. The elapsed wait time tab shows how long the DB was blocked on which task (like CPU operation or I/O task) per execution and accumulated over every execution. The statistics tab provides different values like how many rows were processed per execution or how many executions were done per second.

## SQL Statement Text and EXPLAIN Function

The records displayed provide identifying information, statistics, and part of the SQL statement text.

You can use the explain button to see all the details of the access path for the SQL statement in the same way as in the dynamic statement cache.

### ⚠ Caution

The "Explain" Button is visible **only** for explainable statements (they have the attribute `explainable = true`).

## 6.3.4 Db2 Subsystem Summary

For information about the overall Db2 database performance, call transaction `DBACOCKPIT`, and choose [► Performance ► Db2 Subsystem Summary ►](#).

The *Db2 Subsystem Summary* screen provides a complete cross-system overview of the database systems being monitored. The information is divided into five sections :

- Db2 Resources
- Hardware Resources

- Global Times
- Alerts
- Top 5 SQL Statements

### Note

The data is displayed in the following forms:

- High Water Mark values: The maximum value the counter has reached since you started the Db2 subsystem the last time, or the data collection has not yet been started. You could observe this for counters:
  - HWM f. aux. storage in use 64 Bit
  - HWM f. real storage in use 64 Bit
- -1 Value: This value appears on the screen if the data is not available for the requested period of measurement
  - The Global Times counters and Avg. Disk I/O Response Time is supported by Db2 Version 10 and higher.
  - Avg. CP+zIIP Utilization LPAR is supported with SAPCL 720 Patch Level 16 or higher.

## 6.3.5 Db2 Subsystem Activity

### Note

This system overview has become obsolete.

The structure of the existing Db2 Subsystem Activity overview screen has become very complex over time. Informational counters are just successively listed on the screen without providing any information about the counter's category or where problems occur. Moreover, some of the counters are meaningful only for DB experts, others for DB administrators.

For more information see, SAP Note [1896293](#)

For information about the overall Db2 database performance, call transaction `DBACOCKPIT`, and choose

► [Performance](#) ► [Db2 Subsystem Activity](#) ►.

The overview screen, shown by default, 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 group member. For more information, see *Data Sharing Environment* in this section.

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**

The maximum value the counter has reached since the time you started the Db2 subsystem

You can display values for three measurement periods:

- Since Db2 start
- Since reset

When you call transaction `DBACOCKPIT` and choose **► Performance ► Db2 Subsystem Activity ►**, the system displays the data since the Db2 subsystem start by default.

#### 📘 Note

The type of period measured you select the same for all subscreens of the [Db2 Subsystem Activity](#) screen.

## Data Sharing Environment

If Db2 data sharing is employed, you can access the information for any Db2 member by selecting that member from the dropdown list box [Mem.](#):

## Standard Functions

The following standard functions are available:

Standard Db2 Subsystem Activity Functions

Function	Description
<a href="#">Reset</a>	Sets the reset time to the current time. All accumulated values are set to zero.
<a href="#">Since reset</a>	The period measured is the duration between the previously determined reset time and the current time.
<a href="#">Since DB start</a>	The period measured is the duration between the time the subsystem was started and the current time.

## Detail Screens for Db2 Subsystem Activity

For information on specific areas, you can choose, for example, [Bufferpool activity](#) or [SQL activity](#) from the dropdown list box.

The detail screens have the same functions as the [Overview](#) screen.

From the [Db2 for z/OS: Subsystem. Act./Buffer Pool List](#) screen, you can access detailed information on a specific buffer pool by selecting the buffer pool from the list and choosing [Details](#).



## 6.3.5.1 Lock Waits

A lock wait is a situation where a thread waits to use a resource locked by another thread.

You can view all lock waits within your Db2 subsystem as follows:

1. Call transaction `DBACOCKPIT`
2. Choose ► *Performance* ► *Db2 Subsystem Activity* ▾
3. Choose *Lock waits* in the dropdown list box.

The *Lock waits* screen displays a list of all waits. The list includes the following:

- Locked resource
- The process currently using this resource
- The process of waiting to use this resource

## 6.3.6 Thread Activity

To display the list of active threads connected to the Db2 subsystem you are monitoring, proceed as follows:

1. Call transaction `DBACOCKPIT`
2. Choose ► *Performance* ► *Thread Activity* ▾

The list of active threads (threads currently performing a transaction) is refreshed when you call this screen.

### Note

If there is no transaction running within a thread at 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 accounting traces 1, 2, and 3.

## 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 the SAP system ID, SAP system number, the application server's hostname, the work process number, and the database connection name.

Among the information 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 alert router identifies itself under the name *Alert router*.

You can select any thread listed and use one of the following functions:

#### Thread Activity Functions

Function	Result
<a href="#">Buffer pool</a>	<p>Displays all buffer pools used by the selected thread in the measurement period</p> <p>For information on <code>read</code> and <code>write</code> activity, both to and from a specific buffer pool, select the buffer pool and choose <a href="#">Details</a>.</p>
<a href="#">Locking</a>	<p>Displays locking activity information on the selected thread</p> <p>Locking problems can be further investigated by choosing <a href="#">Locked resources</a> or <a href="#">lock waits</a>, to examine resources locked by or waited for by the thread monitored.</p>
<a href="#">Lock waits</a>	<p>Displays a list of resources that the selected thread is waiting for.</p>
<a href="#">Locked resources</a>	<p>Displays a list of the resources locked by the selected thread.</p>
<a href="#">DS Locking</a>	<p><b>Data sharing only:</b> The system displays data sharing locking.</p>
<a href="#">SQL activity</a>	<p>Displays the number of times SQL statements are executed when processing a Db2 application. The sections <a href="#">Total DML</a> (Data Manipulation Language) and <a href="#">Total DCL</a> (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 application object type.</p>
<a href="#">SQL statement</a>	<p>Displays information about the current SQL statement being executed by the monitored thread. If you need to see all the access path details for that statement, choose <a href="#">Explain statement</a>.</p>
<a href="#">Group BP</a>	<p><b>Data sharing only.</b> The system displays all active group buffer pools.</p>
<a href="#">Times</a>	<p>Displays various times for processing in Db2 and out of Db2 and suspension times</p> <p>Displays the response times of various actions performed by the currently executing transaction of the selected thread</p> <p>The times displayed can help guide your investigation into application performance and tuning. You must start accounting trace class 2 for this information. If you use accounting trace class 3, the values are more detailed.</p>

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.

## Db2 Data Sharing Environment

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.

You can choose *ALL* from the dropdown list box for a consolidated view of all members' Db2 threads in the data sharing group.

### 6.3.7 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 obtain the average times of the different kinds of suspension in milliseconds.

To display this screen, proceed as follows:

1. Call transaction `DBACOCKPIT`
2. Choose [► Performance ► Global Times ►](#)

The global times can be presented in two ways: as a **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 calculated as averages of the snapshots of active threads collected in the Db2 statistics interval. The timers remain in that table for about three days before they are automatically deleted.

To display global times, the accounting trace classes 1, 2, and 3 must be active. They are automatically started during the SAP system startup. You can start these trace classes manually by issuing the following command:

```
START TRACE (ACCTG) CLASS(1,2,3)
```

This command is among the predefined commands of transaction [► DBACOCKPIT ► Diagnostics ► Db2 Commands ►](#).

#### Note

All the values are global times, 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.

## 6.3.8 Data Sets Statistics

Using ► [Performance](#) ► [Data Sets Statistics](#) ] on the main screen of transaction `DBACOCKPIT`, you can analyze the data sets' performance data belonging to the Db2 page sets. Performance data for the last completed statistics interval, 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 at least one I/O operation per second. Additional filters are available to restrict further the number 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 \[page 218\]](#).

## 6.3.9 Installation Parameters

To display the installation parameters of the Db2 subsystem, call transaction `DBACOCKPIT`, and choose ► [Performance](#) ► [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 \[page 24\]](#), 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 ([Protect/data def](#))
- [Logging](#)
- Application programming defaults ([Appl. prog](#))
- [DDF](#)
- [Miscellaneous](#)

If Db2 data sharing is employed, it is also possible to view the other members of the data sharing group's installation parameters.

For optimal operations and performance, SAP systems require (for all releases) specific Db2 system parameter settings:

- Required settings  
These settings are necessary for an SAP system to function correctly. It would help if you did not change them.

- Highly recommended settings  
Although they do not influence SAP system functions, these settings are essential for ensuring optimal performance and should not be changed
- 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.

## 7.02 and higher

To monitor db2 subsystem parameters, a new DBA Cockpit navigation tree entry, [Installation Parameters](#), has been introduced. It provides a customizable check, a searchable view containing all zParms, a custom view, and an automatic update of the recommendations via SAP Note that can be triggered manually. The parameters on the Status Tab are sorted by severity.

Information provided is received via stored procedure ADMIN\_INFO\_SYSPARM.

In contrast to the old installation parameter display, which was unclear and parameters were spread across several screens without any possibility to search for a specific parameter name, the new one provides several features to improve usability.

The new architecture is very dynamic. For example, if SAP changes or introduces the recommendation for a specific parameter, the respective SAP Note is updated, and the recommendations are immediately available. Thus the check on the status tab is always current.

All information concerning Parameters / Parameter Check can now be found in the DBACOCKPIT when you choose [► Performance ► Installation Parameters ►](#).

### Note

The Db2 Parameter Check (DB16) has been removed from the DBA Cockpit [► Performance ► Db2 Parameter Check \(DB16\) ►](#), as well as the Maintain Check Parameters (DB17) entry [► Diagnostics ► Maintain Check Parameters \(DB17\) ►](#)

## 7.00-7.11

To monitor the subsystem parameters' status, a new DBA Cockpit navigation tree entry [Installation Parameters Check](#), has been introduced. It lists all installation parameters for which an SAP recommendation exists and displays its status. The parameters are sorted by severity. Information provided is received via stored procedure ADMIN\_INFO\_SYSPARM.

### Note

The Db2 Parameter Check (DB16) has been removed from the DBA Cockpit [► Performance ► Db2 Parameter Check \(DB16\) ►](#)

## 6.4 Space

You can use the monitor (transaction DB02) to determine the size of different tables, tablespaces, and indexes and display their past growth. Using the transaction, you can view important statistics about the database.

The tables and indexes monitor offers the following features:

- Support for remote non ABAP systems  
Up to now, the data that on displayed in the monitor has been stored in the monitoring database `MONI`. Since the cluster table, `MONI` is not remotely accessible, and it has been replaced with database tables `DB2QDB02_SCHEMA` and `DB2QDB02_OBJECT`, which can be called via SQL.
- Usability improvements (filter and navigation functions)
- Object and partition level view
- History for all objects
- Better LOB association
- Introduction of Key Performance Indicators (KPIs)

The KPIs allow you to check the system's database growth and display warnings concerning database limits if thresholds are exceeded.

### 6.4.1 Key Performance Indicators

The Key Performance Indicators (KPIs) allow you to check the database growth. Warnings concerning the limits of database objects are displayed on the [SAP System Overview](#) screen (see chapter *Space*).

#### Procedure

To display the KPIs, proceed as follows:

1. Call transaction `DBACOCKPIT`.
2. Choose **► Space ► KPIs ► ... ►**
3. Enter the selection criteria for the selected KPI.

KPI:	Description:
Tablespace Limits	<p>The last column (<i>% Used</i>) indicates the percentage of space that has been required per tablespace partition.</p> <p>The following values are used for the limit calculation:</p> <ul style="list-style-type: none"> <li>• <b>Partitioned TS:</b> DSSIZE* <ul style="list-style-type: none"> <li>• With LARGE option: 4GB/ partition</li> <li>• LOB TS: DSSIZE</li> </ul> </li> <li>• <b>Non-partitioned TS:</b> <ul style="list-style-type: none"> <li>• If DSSIZE &lt;&gt; 0: DSSIZE</li> <li>• If DSSIZE = 0: 64 GB</li> </ul> </li> </ul> <p>*DSSIZE reports the maximum size of a dataset.</p>
Index Limits	<p>The last column (<i>% Used</i>) indicates the percentage of space that has been required per index partition.</p> <p>The following values are used for the limit calculation:</p> <p><b>Partitioned/Partitioning Index:</b> DSSIZE</p> <p><b>Nonpartitioned Index:</b></p> <ul style="list-style-type: none"> <li>• Partitioned TS and PIECESIZE &lt;&gt; 0: PIECESIZE</li> <li>• Partitioned TS and PIECESIZE = 0: 64GB</li> <li>• Partitioned TS: PIECESIZE</li> </ul>
Partition Limits	<p>The last column (<i>% Used</i>) indicates the percentage of the number of available partitions that have been required.</p> <p>The following values are used for the calculation:</p> <p><b>Partitioned by growth:</b> MAXPARTITIONS</p> <p><b>Range-partitioning:</b> a combination of PGSIZE and DSSIZE</p>
Largest Tables	Column <i>Allocated Space</i> is the sort criterion.
Largest Indexes	
Largest Multi-Table Tablespaces	
Largest Free Space	Column <i>Freespace</i> using KB as a unit. The last column ( <i>% Free</i> ) indicates the percentage of space that has not been used.
Fastest Growing Tables	Column <i>Growth</i> : Average growth over daily history (in KB) for the configured time range

## 6.4.2 Unencrypted Objects

The unencrypted objects screen displays the objects that haven't yet been encrypted when DB2 encryption is activated. This screen enables you to find datasets created by manual DBA activities outside the SAP world or moved out of the SAP environment. For example, using backup operations can no longer be managed by the SAP system and may not be encrypted together with the rest of the DB objects.

### Selection Criteria

The *Selection Criteria* screen area provides a filter function in the left area. In the right pane, the results can be sorted, and the number of unencrypted objects matched by the set filter is displayed.

You can filter by the following criteria:

- *Schema*: Shows the schema of the database currently being examined and can't be changed in any way.
- *HLQ*: A High-Level Qualifier (HLQ) is the most basic filter. The dropdown lists all DB2 systems that are connected to the app server, both directly and remotely.
- *Filter DS*: In this user-defined text field, you can select the DB2 elements up to any number, whereby the "\*" is used as a wildcard.
- *Sort by fetches*

### Result Table

Below the *Selection Criteria* is an ALV table that provides the usual NW table functions (such as the sorting of the columns, export, and so on).

Columns:

- *Dataset Name*: Displays the full name of the dataset  
Important: The entries in the column begin with the HLQ. If the result is further restricted in the "Filter DS" text field, you can't directly transfer the entry in this column into the text field, but the HLQ must be deducted.
- *Encrypted*: Shows the Encryption Status. Possible values:
  - N - Unknown State
  - U - Not Encrypted
- *Spacetype*: Type of usage
  - TS: Contains table contents (tablespaces)
  - IS: Contains indexes (index space)
- *Database*: Database name – also included in the *Dataset Name* column
- *Spacename*: Name of the Space - Also contained in the *Dataset Name* column  
Possible action: Click on Space name to open the screen with the details above the space, depending on the type of area (*TableSpace* or *IndexSpace*).
- *Partition*: Indicates whether space is partitioned.
  - 1: Partitioned space



- 0: Unpartitioned space
- *Partitions*: The number of partitions (only possible if it's a partitioned space).
- *Creator*: Username of Object Creator

## Details

To display further details, you can click on an entry in the *Spacename* column to navigate a detailed view. For more information about the relevant detail screens, see the relevant section in the DBA Guide.

### 6.4.3 Tables and Indexes Monitor

You can use the monitor (transaction DB02) to determine the size of different tables, tablespaces, and indexes and display their past growth. Using the transaction, you can view important statistics about the database.

The tables and indexes monitor offers the following new features:

- Support for remote non ABAP systems  
Up to now, the data that is displayed in the monitor has been stored on the monitoring database `MONI`. Since the cluster table, `MONI` is not remotely accessible. It has been replaced with database tables `DB2QDB02_SCHEMA` and `DB2QDB02_OBJECT`, which can be called via SQL.
- Usability improvements (filter and navigation functions)
- Object and partition level view
- History for all objects
- Better LOB association
- Introduction of Key Performance Indicators (KPIs)

The KPIs allow you to check the system's database growth and display warnings concerning database limits if thresholds are exceeded.



### 6.4.4 Space

The *Space Overview* monitor provides information about the allocation of space in the database. The initial screen contains essential information about:

- The entire database
- Individual database objects like tables, indexes, and tablespaces

In addition to this information, the monitor offers a set of functions that help you to analyze and check various properties of the database, tables, or indexes in greater detail.

## Prerequisites

- Configure the collector profile in the DBA Cockpit.  
Call transaction `DBACOCKPIT` and choose [► Configuration ► SAP Collector Profile](#) .  
Choose your settings under [► Space History](#) .  
For more information, see *SAP Collector Profile*.
- The database alert router does the data collection for space monitoring. The database alert router does the data collection for space monitoring. Make sure you have installed SAPCL Release 7.20 on the DB subsystem.


### Note

As of Release 7.02, 7.30, and higher, it is no longer necessary to schedule a batch report.

- To migrate history data from table `MONI`. Execute report `RSDB2_DB02_MONI_CONV` in the ABAP Editor (transaction `SE38`). Select [Preview](#) to get an overview of the data which will be migrated.

## Procedure

To display the database space statistics, proceed as follows:

1. Call transaction `DBACOCKPIT`.
2. Choose [► Space ► SAP System Overview](#) 

The *SAP System Overview* screen is divided into the panels *Overview*, *Status*, and a tabstrip with *Details*, *DB2 Subsystem View*, and *History*. The date and time when you generated the information shown at the top of the screen.

Since its introduction, the DB02 System Overview only shows technical DB2 limits and object sizes, but not about the storage pools' physical utilization.

A tab called *Disk Space* has been added to the SAP System Overview. The information displayed is received via stored procedure `ADMIN_INFO_SMS (DB2 10)`, enabling the SMS Data integration in the Space Monitor. This new tab provides information about the capacity, free space, number of volumes, and largest free extend of the data and log copy pools, including a history. The possibility to filter for specific Copy Pools, Stogroups, and Volumes to get the current snapshot of the respective object has been introduced. A new alert traffic light showing the physical data copy pool's status has been added to the status section at the top of the screen.

Screen Area:	Information:
Overview	Allocation of space in the database
Status	Traffic lights indicate the status of tablespace, index, and partition limits.  For more information, see <i>Key Performance Indicators</i> .
Details	Overview information for tables, tablespaces, and indexes

Screen Area:	Information:
History	History of database space information. You can choose between a daily, weekly, or monthly view of the data.
DB2 Subsystem	It provides information about schema, many tables, and the system type.
Disk Space	History for the DSN\$DDFD6M0\$DB and DSN\$DDFD6M0\$LGCopypool;  Current snapshot with a filter for specific Copy Pools, Stogroups, and Volumes

## 6.4.5 Tables

The [Tables](#) monitor gives you information about the allocation of space for specified tables. In addition to the information presented, the monitor helps you analyze tables in greater detail (history, structure, indexes, partition details, and LOBs).

### Procedure

To display the tablespace statistics, proceed as follows:

1. Call transaction DBACOCKPIT.
2. Choose .
3. Enter the name of the database object and select the schema. When you specify the table's name, you can also enter \* as a wildcard for any character string (T0\* for all table names that start with T0, for example).
4. Choose the criterion by which the list will be sorted ([Sort by](#)).
5. Specify the maximum number of displayed rows ([# Hits](#)).
6. Decide if you want to have an object-level view or a partition-level view (partitions are displayed collapsed or expanded).
7. If necessary, select the required object in the list provided for any additional analysis.

[Table Info](#) displays the chosen table's attributes, the creator's name, the tablespace, further information on pages and rows, the creation date, and the date and time of the last change.

For a detailed analysis, use the following tabs:

Tab:	Description:
Tablespace Details	Detailed information about the size of the space the table resides in. Also, the tablespace's various properties are shown, such as the time of creation, the last change, etc.

Tab:	Description:
Tablespace History	History of the tablespace on a daily, weekly, or monthly basis  The database can trace the growth of a tablespace over a particular period. Changes to the size and free space are recorded.
Structure	Fields of a table and the database specifics of each field
Indexes	All indexes that exist in the database for a table. The release dependency marker is listed in the <code>IBMREQD</code> column.
Partitions / Partition Details	Mark the required partition in the list of partitions for any additional analysis.

## 6.4.6 Indexes

The *Indexes* monitor gives you information about the allocation of space for specified indexes. In addition to the information presented, the monitor helps you analyze indexes in greater detail (history, structure, partition details, and datasets).

### Procedure

To display the index space statistics, proceed as follows:

1. Call transaction DBACOCKPIT.
2. Choose ► *Space* ► *Indexes* ►.
3. Enter the name of the index and select the schema. When you specify the index's name, you can also enter \* as a wildcard for any character string (TO\* for all index names that start with TO, for example).
4. Choose the criterion by which the list will be sorted (*Sort by*).
5. Specify the maximum number of displayed rows (*# Hits*).
6. Decide if you want to have an object-level view or a partition-level view (partitions are displayed collapsed or expanded).
7. If necessary, select the required object in the list provided for any additional analysis. *Index Info* displays the chosen index attributes: the name of the creator, the database, table name, index space, the type of the index (primary or secondary index), and other attributes.

For a detailed analysis, use the following tabs:

Tab:	Description:
Details	Detailed information about the size of an index, its allocated space, and row information. Also, various properties of the index are shown, such as the time of creation, the last change, etc.
History	History of an index on a daily, weekly or monthly basis  The database can trace the growth of an index over a particular period. Changes to the size and free space are recorded.
Structure	Fields of an index and the database specifics of each field
Partitions / Partition Details	Mark the required partition in the list of partitions for any additional analysis.
Datasets	The list of datasets for this index  You can allocate indexes that are not partitioned in one dataset. If an index is partitioned, each partition has its dataset.

## 6.4.7 Tablespaces

The *Tablespaces* monitor gives you information about the allocation of space for specified tablespaces. In addition to the information presented, the monitor helps you analyze tablespaces in greater detail (history, structure, tables, partition, partition details, and data sets).

### Procedure

To display the tablespace space statistics, proceed as follows:

1. Call transaction DBACOCKPIT.
2. Choose **► Space ► Tablespaces ►**.
3. Enter the name of the tablespace and select the schema. When you specify the index's name, you can also enter \* as a wildcard for any character string (TO\* for all index names that start with TO, for example).
4. Choose the criterion by which the list will be sorted (*Sort by*).
5. Specify the maximum number of displayed rows (*# Hits*).
6. Decide if you want to have an object-level view or a partition-level view (partitions are displayed collapsed or expanded).
7. If necessary, select the required object in the list provided for any additional analysis. *TS Info* displays the chosen tablespace attributes: the creator, the database, table name, index space, the type of the tablespace, and other attributes.

For a detailed analysis, use the following tabs:

Tab:	Description:
Details	Detailed information about the size of a tablespace and its allocated space  Also, the tablespace's various properties are shown, such as the time of creation, the last change, etc.
History	History of a tablespace on a daily, weekly, or monthly basis  The database can trace the growth of a tablespace over a particular period. Changes to the size and free space are recorded.
Tables	Tables of a tablespace and the table creator of each table
Partitions / Partition Details	Mark the required partition in the list of partitions for any additional analysis.
Data sets	The list of datasets for this tablespace.

## 6.5 Backup and Recovery

### 6.5.1 Backup

The DBA Planning Calendar gives you the following backup options:

- Complete backup (full image copy) or an incremental backup (incremental image copy) of all SAP tablespaces
- Backup of SAP tablespaces according to the recommendations of stored procedure `DSNACCOX`

### CHANGELIMIT

The incremental backup uses a `CHANGELIMIT` threshold that switches to full image copy if this threshold of changes has been reached. While a backup job is running, other programs have read-write access to the relevant tablespace (`SHRLEVEL CHANGE`).

#### ⚠ Caution

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 different 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. Ensure that this job is only executed after the other COPY jobs have run; otherwise, locking conflicts could arise.

## Template Utility

For the allocation of data sets, the TEMPLATE utility is used, which has the advantage of being able to size the data sets based upon catalog statistic information automatically. Please ensure that the catalog statistics are reasonably up to date to prevent Out of Space conditions.

## Profile Parameters for Backup

The following profile parameters exist for backup. You maintain these as follows:

1. Call transaction `DBACOCKPIT`
2. Choose **► Configuration ► JOB Profile ►**
3. Choose the relevant tab, as listed below.

### Storage Parameters (Tab Storage)

- *HLQ for Backup data sets*: `MGMTCLAS (SMS)`, `STORCLAS (SMS)`, `DATACLAS (SMS)`
- *Volume count*
- The *Number of GDG generations*:

GDG is the abbreviation for the “generation data group”. This is the administrative unit for collecting historically related non-VSAM data sets 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 user data. A GDG can consist only of a limited number of generations. This number is specified at the 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

<HLQ for Backup data sets>.<database>.<tablespace>.<CCC>

If you use GDGs, otherwise, the data sets have the following naming convention:

<HLQ for Backup data sets>.<database>.<tablespace>.<CCC><nnnnnn>

Where 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 the remote copy.

The third character is P for primary or B for a 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 six months (26 weeks), you should specify  $3 * 26 = 78$  for the *Number of GDG generations*:

### COPY-specific parameters (Tab COPY)

- *Number of parallel jobs*

- *CHANGELIMIT*

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 12 for z/OS Utility Guide and Reference*

- *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 importantly, speeds up processes that access data from the tables mentioned above, such as `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". 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, which has the same value as the `AGE` parameter of `MODIFY RECOVERY`.

- A record in `SYSIBM.SYSCOPY` without the backup data set is useless.
- The `RECOVER` utility can not use a backup data set without a record in `SYSIBM.SYSCOPY`. You need to take a series of complicated steps to use it still (for example, `DSN1COPY` or `LOAD` and ancient archive logs). This should be reserved for emergencies and will not be your usual mode of operation.

Here are some recommendations:

- If you do not use GDGs or 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 the *Number of GDG generations*: In the example above, you specified **78** for the *Number of GDG generations*;, sufficient to keep six months worth of backups. Correspondingly, you should set 183 days for *AGE for MODIFY RECOVERY*.

#### ⓘ Note

The backup jobs use data from the table and index monitor to size the backup data sets correctly if the statistics data in the catalog or Real-Time Statistics is not available.

### Number of Objects to be Copied (Tab COPY)

There is a threefold radio button group allowing you to choose between the following options:

- *Backup all objects in the Db2 subsystem*  
This option will choose all tablespaces in catalog table `SYSDIBM.SYSTABLESPACE` that are physically allocated (i.e., `SYSDIBM.SYSTABLEPART(SPACE) <> -1`) and index spaces that are copiable. If you have any SAP unrelated data in the Db2 subsystem, it will also be copied.
- *Backup ABAP & Java systems and Db2 catalog*  
All copiable database objects that belong to all ABAP systems and the Db2 catalog are copied. All the table creators with the name `J2EE_CONFIG` are determined and used to generate a list of copiable database objects belonging to Java AS systems. All members of this list are also copied.
- *Backup ABAP systems and Db2 catalog*  
This is the old habit (which is also the default) that will copy all copiable database objects that belong to all ABAP systems and the Db2 catalog.

In any case, take care that the user who will execute the COPY utility is authorized to run the COPY utility on all objects that will be copied.

### Backup on Partition Level (Tab COPY)

Partitioned tablespaces are backed up on the partition level by default. To deactivate this function, deselect the *Enable backup on partition level* indicator in this dialog.

## 6.5.2 Recovery

If required, you can recover damaged SAP tablespaces or indexes to the current status. If it is a partitioned table/index space, you have the opportunity to specify a single partition that should recover. Recovery to a prior point in time should generally not be performed for individual tablespaces for consistency reasons. SAP Support needs to be contacted first to undo changes that may have been caused by a faulty transport, for example.

## 6.5.3 Backup Monitor for Image Copies

The Backup Monitor provides information on object-based backups.

### Procedure

1. Call transaction DBACOCKPIT.
2. Choose ► [Backup and Recovery](#) ► [Backup Monitor](#) ►.

The initial screen shows a list of all tablespaces with the last `BACKUP SYSTEM` or image copy. The information is provided by the catalog table `SYSIBM.SYSCOPY`.

The following filter options are available:

Filter Option:	Description:
Database and Spacename	Filter the objects by database and spacename.
Sort By	Sort the objects by different criteria.
View	Display the objects with <a href="#">Objectlevel</a> or <a href="#">Partitionlevel</a> view.
Object Type	Choose <a href="#">Tablespaces</a> or <a href="#">Indexspaces</a> .
Backup Status	Display all objects, all backups, or all objects without a backup.
<code>BACKUP SYSTEM</code>	Choose this option to mix in your <code>BACKUP SYSTEM</code> backups.

Click on a space name to get the complete backup history and the `REPORT RECOVERY` output of an object.

The tab [Recommended](#) provides a list of all objects for which a backup is recommended. These objects are determined once a day by stored procedure `DSNACCOR`.

For more information, see *CCMS Monitor Set* in *Monitoring and Performance* and the IBM documentation *DB2 for z/OS SQL Reference*.

## 6.6 Configuration

### 6.6.1 Failover Configuration

The [Failover Configuration](#) tool will help transition from the legacy `connect.ini` based failover (SAP Failover) to the new CLI failover `db2dsdriver.cfg` mechanism.

## Features

To launch the *Failover Configuration* tool, call transaction Db2, and choose ► *Configuration* ► *Failover Configuration* . The tool can read the ABAP stack's failover configuration in the legacy connect.ini and the CLI failover db2dsdriver.cfg format.

In the *toolbar* area, the following features are available:

- Load the current active failover configuration from the SAP application server
- Read selected failover configuration from the disk of the SAP app server
- Import a failover configuration from the frontend (SAPGUI upload)
- Export a failover configuration to the frontend (SAPGUI download)
- For saving the changed failover configuration, use the *Save* button.

The following *general information* tabs are available:

- The *System Type* of the currently monitored system is shown
- Editable *DDF Location Name* of connected Db2 database
- The *Alias* (only relevant for db2dsdriver.cfg files)
- The *Host* (always the first entry from the DB2 Server list)
- Selection of *Input Format*, to read other than the default configuration
- Selection of *Output Format*, relevant for export and writing

### Note

If more than one connection is defined in the db2dsdriver.cfg file, an attempt is made to read the section that belongs to the local system instead of the first section in the file.

In the *configuration* area, the components of a connect.in file are displayed and modified. This area is divided into the following tabs:

- *Input*
- *Db2-Servers*
- *Affinity Lists*
- *App-Servers*
- *Parameters*
- *Check*
- *Result*

### Input Tab and Result Tab

The *Input* tab displays the original contents of the connect.ini to be converted. This tab is not editable, and its content is not changed by the tool.

If the output format is set to db2dsdriver.cfg, the conversion can already be displayed in the *Result* tab. This tab is automatically updated with every change to the configuration.

### Db2-Servers Tab

With this tab, you can add, update, and delete a data sharing member. The first member in the panel is always the default connection. The logical names of the DB2 servers can be renamed using the Renumber button (server1... serverN). These names are used in the affinity list to determine the order of the failover.

Data sharing members are equivalent to connections in the legacy failover mechanism. The member names (SSID) are not relevant for converting to `db2dsdriver.cfg` and do not appear in the generated `db2dsdriver.cfg`.

### Note

The Db2 CLI driver requires fully qualified domain hostnames with the port number. There is no check for correct names and ports of the DB2 servers! Therefore make sure always to specify fully qualified names like `ihsapjc.wdf.sap.corp`. For more information about the `db2dsdriver.cfg` and Alert Router, see SAP Note [2068561](#).

## Affinity Lists Tab

This tab allows the user to define, remove, and edit affinity lists. You can only delete an affinity list if it is not assigned to an application server. Each Db2 member can be present in multiple affinity lists, and a Db2 member can appear only once in an affinity list. Its order uniquely defines each affinity list.

An affinity list contains at least one data-sharing member and, at maximum, all defined data sharing members.

A double click on an *Affinity Lists* name in the left box selects the list's content and appears in the centered list named *DB2-Server Order*. The remaining data sharing members are listed on the right in the list labeled *Alternative Db2 Members*. Members can be assigned, removed, and modify the order in an Affinity Lists. You can use the arrow buttons to add or remove DB2 servers from the current *DB2-Server Order* list and change their place.

## Application Servers Tab

Each application server is defined by its hostname and an assigned affinity lists. The name of the application server is retrieved from the configuration read in. The currently assigned affinity lists are previewed upon selecting the application with a double-click in a separate list box to the right. You can choose the affinity list from a drop-down list in the appropriate column. If a failover configuration with unknown application servers is loaded, a warning message appears.

### Note

The Db2 CLI driver configuration file `db2dsdriver.cfg` has no such concept as a “default connection” like the `[DEFAULT_GROUP]` in `connect.ini`.

Therefore you need to specify an explicit affinity list for each application server.

## Parameter Tab

A specific set of parameters is required to use the new CLI failover mechanism. Those five parameters are listed with their SAP recommended values in the following table:

Parameter	Value
<code>acrRetryInterval</code>	32000
<code>affinityFailbackInterval</code>	300
<code>enableAcr</code>	true
<code>enableSeamlessAcr</code>	true

Parameter	Value
<code>maxAcrRetries</code>	5

You can adjust the values for the parameters `acrRetryInterval`, `affinityFailbackInterval`, and `maxAcrRetries`. You can select the section where the displayed parameters are to be stored in the [Parameter](#) section list box. The “common” area is ignored for the conversion to `db2dsdriver.cfg`. The “acr” section is automatically selected if the output format is set to `db2dsdriver.cfg`.

The standard parameters required for the cli failover can be entered automatically using the “fill in” button.

### Check Tab

The results of the check are displayed in this tab. It is mainly used to check the relationships between DB2 and App servers using the affinity lists. It also checks the parameters required for CLI failover.

### Result Tab

Depending on the output format set, either the generated XML for the `db2dsdriver.cfg` file or the generated `connect.ini` is displayed. You can make no changes in this tab.

## 6.6.2 Setup of Database Performance Monitoring with Stored Procedure SAPCL

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 `DBACOCKPIT`.

This monitor uses the stored procedure `SAPCL` to gather Db2 performance data from the Instrumentation Facility Interface (IFI). In previous releases, `rfcoscol` provided this function.

`SAPCL` is accompanied by an alert router process that is controlled by the stored procedure. `SAPCL` can also be used to monitor remote Db2 subsystems.

### Note

`sapocol` uses the CCMS agent `sapccmsrt` to communicate with SAP application servers. Therefore, `rfcoscol` has completely been replaced and is not needed anymore.

Concerning performance monitoring, `SAPCL` serves the following purposes:

- Provides current snapshot information of the overall system
- Allows you to submit Db2 commands
- The alert router process catches database alerts, which are raised by Db2, and propagates them to the SAP system

To ensure that these features, function properly, it is essential that `SAPCL` is correctly installed, and its alert router process is always running. This is essential for proper performance monitoring. You can do this from within the SAP system in transaction `Db2`.

## 6.6.2.1 Authorization Profiles

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

Authorization Profiles

Authorization Profile	Authorizations Permitted
S_DB2_DBADM (Authorization profile of the Db2 database administrator)	With this authorization, you can: <ul style="list-style-type: none"><li>• Execute an <code>ALTER</code> on the secondary quantity of a tablespace or index (tables and indexes monitor)</li><li>• Change and delete the JCL Jobs of any user (see <a href="#">JCL Job Submission [page 143]</a>)</li><li>• Change the TSO password of any user (see <a href="#">JCL Job Submission [page 143]</a>)</li><li>• Execute all Db2 commands, create, change and delete new commands (SAP performance monitor)</li><li>• Execute <code>SELECT</code> on Db2 catalog tables (Db2 catalog browser)</li><li>• Switch an application server to a different Db2 data sharing member</li></ul>
S_DB2_COMM	With this authorization, you can: <ul style="list-style-type: none"><li>• Execute, change and delete all Db2 commands, create new commands (SAP performance monitor)</li></ul>
S_DB2_EXPC	With this authorization, you can: <ul style="list-style-type: none"><li>• Execute all Db2 commands, create, change and delete new commands (SAP performance monitor) that have the command user <code>ALLUSER</code></li></ul>
S_DB2_ALLU	With this authorization, you can: <ul style="list-style-type: none"><li>• Execute all Db2 commands that have the command user <code>ALLUSER</code> (SAP performance monitor)</li></ul>
S_DB2_DBUTIL	With this authorization, you can: <ul style="list-style-type: none"><li>• <code>READ</code> the TSO password of the user entered in DB2J for setting up a multiconnection to call Db2 utilities.</li></ul>

You call transaction SU01 to give the appropriate authorization profile to a user.

## 6.6.2.2 Permissions, Privileges, and Authorizations for SAPCL

This section describes the privileges necessary to install and run SAPCL.

## User ID to Install SAPCL

The user ID used to install SAPCL needs the following permission bits for the files `sapdb2c1` and `DBRM.db2c1db` in UNIX System Services:

- EXECUTE and READ permission for file `sapdb2c1`
- READ permission for file `DBRM.db2c1db`

## DB Connect User ID

The DB Connect user ID requires the MONITOR1 and MONITOR2 privilege to collect trace data.

The TRACE privilege to start the monitor trace needs to be granted to this user ID as well.

## SAPCL Schema

The stored procedure SAPCL runs in the WLM application environment specified on the configuration panel of SAPCL. The stored procedure must not be program-controlled.

The schema of the stored procedure, which is the SAP system's schema in the local case, needs to be authorized to create the stored procedure in this WLM application environment. If it is generally prevented from creating stored procedures in this WLM environment, the schema's RACF group needs to be explicitly authorized.

The following example shows the RACF command that authorizes the RACF group of schema `SAPR3` to create stored procedures in the WLM application environment `D8T0CCA`:

### ❖ Example

```
PERMIT D8T0.WLMENV.D8T0CCA CLASS(DSNR) ID(SAPR3) ACCESS(READ)
```

## User ID in transaction DB2J

Moreover, to bind the packages of `SAPCL`, the user ID specified in transaction `DB2J` must have `BINDADD` system privilege in `Db2`.

## 6.6.2.3 Installing Stored Procedure SAPCL

### Use

The executable of the stored procedure `SAPCL` and its database request module need to be placed in a UNIX System Services directory on z/OS.

These files are:

- `sapdb2c1`, which is the executable of the stored procedure
- `DBRM.db2c1db`, which is the database request module.

### Procedure

#### Installing SAPCL executables

- Create a directory `<sapcl_dir>` in z/OS UNIX System Services
- Download the newest version of `SAPDB2CL.SAR`. You can find `SAPDB2CL.SAR` at: <http://support.sap.com/swdc>.

## 6.6.2.4 Considerations for Remote Systems

### Use

Transaction `DBACOCKPIT` supports configuring `SAPCL` for remote Db2 subsystems from an SAP system that runs on Db2 for z/OS. To perform these administrative activities, proceed as follows:

### Procedure

#### Configuring SAPCL for Remote Db2 Subsystems

To configure `SAPCL` for remote Db2 subsystems from an SAP system, proceed as follows:

1. Make sure that you have defined a connection to the remote Db2 subsystem. For more information, see sections [Step 1: Adding a Remote Database Connection \[page 88\]](#) and [Step 2: Adding a Remote SAP System \[page 91\]](#).

#### Note

Carefully choose the schema name that you specify for the remote connection.

- The schema specified in the connection **must not** be identical to the schema used for an SAP system version 6.40 or earlier in the remote Db2 subsystem. Suppose there is an SAP NetWeaver 7.0 system in



the remote Db2 subsystem. In that case, you can reuse the stored procedure SAPCL that may already exist in the remote Db2 subsystem for remotely monitoring this subsystem. If you intend to do this, you need to specify the schema of this SAP system.

- If SAPCL is created under a new schema, the tables required by SAPCL are created implicitly in the remote Db2 subsystem.
2. Maintain the JCL settings for the remote system. To do this, perform the following steps:
    1. Call transaction Db2 and select the remote Db2 subsystem from the *System* drop-down menu.

#### Note

You can only select a connection if you can establish a connection to the remote system.

2. Choose ► *Jobs* ► *Jobs Control* ►.
3. Maintain the JCL job head and the JCL profile parameters by choosing *Job head* and *Profiles*, respectively.
3. When performing the steps in the section [Installing Stored Procedure SAPCL \[page 128\]](#), make sure you copy the files `sapdb2c1` and `DBRM.db2c1db` to a UNIX System Services directory on the LPAR of the remote Db2 subsystem (or for each remote Db2 member of a data sharing group). You can configure this directory in transaction DB2 by choosing ► *Checks/settings* ► *SAP Collector settings* ►.
4. When performing the steps in the section [Creating and Configuring Stored Procedure SAPCL \[page 129\]](#), make sure you have selected the remote system as follows:
  1. Call transaction Db2.
  2. In the *System* drop-down menu, select the remote Db2 subsystem.
  3. Choose ► *Checks/settings* ► *SAP Collector Settings* ►
  4. Then perform the box *SAP Collector Administration* steps as described in the item section [Creating and Configuring Stored Procedure SAPCL \[page 129\]](#).  
If you only have one remote data sharing group, you need to perform the steps only once. **Exception:** you need to start an alert router per Db2 member. If you want to do this, you have to establish a connection to each member described in [Step 1: Adding a Remote Database Connection \[page 88\]](#).

## 6.6.2.5 Creating and Configuring Stored Procedure SAPCL

We recommend creating a separate directory `pro <SID>`, which would include `SAPCL` executable files and a sub-directory for `SAPCL` traces. Such a setup would make it easier to maintain different versions of `SAPCL pro <SID>` and, if required, easily switch between `SAPCL` binaries for a given `<SID>` without affecting other systems.

When the executable and the database request module of `SAPCL` are available in the HFS on z/OS, the next step is to create the stored procedure in Db2 and configure it. You can perform the necessary tasks in transaction `DBACOCKPIT`.

### Creating a Stored Procedure

To create a stored procedure, you call transaction `DBACOCKPIT` and choose

## Setting the Configuration Parameters

You can set the configuration parameters per system. Every Db2 member of the data sharing group and the enabled remote database connections are available in the *System* drop-down list.

To enable a remote database for remote monitoring, choose *Add remote database*. You can select from the remote database connections for Db2 for z/OS database configured in table DBCON (transaction DBCO).

The required configuration parameters per system are as follows:

Parameter	Description
<i>Plan name</i>	Plan name of SAPCL and its alert router  Default: CSAPCL
<i>Collection ID</i>	Collection ID of SAPCL and its alert router  Default: CSAPCL
<i>HFS path of collector</i>	The HFS path of the directory into which the executable of SAPCL and its database request module reside.  Default: /u/sap3/<SAPSID>run
<i>HFS path of trace directory</i>	The HFS path of the directory into which SAPCL and the alert router write trace data.  Default: /u/sap3/<SAPSID>run/trace
<i>WLM application environment</i>	The WLM application environment for the stored procedure SAPCL  This must not be a program-controlled environment. Also, it is recommended that the parameter NUMTCB of the application environment be larger than 1. It is recommended that you use the standard WLM application environment <DB2SSID>_GENERAL, created by Db2 and used for many Db2-supplied stored procedures for SAPCL.  Note, however, that activating new versions of a stored procedure requires that the stored procedure's application environment has to be refreshed. This means that all stored procedures in the application environment are unavailable for a brief period.

Parameter	Description
<a href="#">z/OS system library</a>	<p>The high-level qualifier for the z/OS system libraries (SCEEBND2, CSSLIB, SCEECPP, SCEELKED and SCEELKEX)</p> <p>These libraries are required for the job that link-edits the SAPCL module to the Db2 load library.</p> <p>Default: SYS1</p>
<a href="#">Db2 load library for link-edit</a>	<p>This parameter specifies the Db2 load library into which the SAPCL module is copied.</p> <p>Note that the SAPCL module needs to be APF-authorized. Moreover, the specified load library needs to be in the concatenation of the selected WLM application environment.</p> <p>Default: &lt;DB2SSID&gt; .V&lt;DB2_REL&gt;10 .SDSNLOAD</p>
<a href="#">Db2 load library (only for remote databases)</a>	<p>The Db2 load library of the remote database</p> <p>Default: &lt;DB2SSID&gt; .V&lt;DB2_REL&gt;10 .SDSNLOAD</p>

When the parameters are correct and saved, perform the following steps to install and prepare the stored procedure `SAPCL` in Db2. A prerequisite for this is that a `RACF` user ID and its password and the general JCL job submission profile parameters are maintained in the *JOB Control* screen (call transaction `DBACOCKPIT` and choose [► Configuration ► JOB Control ►](#)).

All of the following steps are performed under the user ID that is maintained in the *JOB Profile* screen.

For more information about the authorities that are required to install and use `SAPCL`, see [Permissions, Privileges and Authorizations for SAPCL \[page 126\]](#).

## Prerequisites for Installing, and Preparing the Stored Procedure

The `RACF` user ID along with its password and the general JCL job submission profile parameters are maintained in the *JOB Profile* and *JOB Control* screens (call transaction `DBACOCKPIT` and choose [► Configuration ►](#)).

All of the following steps are performed under the user ID maintained in the *JOB Profile* screen.

### 6.6.2.5.1 Step 1: Creating SAPCL in the Db2 Catalog

#### Use

This section describes how to create the stored procedure in the Db2 catalog.

## Prerequisites

For this step, the configuration parameters `Plan name`, `Collection ID`, and `WLM application environment` will be inserted into the Db2 system routines table.

If `SAPCL` is created in a local Db2 subsystem, its schema is the SAP system's schema.

If `SAPCL` is created in a remote Db2 subsystem, the schema maintained for the remote database connection to the Db2 subsystem is used.

## Procedure

1. Choose *Create stored procedure*.
2. If the user that is specified in transaction `DB2J` does not have the privilege to access the WLM application environment, the following message is shown in the SDSF log in z/OS:

### Sample Code

```
ICH408I USER(R3USER ) GROUP(DE#03303) NAME(USER F. DRDA ) 837
D8N0.WLMENV.D8N0CCA CL(DSNR ) INSUFFICIENT ACCESS AUTHORITY
FROM * (G) ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )
```

To solve this problem, grant the user ID read privileges to the WLM application environment (in the example `D8N0.WLMENV.D8N0CCA`).

## 6.6.2.5.2 Step 2: Link-Edit `sapdb2cl`

### Use

The next step is to link-edit the executable `sapdb2cl` to a Db2 load library in the concatenation of the stored procedure's WLM application environment.

This step links the stored procedure definition in the Db2 catalog with the executable.

## Prerequisites

The following configuration parameters are used for this step:

- `z/OS System Library`
- `DB2 load library for link-edit`
- `DB2 load library (for remote databases)`

## 6.6.2.5.3 Step 3: Binding the Database Request Module

### Use

This section describes how to bind the database request module (DBRM) of `sapdb2c1`.

### Prerequisites

- For this step, the following configuration parameters are used:
  - `Plan name`
  - `Collection ID`
  - `DB2 load library` (for remote databases)
- This bind is necessary since the stored procedure `SAPCL` issues SQL statements against the selected Db2 subsystem. The database request module, which is released together with the `SAPCL` executable, must reside in the same directory as `sapdb2c1`. Otherwise, the execution of the generated bind job will fail.
- Be sure that `SAPCL` is not executing and that the alert routers using the plan are stopped. You can only rebind the plan if it is currently not in use.
- You have to resubmit the BIND job when the database request module has been changed. This is always the case when a new `SAPCL` level is installed.

### Procedure

1. To bind the database request module (DBRM) of `sapdb2c1`, choose `Bind DBRM`.
2. As soon as you change the plan name or the collection ID, you have to resubmit the new setting's BIND job.

## 6.6.2.5.4 Step 4: Refresh the WLM Application Environment

After link-editing new versions of `SAPCL`, it is necessary to refresh its WLM application environment.

There are two ways to refresh the WLM application environment:

- If you would like to link-edit by choosing [WLM refresh](#) on the panel, you need to permit access to the `WLM_REFRESH` RACF resource profile for the specified application environment.

### ❁ Example

The RACF `RDEFINE` command for Db2 subsystem D8T0, WLM application environment D8TOCCA, and the RACF group DEVL7083 is as follows:

### ⌘ Sample Code

```
RDEFINE DSNR (D8T0.WLM_REFRESH.D8TOCCA)
PE D8T0.WLM_REFRESH.D8TOCCA +
CLASS(DSNR) ID(DEVL7083) ACCESS(READ)
END
```

If you install the stored procedure `WLM_REFRESH` with the unmodified job `DSNTIJSG`, the DB Connect user ID must be granted this authority. If you modify `DSNTIJSG` and specify `SECURITY USER` for `WLM_REFRESH`, then the user ID specified in transaction `DB2J` for stored procedure execution needs to be granted this authority.

- Alternatively, you can issue the following z/OS command to refresh the application environment every time a new version of `SAPCL` is link-edited:

```
/V WLM,APPLENV=D8TOCCA,REFRESH
```

## 6.6.2.5.5 Step 5: Grant User Privileges

As the stored procedure is created using the user ID specified in transaction `DB2J`, it is necessary to provide usage privileges for the stored procedure `SAPCL`.

There are two methods you can use:

- Choose *Grant usage privileges*  
A `db2radm` is invoked that submits the necessary grants.
- Alternatively, you can also directly invoke `db2radm` at operating system level with the following options:

### ⌘ Sample Code

```
db2radm -m db2i -G only -Q <Schema> -O <Collection for SAPCL> -R <Plan for SAPCL> -u <ADMIN_USER> -p <ADMIN_PASSWORD>
```

For remote databases, use the following:

### ⌘ Sample Code

```
db2radm -m db2i -G only -H <Host> -P <Port> -L <Location> -S <SSID> -C <Collection for CLI Packages> -Q <Schema> -O <Collection for SAPCL> -R <Plan for SAPCL> -U <Connect User> -u <ADMIN_USER> -p <ADMIN_PASSWORD>
```

### Variables Used in the Example

- `<SSID>` is the SSID of the Db2 subsystem.
- `<SCHEMA>` stands for the schema/creator of the stored procedure `SAPCL`.
- `<ADMIN_USER>` is the user that grants the usage privilege of `SAPCL`. It is the owner of `SAPCL` or a user with `SYSADM` rights.

## 6.6.2.5.6 Step 6: Create Necessary Tables

Choose *Create tables* to make `SAPCL` create the tables that it requires to operate.

`SAPCL` and the alert router pass Db2 performance indicators and alerts to SAP via regular and temporary tables.

After this step, `SAPCL` is ready to be used by transaction `DBACOCKPIT` to provide Db2 performance indicators.

- If the `SAPCL` tables are created for a local Db2 subsystem, their schema has the SAP system's schema.
- If the tables are created in a remote Db2 subsystem, the schema maintained for the remote database connection to the Db2 subsystem is used. The schema of `SAPCL` and the schema of its tables are always identical.

## 6.6.2.5.7 Step 7: Loading the Alert Router

### Use

The alert router is a USS process that `SAPCL` (Db2 stored procedure) starts. `SAPCL` reads `sapdb2c1` from the table `DB2QALRTLOAD` and starts the alert router. You can do this as follows:

### Procedure

1. Choose *Load* in the *Alert Router* frame on the panel. This loads the alert router into table `DB2QALRTLOAD`.
2. Start the alert router by choosing *Start*.
3. To rebind the `SAPCL` and the alert router, which share the same plan and collection ID, you need to stop the alert router. Choose *Stop*.
4. To check the alert router's status, you can choose *Check Status* in the *Alert Router* frame.

### Result

If the automatically created jobs do not exactly match your requirements, you may change the job skeletons in the *JOB Control* screen in *JOB List* tab. The names of these jobs are as follows:

- `SAPCLLINK`
- `SAPCLBIND`

If you want to execute these jobs directly on z/OS, you can download them.

- Call transaction `DBACOCKPIT` ► *Jobs* ► *JOB Control* ►
- Choose *Jobs*
- Select the job that you wish to download in the job list

- [Edit](#) the file
- Choose [Save](#) to download the file.
- Choose [Back](#) without saving to leave the editor.

Since some requests to collect performance data are resource-intensive, these requests can be limited. The maximum number of concurrent users can be specified and adjusted to your needs.

## 6.6.2.6 SAP Collector Profile

[SAP Collector Profile](#) enables you to customize the behavior of Stored Procedure SAPCL and the alert router. SAPCL is used to gather Db2 performance data and provide it to the SAP system.

To access the SAP Collector Profile initial screen, call transaction `DBACOCKPIT`, and choose [► Configuration ► SAP Collector Profile ►](#).

You can configure the following areas:

Area:	Description:
Statement Cache History	<p>Configures how many statements are collected for the Statement Cache History. Other settings that can be changed here are <b>Collection Interval</b>, <b>Collection Offset</b>, <b>Retention Time</b>.</p> <p>You can display the Statement Cache History by choosing <a href="#">► DBA Cockpit ► Performance ► SQL ► Statement Cache History ►</a>.</p>
Global Times History	<p>Configures the interval of collected global times</p> <p>The collected data are computed into long term analysis values. You can display the Global Times History by choosing <a href="#">► DBA Cockpit ► Performance ► Global Times ► Longterm monitoring ►</a>.</p>
DBM1 Storage History	<p>Configures how many DBM1 virtual storage snapshots are taken and kept. You can display the space history by choosing <a href="#">► DBA Cockpit ► Performance ► Db2 Subsystem Activity ► DBM1 virtual storage ►</a>.</p>
Accelerator Statistics History	



Area:	Description:
Space History	<p>The space history collection is turned on by default.</p> <div data-bbox="801 421 1398 573" style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px;"> <p><b>Note</b></p> <p>It is unnecessary to start an updated job for transaction DB02 as of SAP NetWeaver 7.02, 7.30, and higher.</p> </div> <p>You can display the space history by choosing <a href="#">DBA Cockpit &gt; Space</a>.</p>
Space Alert Limits	<p><a href="#">DBA Cockpit &gt; Space &gt; SAP System Overview &gt; Status</a> shows information, limits for tablespaces, indexes, and partitions. You can configure the thresholds for the limits in percent for yellow and red limits. See also <a href="#">DBA Cockpit &gt; Space &gt; KPIs &gt; Tablespace Limits, Index Limits, and Partition Limits</a>. The tablespace limit indicates, for example, if the maximum size of a tablespace has been exceeded.</p>
Alert History	<p>Configures how many alerts are collected by the Alert Router. Alerts are lock escalations, long-running URs, deadlocks, timeouts, and active log shortage. You can display space history by choosing <a href="#">DBA Cockpit &gt; Alerts</a>.</p>
Alert Router Detection of Standby DB2 Data Sharing Members	
Behavior of Alert Router	
Db2 Traces Started by Alert Router	<p>Configures the Traces started on the DB by the Alert Router. Besides disabling and enabling all DB traces, it is possible to enable and disable: "Start Accounting Traces", "Start Trace for Statement cache", "Trace for static SQL", and "Trace to Stop Real Storagecontainers".</p> <div data-bbox="801 1532 1398 1715" style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px;"> <p><b>Note</b></p> <p>Enabling or disabling the checkboxes will not start or stop any trace but changes the configuration that traces the Alert Router will start on its next restart.</p> </div>

**Note**

You can migrate history data from table MONI. Execute report RSDB2\_DB02\_MONI\_CONV in the ABAP Editor (transaction SE38). Select *Preview* for an overview of the data which will be migrated.

You can configure the behavior of the alert router in case that Db2 stops. The alert router can disconnect from Db2 or stop.

Choose [Notify Alert Router](#) to notify the alert router of changed settings. The settings will automatically be saved before the alert router is notified.

## 6.6.2.7 Troubleshooting for SAPCL

When troubleshooting for SAPCL, analyze the `syslog` error message and the trace file `db2c1.<pid>.*.trc`.

To view the `syslog`, proceed as follows:

1. Call transaction `DBACOCKPIT`
2. Choose [Diagnostics](#) [Traces](#) [SAP System Log \(SM21\)](#)

To view the trace file `db2c1.<pid>.*.trc`, proceed as follows:

1. Call transaction `DBACOCKPIT`
2. Choose [Configuration](#) [SAP Collector Settings](#)  
The trace files are stored in the trace directory specified in the field *HFS path of trace directory*.

If the database connection was successful, you could also display the trace files from transaction `DBACOCKPIT` [Diagnostics](#) [Traces](#) [IFI Data Collector Trace](#) by selecting *Show trace* and *Execute*.

At this point, you can dynamically set the trace level for SAPCL. If `DBACOCKPIT` calls SAPCL, it displays its own version number in the `db2c1.<pid>.*.trc` trace file.

## 6.6.3 Seamless Failover of SAP Application Servers to Other Db2 Members

In Db2 data sharing topologies, you sometimes need 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 needs 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 reconnect its work processes to a different Db2 member dynamically.

To initiate the reconnection to a different Db2 subsystem, call transaction `DBACOCKPIT`, and choose [Configuration](#) [Main Connection Profile](#). From this screen, you can inspect the currently connected database host for each work process and then redirect the work processes.

To check the work process status to database host mapping, call transaction `DBACOCKPIT`, and choose [Diagnostics](#) [DB Connections](#) [Active DB Connections](#).

### Note

The database names that are specified for the work processes are neither Db2 subsystem nor Db2 member names. Instead, they are the database connection's logical names, as defined in the connection profile `connect.ini`.

The [Diagnostics](#) [DB Connections](#) [DB Connection List](#) shows all possible logical connections for the application server. 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 application server's work processes are reconnected to the subsystem, which is specified by the connection.

#### Note

Be aware that this mechanism is intended to temporarily move SAP work processes to another Db2 subsystem if you are using `db2dsdriver.cfg`. If you plan to change your data sharing topology on a longer-term for workload balancing, you should change the affinity lists in the `db2dsdriver.cfg` configuration file. You can accomplish this in an online manner in `DBACOCKPIT`. Choose [► Configuration > Failover Configuration](#). Otherwise, SAP work processes can establish a new database connection to the original primary member, such as when a work process terminates abnormally or is restarted.

## Automatic SAP Instance Reconnect to Another Db2 Member

There are ABAP function modules available that redirect SAP instances to another Db2 member. These function modules are RFC-enabled and can be invoked from automation software via the RFC SDK. You can call them from any operating system and in any programming language supported by RFC SDK.

### Functions Module Enablement

You can enable the function modules by downloading the Basis Support Package 10 or later.

The new parameters `WAIT_FOR_COMPLETION`, `MAX_WAIT_TIME`, and the ABAP report `RSDB2SWITCH` are provided in Basis Support Package 10 or later.

### STU3\_ADMIN\_SWITCH\_DB\_CON

The ABAP function `STU3_ADMIN_SWITCH_DB_CON` allows you to switch a single SAP ABAP instance or all ABAP instances of an SAP system to another DB connection (Db2 member)

#### Note

You must assign the user ID – under which `STU3_ADMIN_SWITCH_DB_CON` is invoked – the `S_DB2_DBADM` authorization profile.

### Sample Call of STU3\_ADMIN\_SWITCH\_DB\_CON

The document attached to SAP Note [915482](#) illustrates a sample call of this ABAP function in the C programming language using the RFC library.

### SWITCH\_ALL\_APP\_SERVERS

The import parameter `SWITCH_ALL_APP_SERVERS` controls whether a single instance or all instances are redirected. The default behavior is to redirect a single instance only.

### NEW\_DB\_CON

The import parameter `NEW_DB_CON` specifies the DB connection to which the work processes are supposed to be redirected. The connection names that you need to set are those that are used in the `connect.ini` configuration file.

### WAIT\_FOR\_COMPLETION and MAX\_WAIT\_TIME

The optional parameters `WAIT_FOR_COMPLETION` and `MAX_WAIT_TIME` allow you to call `STU3_ADMIN_SWITCH_DB_CON` to switch the DB connection synchronously. If `WAIT_FOR_COMPLETION` is set

to a non-blank value, the function waits until the SAP work processes have changed to the new Db2 member before it returns. `MAX_WAIT_TIME` allows you to limit the maximum wait time. It is specified in seconds. `MAX_WAIT_TIME`, and only has an effect if `WAIT_FOR_COMPLETION` is also specified.

### STU3\_ADMIN\_GET\_DB\_CON\_SYSWIDE

The ABAP function `STU3_ADMIN_GET_DB_CON_SYSWIDE` can be used to examine the current DB connection currently in use by each connection of the SAP work processes system. The Db2 member to which a connection has been established is also provided. This information is contained in the parameter `DB_CONNECTIONS`, which is an internal table.

### Integration of ABAP Reports in your External Scheduler

If it is more convenient to integrate ABAP reports in your External Scheduler, you can use the ABAP report `RSDB2SWITCH` for this purpose. It provides a wrapper report for `STU3_ADMIN_SWITCH_DB_CON`.

## 6.6.4 Homogeneous System Copy: Jobhead Correction

### Use

#### Note

This section is only relevant for customers using CCMS to monitor their SAP systems.

After copying your system, the CCMS jobhead still points to the former database SSID. To complete the homogeneous system copy, the SSID needs to be set to the target system.

#### Caution

Only experienced users should use this utility.

### Procedure

To set the SSID to the target system:

1. Call transaction `DBACOCKPIT`.
2. Choose **► Configuration ► Homogeneous System Copy: Jobhead Correction ►**
3. Modify the necessary data.

This screen allows you to copy all jobheads from D6Y0 to ABCD. By default, the original jobheads of D6Y0 will be removed. You can keep the original jobheads by selecting the check box.

The JCL job card (SAP term: Jobhead) of a JCL job determines, for example, the user under which the JCL job is executed or the message class. JCL jobs are used to LINKEDIT and BIND SAPCL and to retrieve the BACKUP SYSTEM Utility History. DB13 can create and upload JCL jobs for `RUNSTAS`, `REORG`, and `COPY`.

The jobheads are stored in table `DB2JOB` and SSID as part of the jobhead key. The key is `JOBNAME=JOBHEAD_<SID>`.

You can edit the jobhead in [Jobs > JOB Profile](#).

After performing a system copy (typically the database SSID changes), you need to copy or rename the jobheads to the new database SSID. If you skip this action, a default jobhead is used. The jobheads of the system copy source system are not used.

In the case of Db2 data sharing, you need to proceed with this action for each data sharing member.

In case that you do not use functionality based on JCL jobs, you can completely skip this action.

## 6.7 Jobs

### 6.7.1 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 want to have the

- JCL uploaded
- 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 transaction `DBACOCKPIT` in the *JOB Control* screen and have the creator `SAPR3`.

#### Note

If the *Job name* starts with `UTILS:` as shown for the last four entries in the picture above, this is the utility template used for stored procedure execution. Since this is not a JCL, you can insert only the utility name with valid utility options.

These templates will only come into existence after the first job submission (not upload) in the *DBA Planning Calendar*.

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 as follows:

1. Call transaction `DBACOCKPIT`
2. Choose [Jobs > JOB Profile](#)

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:

```
/usr/sap/<SID>/SYS/global/JESjoblog/
```

```
\usr\sap\<SID>\SYS\global\JESjoblog\
```

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 has to be identical to the TSO user ID) and the [TSO password \[page 147\]](#) entered in the *JOB Profile* screen under *Change. Password* for authentication on the host.

### ⚠ Caution

The name of the partitioned data set (PDS) has to be maintained in the *JOB Profile* screen:

1. Call transaction `DBACOCKPIT`
2. Choose [Jobs](#) [JOB Profile](#)
3. Choose *Upload*
4. Maintain the name of the PDS in the field *OS/390 Partitioned dataset for JCL jobs*.

Make sure that all possible Stored Procedure Users have UPDATE authority for this PDS.

### ℹ Note

For each action in the *DBA Planning Calendar*, you can define a separate job and member name for upload using the path indicated above.

### Upload Fail

Uploading data sets with stored procedure `DSNACCDS` can fail with return code 9990 `Dataset in use`. In this case, someone has the partitioned data set allocated that is the target of the upload.

To resolve this problem, you can either organize the data set in question or change the profile parameter *Partitioned dataset for JCL jobs* in the *JOB Profile* screen. Subsequently, restart transaction `DBACOCKPIT`

[Jobs](#) [DBA Planning Calendar](#) to reload the profile parameters.

## Execute Action Immediately

Db2 online utilities are executed with the stored procedures `DSNUTILU` and `ADMIN_UTL_SCHEDULE`. `DSNUTILS` is suitable for a single utility execution.

`ADMIN_UTL_SCHEDULE` executes utilities on multiple objects.

### ℹ Note

`ADMIN_UTL_SCHEDULE` utilizes `DSNUTILU` 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.

## 6.7.2 JCL Job Submission

The *JOB Control* screen (transaction `DBACOCKPIT`, [Jobs](#) > *JOB Control*) enables you to submit and monitor any z/OS JCL job from within the SAP system.

It is not only a standalone transaction but also an interface used by other transactions.

### 6.7.2.1 JCL Job Submission - Prerequisites

To *JOB Control* screen, your system must fulfill the following prerequisites for the SAP, Db2 and z/OS levels:

#### SAP System

To use the *JOB Control* screen, a user needs a minimum requirement authorization for database administration; in other words, you must enter `S_A.ADMIN` 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.

For instance, for Db2 utilities, this is either:

- Authority `SYSADM`
- Authority `SYSCTRL`, `EXECUTE` on the Db2 Connect Package and `SELECT` on all Db2 catalog tables

For more information, see [Security Considerations for Stored Procedures \[page 157\]](#).

#### 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 ensure the following:

- You have installed the stored procedures according to the instructions in *Installing the Db2 Stored Procedures*.
- 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) or configure the TSO user ID used in transaction `DB2J` as described in [Security Considerations for Stored Procedures \[page 157\]](#).

### Note

JCL job submission, together with the 'Connect User' setting, is not supported.

- The system knows your TSO password (call transaction `DBACOCKPIT` and choose [Jobs](#) > [JOB Control](#) > [Password](#) to enter your password)
- You have the 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.

## 6.7.2.2 Profile Parameters

We recommend that you enter values for **all** profile parameters. You can maintain them as follows:

1. Call transaction `DBACOCKPIT`
2. Choose [Jobs](#) > [JOB Profile](#)

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:

Profile Parameters in JCL Settings

Profile Parameters	Description
<a href="#">General</a> > <a href="#">Db2 run library</a>	A library that contains program <code>DSNTIAD</code>
<a href="#">Storage</a> > <a href="#">MGMTCLAS, STORCLAS, DATACLAS (SMS)</a>	Optional parameters. If not specified, the defaults of the corresponding ACS routine are used.
<a href="#">Storage</a> > <a href="#">Volume count</a>	Overwrites the value defined in <code>DATACLAS</code> , if needed
<a href="#">Upload</a> > <a href="#">OS/390 Partitioned data set for JCL jobs</a>	Partitioned data set for uploading z/OS jobs You may also specify the job names and member names for jobs to be uploaded.

### Note

This data set is **mandatory**.



Profile Parameters	Description
<a href="#">▶ Console ▶ Console output dataset ▶</a>	<p>Sequential data set for the requested part of the z/OS system log</p> <p>It will be overwritten if requested again.</p> <p>The format must be as follows:</p> <p><i>Organization:</i> PS</p> <p><i>Record format:</i> VB</p> <p><i>Record length:</i> 133</p> <p><i>Block size:</i> 27930</p> <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p><b>Note</b></p> <p>This data set is <b>mandatory</b>.</p> </div>
<p><b>Note</b></p> <p>All authorized SAP system users use these parameters.</p>	

## 6.7.2.3 Listing z/OS jobs

### Displaying a List of Specific z/OS Jobs

To display a list of all current z/OS jobs created by a specific SAP system user ID, carry out the following steps:

1. Call transaction DBACOCKPIT
2. Choose [▶ Jobs ▶ JOB Control ▶](#)  
The *JOB Control* screen appears.
3. Enter an SAP system user ID in the *Creator* field.

### Displaying a List of All Existing z/OS Jobs

To see all current z/OS jobs in the SAP system, choose *List all jobs* in the *JOB Control* screen.

## Utility Statements

All utility statements that are used for execution using stored procedures have a name that starts with DSNUTILS.

## 6.7.2.4 Creating and Saving z/OS JCL Jobs

To create a job:

1. Enter a job name

### Note

The character strings \$\$\$ and \* are not allowed in job names. An empty string as the job name is also invalid.

2. Choose *Create*  
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> .....).
3. Choose *Continue* to access the SAP editor.  
The variable \$JOBHEAD is already inserted in your text. A user specific jobcard replaces it at runtime.

### Example

For more information about creating a user-specific jobcard, see [Creating and Configuring an Individual Jobcard \[page 147\]](#)

4. Type in the JCL job and choose *Save* to save the job and leave the editor.

## 6.7.2.5 Changing z/OS JCL Jobs or Stored Procedure Utility Statements

To change a job or a stored procedure utility statement:

1. Select an existing job  
Choose *Change*
2. Enter a job sign and choose *Continue* to access the editor.

### Note

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.

3. If you want to protect your changes to stored procedure utility statements from SAP updates, choose *Change Profiles* and *General*.  
Select *Protect utility statement skeletons* and save the profile.

## 6.7.2.6 Deleting z/OS JCL Jobs

To delete a job, select an existing job and choose **JCL job > Delete job**. You are only allowed to delete your jobs unless you have the authorization profile S\_DB2\_DBADM.

You cannot delete stored procedure utility statements.

## 6.7.2.7 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.

## 6.7.2.8 Entering a TSO Password

If you submit a JCL job, the stored procedures log you on to TSO. The SAP system must know your TSO password.

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.

## Saving Your TSO Password in the SAP System

To save your TSO password in the SAP system:

1. Call transaction DBACOCKPIT
2. Choose ► *Jobs* ► *JOB Profile* ►

### ⚠ Caution

You cannot change your TSO password with this procedure. You can only change your password directly in the TSO interface.

### → Recommendation

The Db2 Connect user should be different from the ID used to execute stored procedures.

## 6.7.2.9 Creating and Configuring 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 jobcard, the following default is used:

```
// $USER $SUFFIX JOB USER=$USER, CLASS=B, MSGCLASS=X,  
// MSGLEVEL=(1,1)
```

The SAP system user ID will replace the variable \$USER 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*.

### Note

Ensure that there are no empty lines in your jobs.

## 6.7.2.10 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.

## 6.7.2.11 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 the SAP system user(s) appears, providing the following information:

Column Name	Description
<i>Jobname</i>	SAP System name of the z/OS Job
<i>Jobno</i>	z/OS job identifier
<i>Submitted by</i>	User ID of the user who has submitted the job
<i>Sub date</i>	The day when the job was submitted
<i>Sub time</i>	The time when the job was submitted
<i>Status</i>	Status of the z/OS job after submission or last refresh

Possible values of the *Status* field are:

Value	Description
<i>submitted</i>	Directly after submission
<i>unknown</i>	No information found on z/OS about the job
<i>active</i>	Job still active on z/OS
<i>finished</i>	Job has finished on z/OS

Value	Description
<i>not run</i>	No z/OS initiator assigned to the job Maybe no initiator started for the CLASS used by that job.
<i>hold</i>	z/OS job assigned to an initiator, but all initiators started for the CLASS used by that job are busy.

By choosing [Refresh](#) the status information of all jobs displayed on the list is 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](#).

## 6.7.2.12 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 client side's job output are 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.

## 6.7.2.13 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:

Column Name	Description
<i>Jobname</i>	SAP name of the z/OS job
<i>Filename</i>	Filename on the client-side. The file name contains the z/OS job ID and an extension numbered consecutively.
<i>Size</i>	Size of the file in bytes on the client-side
<i>Date</i>	Date when the file was transported from z/OS to the client-side
<i>Time</i>	The time when the file was transported from z/OS to the client-side

To display the file contents:

1. Select one or more lines of the list
2. Choose *Display files*.

## 6.7.2.14 Cleaning Up the Directory JESjoblog

The JCL Submission Service of CCMS uses Db2 stored procedure to access the JES component of z/OS. To make them compatible with the old protocol via FTP, the JCL jobs are first of all placed as file \*.jcljob\* on the application server in the directory:

```
/usr/sap/<SID>/SYS/global/JESjoblog
```

```
\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 JESjoblog directory.

### Displaying JESjoblog Contents

To display the contents of the directory, choose ► *Extras* ► *Clear directory...* ►

The contents of the JESjoblog directory is displayed.

### Displaying Content of JESjoblog Files

To display the contents of JESjoblog files, choose one of the following methods:

- On the *JOB Control* screen:
  1. Enter a job name under ► *Processing options* ► *Jobname* ►
  2. Choose ► *JCL job* ► *Display job* ►.
- On the *JOB Control* screen:
  1. Choose ► *Extras* ► *Clear directory...* ►
  2. Select one or more jobs on the list.
  3. Choose ► *File* ► *Display* ►

## Deleting JESjoblog Files

To delete JES joblog files, choose one of the following methods:

- On the *JOB Control* screen:
  1. Enter a job name under ► *Processing options* ► *Jobname* ►
  2. Choose ► *JCL job* ► *Delete job* ►.
- On the *JOB Control* screen:
  1. Choose ► *Extras* ► *Clear directory...* ►
  2. Select one or more jobs on the list.
  3. Choose ► *File* ► *Delete* ►

## 6.7.3 Job Descriptions

### Use

You can find the following functions under *Jobs* in the navigation tree of the DBA Cockpit. For more information about these functions, see the sections listed in the table.

Types of Job Scheduling

Function	Description
<i>Central Calendar</i>	<a href="#">Central DBA Planning Calendar [page 164]</a>
<i>DBA Planning Calendar</i>	<a href="#">DBA Planning Calendar [page 165]</a>

### Features

#### Job Scheduling in DBA Planning Calendar

To schedule jobs in the *DBA Planning Calendar* (transaction DB13), do the following:

1. Choose ► *Jobs* ► *DBA Planning Calendar* ►.  
A screen with a monthly calendar, a day calendar, and an *Action Pad* (list of available jobs) appear.
2. To schedule jobs for your system, choose a job from the *Action Pad* list.
3. To clean up all scheduled actions and all log files related to the job that has previously run, are presently running, or are expected to run in the future, choose the *Cleanup* indicator.

#### ⚠ Caution

This action is irreversible.

The following is a list of available jobs. The individual jobs are explained in more detail in the sections following the table. If you would like to integrate these jobs into external schedulers – for example, to ensure that a job starts when a nightly SAP batch job completes, follow the steps outlined in SAP Note [1264471](#).

Jobs Available in DBA Planning Calendar

Available Jobs	Job Execution Type
<i>Backup System at Db2 System Level</i>	Execute
<i>Backup for all SAP tablespaces</i>	Execute and Upload
<i>Backup for recommended objects</i>	Execute
<i>Backup for all SAP tablespaces (incred.)</i>	Execute and Upload
<i>Online reorg of one SAP index</i>	Execute and Upload
<i>Online reorg of one SAP tablespace</i>	Execute and Upload
<i>Online reorg of recommended indexes</i>	Execute and Upload
<i>Online reorg of recommended tablespaces</i>	Execute and Upload
<i>Update reorg,stats,copy recommendations</i>	Execute and Upload
<i>Rebuilding of one SAP index</i>	Execute and Upload
<i>Recovery of one SAP index</i>	Execute and Upload
<i>Recovery of one SAP tablespace</i>	Execute and Upload
<i>Update data for transaction DB02</i>	Execute
<div style="background-color: #e0e0e0; padding: 5px;"> <p><b>Note</b> Only valid for SAP Basis 7.00, 7.01, 7.1, and higher releases.</p> </div>	
<i>Update stats for all SAP objects</i>	Execute and Upload
<i>Update stats for one SAP object</i>	Execute and Upload
<i>Central Calendar Log Collector</i>	Execute
<i>Update conversion recommendations</i>	Execute
<i>Execute conversion recommendations</i>	Execute
<i>Update reorg,stats,copy recommendations</i>	Execute

### Job Execution Types

The following is a description of the job execution types mentioned in the table *Jobs Available in DBA Planning Calendar*.



- [Execute](#)  
Stored procedure DSNUTILS or ADMIN\_UTL\_SCHEDULE executes the utility control statements.
- [Upload](#)  
A JCL job containing utility control statements is created and uploaded with FTP to the database server host.

### Backup System at Db2 System Level

Performs a backup of the complete Db2 subsystem using the `BACKUP SYSTEM` utility

The `BACKUP SYSTEM` utility invokes z/OS DFSMSHSM to copy the volumes on which the Db2 data and log information reside.

### More Information

For more information about the environment for the `BACKUP SYSTEM` utility, see the IBM documentation *Db2 for z/OS Utility Guide and Reference*.

For availability information, see SAP Note [1225355](#).

### Backup for all SAP tablespaces

Performs backups of tablespaces, index spaces, and LOB tablespaces

The `COPY` utility performs the backups.

The following backup options are available in transaction DB2J ► [Profile](#) ► [COPY](#) ⌵:

- [Backup all objects in the Db2 subsystem](#)
- [Backup ABAP & Java systems and Db2 catalog](#)
- [Backup ABAP systems and Db2 catalog](#)

### Special Considerations

The default is [Backup ABAP systems and Db2 catalog](#).

In an MCOD environment, all SAP systems in the Db2 subsystem are backed up.

### Backup for recommended objects

Performs backups of suggested tablespaces, index spaces, and LOB tablespaces

The suggestions are retrieved from the stored procedure DSNACCOX. The `COPY` utility performs the backups.

### Special Considerations

The backup options mentioned in the previous job description do not apply.

In an MCOD environment, the suggestions for all SAP systems are considered.

### Backup for all SAP tablespaces (increment)

Performs incremental backups of tablespaces, index spaces, and LOB tablespaces

The `COPY` utility performs the backups.

The following backup options are available in transaction DB2J ► [Profile](#) ► [COPY](#) ⌵:

- [Backup all options in the Db2 subsystem](#)
- [Backup ABAP & Java systems and Db2 catalog](#)

- [Backup ABAP systems and Db2 catalog](#)

### Special Considerations

The default is [Backup ABAP systems and Db2 catalog](#).

The COPY utility is called with the utility option CHANGELIMIT to realize incremental functionality. You can change the value for CHANGELIMIT in transaction DB2J ► [Profile](#) ► [COPY](#) ⌵.

In an MCOB environment, all SAP systems in the Db2 subsystem are backed up.

### Online reorg of one SAP index

Performs a reorganization of a specific index

### Special Considerations

You can reorganize only indexes of the current SAP system.

This job does not apply to Db2 catalog indexes.

### Online reorg of one SAP tablespace

Performs a reorganization of tablespace and LOB tablespace

### Special Considerations

You can reorganize only table spaces and LOB tablespaces of the current SAP system.

This job does not apply to Db2 catalog tablespaces.

If a base tablespace is specified, the associated LOB tablespaces are not reorganized.

### Online reorg of recommended indexes

Performs a reorganization of suggested indexes

The suggestions are retrieved from the stored procedure DSNACCOX.

### Special Consideration

Only suggested indexes of the current SAP system or the Db2 catalog are reorganized.

### Online reorg of recommended tablespaces

Performs a reorganization of suggested tablespaces and LOB tablespaces. The suggestions are retrieved from the stored procedure DSNACCOX.

The following reorganization options are available in transaction DB2J ► [Profile](#) ► [REORG](#) ⌵:

- [All tablespaces, which are suggested, are reorganized](#)
- [Only suggested LOB tablespaces are reorganized](#)
- [Non-LOB tablespaces will be reorganized, even if suggested](#)  
(Only suggested non-LOB tablespaces)

### Special Consideration

Only suggested tablespaces and LOB tablespaces of the current SAP system or the Db2 catalog are reorganized.

### Update stats for recommended objects

Compiles new statistics of suggested tablespaces, index spaces and LOB tablespaces. The suggestions are retrieved from the stored procedure `DSNACCOX`.

The following `RUNSTATS` options are available in transaction `DB2J` ► [Profile](#) ► [RUNSTATS](#) ►:

- [RUNSTATS also on catalog tablespaces \(Beware performance impact at runtime\)](#)
- [No RUNSTATS on catalog tablespaces \(Beware outdated statistics\)](#)

#### **Special Consideration**

Only suggested tablespaces, index spaces, and LOB tablespaces of the current SAP system or the Db2 catalog obtain new statistics.

#### **Rebuilding of one SAP index**

Rebuilds a specific index.

#### **Special Considerations**

You can rebuild only indexes of the current SAP system.

This job does not apply to Db2 catalog indexes.

#### **Recovery of one SAP index**

Performs recovery of a specific index

#### **Special Considerations**

You can recover only indexes of the current SAP system.

This job does not apply to Db2 catalog indexes.

#### **Recovery of one SAP tablespace**

Performs recovery of specific tablespace and LOB tablespace.

#### **Special Considerations**

You can recover only tablespaces and LOB tablespaces of the current SAP system.

This job does not apply to Db2 catalog tablespaces.

#### **Update data for transaction DB02**

Performs a refresh of the data shown in transaction `DB02`

The job retrieves z/OS catalog information regarding the data sets used by Db2 and calculates the size history and growth history of the database objects.

#### **Note**

Only valid for SAP Basis 7.0 and higher releases

#### **Update stats for all SAP objects**

Compiles new statistics for all tablespaces, index spaces, and LOB tablespaces

The following `RUNSTATS` options are available in transaction `DB2J` ► [Profile](#) ► [RUNSTATS](#) ►:

- [RUNSTATS also on catalog tablespaces \(Beware performance impact at runtime\)](#)
- [No RUNSTATS on catalog tablespaces \(Beware outdated statistics\)](#)

### Special Consideration

Only tablespaces, index spaces, and LOB tablespaces of the current SAP system or the Db2 catalog obtain new statistics.

### Update stats for one SAP object

Compiles new statistics for a specific table

The associated indexes and LOB tablespaces obtain new statistics, as well.

### Special Considerations

Only new statistics are compiled for tables of the current SAP system.

This job does not apply to Db2 catalog tables.

### Update reorg,stats,copy recommendations

Checks all database objects if `REORG`, `RUNSTATS`, or `COPY` Utility has to be performed.

### Special Considerations

This Planning Calendar Job is only displayed in the Planning Calendar action pad if the checkbox "Update reorg,stats,copy recommendations via DB13 instead of RZ20" is selected.

### Update conversion recommendations

Performs the analysis phase of the Conversion Tool (see also [SAP Db2 Conversion Tool \[page 167\]](#) section). In this phase, database objects like tables or tablespaces are detected, which need to be converted.

### Special Considerations

Depending on the SAP Basis Release and Db2 Release, this job might not be displayed.

### Execute conversion recommendations

Performs the Conversion Tool's conversion phase and performs conversions on objects analysed in the Jobs "Update conversion recommendations".

### Special Considerations

Depending on the SAP Basis Release and Db2 Release, this job might not be displayed.

### Central Calendar Log Collector

Performs log collection for the *Central Calendar*

To collect the calendar data of remote systems, carry out the following steps:

1. Call transaction `DBACOCKPIT`.
2. Choose *System Configuration*.
3. Double-click the appropriate database system in the database list.
4. On the *Administration Data* tab page, select the *Collect Central Planning Calendar Data* indicator.

### Modify Recovery

The job is executed automatically after the following jobs:

- *Backup of all SAP tablespaces*
- *Backup of suggested objects*
- *Increment backup for all SAP tablespaces*

## Special Considerations

The job *Modify Recovery* is not available in the *Action Pad*. To activate this feature, carry out the following:

In transaction, `DB2J` ► *Profile* ► *REORG* ► set the value of *Age for MODIFY RECOVERY* to greater than zero (>0) in the appropriate field.

See also SAP Note [1173622](#).

## 6.7.4 Security Considerations for Stored Procedures

Stored procedures are executed by the user connecting to Db2. To ensure successful execution of the stored procedures, the executing user needs an OMVS segment and the authority to execute the action in question.

### JCL Settings Screen

For your convenience, the security model is maintainable in the *JOB Profile* screen. To obtain the screen, call:

1. Transaction `DBACOCKPIT`
2. Choose ► *Jobs* ► *JOB Profile* ►
3. Choose *General* and choose the user that you want to execute the stored procedures:

#### Db2 Connect User

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 the *JOB Profile* screen under *ChangePassword* 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.

On the other hand, Db2 Connect user should have a password that does not expire (see [User and Password Management \[page 61\]](#)). Some company's security policies may prohibit giving this user an OMVS segment.

#### Administrator

If you choose *Administrator*, then the old behavior is emulated by performing a multi connect to Db2 under the administrator's authority. Each administrator have an SAP user with the same name as his TSO ID and has to maintain his password. The DBA actions will be executed under the administrator's TSO user that has scheduled it.

#### User

If you choose *User*, then you can specify a user in the related input field. All DBA actions are executed under this user by performing a multi connect 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 (1) and User (3):

Security Models 1 and 3 are protected from uncontrolled stored procedure executed by the SAP security system. 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 help you when logged on to your system without needing an identically named TSO user, which is often not possible due to company policies.

## Regarding Security Models with Administrator (2) and User (3):

The user, that will execute the DBA actions needs one of the following sets of authorities in the Db2 subsystem:

- `SYSADM`
- `SYSCTRL`, `EXECUTE` on the Db2 Connect Package and `SELECT` on all Db2 catalog tables.

You can best achieve this with the `db2radm` tool as detailed in SAP Note [843808](#). For example, issue the following command on the OS system console on which your application server is running:

```
db2radm -m db2i -L <DDF-Location>-P <DDF-Port>-S <SSID>-H <hostname>
```

`-Q` is a user that executes the DBA actions

`-U` is a user that executes the DBA actions

`-u` is the granting user that needs `SYSADM` authority

`-p` is the password of the granting user

You should omit option `-C <CollectionID>` if the *Default*'s connection is not configured in table `DBCON`. If it is configured and the parameter `PS` is maintained in column `CON_ENV`, then the option should be identical to this parameter.

## 6.7.5 Restarting Jobs

The utility executions started from the DBA Planning Calendar can be aborted for different reasons, for example, if there is not enough 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.

Utilities detect automatically if a restart is necessary. This means that the DBA Planning Calendar does not offer the possibility to restart the utility explicitly with the `RESTART (PHASE)` option but reschedule the failed job.

For example, selecting the job in the DBA Planning Calendar and choosing *Execute* is sufficient.

## 6.7.6 Adjusting the Jobs

It is possible to adjust the job skeletons used in the DBA Planning Calendar to suit your 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 (transaction `DBACOCKPIT` ► [Jobs](#) ► [DBA Planning Calendar](#) ►), proceed as follows:

1. Call transaction `DBACOCKPIT`
2. Choose ► [Jobs](#) ► [JOB Control](#) ►
3. Choose [List all jobs](#) to display all jobs.
4. Select a job whose skeleton you would like to modify. The names of the jobs are self-explanatory. For example the `RUNSTATS` job skeleton for an individual table (or tablespace) is called `RUNSTATS_TABLE`.

### Note

The job must first be selected in the list and then chosen by choosing the [Select job](#).

5. Choose [Change](#) to go to an editor in which you can make changes to the skeleton.  
If the job name does NOT start with `UTILS`, the job is suitable for uploading a JCL job. There any meaningful addition that will generate a syntactically correct JCL job is allowed. However, keep the job's rough structure, as the internal parsing routines will otherwise not be able to deal with the skeleton.  
If the job name start with `UTILS`: the utility template will be used for stored procedure execution. Since this is no JCL, you can only the utility name with valid utility options there. Do not insert any JCL cards.  
The following is an example of a section of the `RUNSTATS_TABLE` skeleton:

### Sample Code

```
RUNSTATS
  TABLESPACE $DB.$TS
  TABLE ($TB)
  SAMPLE ($SAMP)
  COLUMN (ALL)
  INDEX ($IX)
  SHRLEVEL (CHANGE)
```

To use the history option, you can change the skeleton here as follows:

### Sample Code

```
RUNSTATS
  TABLESPACE $DB.$TS
  TABLE ($TB)
  SAMPLE ($SAMP)
  COLUMN (ALL)
  INDEX ($IX)
  HISTORY ALL
  SHRLEVEL (CHANGE)
```

6. Save the changes. They will come into effect the next time the relevant job is run.

### Note

Be aware that the stored procedure templates will only come into existence after the first job submission (not upload) in the DBA Planning Calendar. If you wish to protect your changes to stored procedure

templates against interference with SAP Support Packages, you can do so in transaction `DBACOCKPIT` under [Jobs > JOB Profile](#).

## 6.7.7 Reorganization

The `REORG` job allows the reorganization (by a 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 Tablespace

#### ⚠ Caution

Depending on the DB2 Release, different share levels are used to reorganize LOB tablespaces:

- DB2 9: `SHRLEVEL REFERENCE`
- DB2 10 and higher: `SHRLEVEL CHANGE`

Availability will be restricted, if you perform `REORG LOB` with DB2 release lower than DB2 10

For all tablespaces for which inline `RUNSTATS` is allowed, inline statistics are generated.

### When a Mapping Table is Necessary

When performing an online reorganization of a tablespace (not an index), you need to create a mapping table (see the IBM documentation *DB2 for z/OS Utility Guide*).

#### ⚠ Caution

The names `QT<nnnnnn>` or `QX<nnnnnn>` are reserved for this application. Do not use these names to create any tables or indexes.

### Profile Parameters for REORG

The following profile parameters exist for the reorganization. You maintain these in the *JOB Profile* screen as follows:

1. Call transaction `DBACOCKPIT`
2. Choose [Jobs > JOB Profile](#)
3. Choose the respective tab as listed below.

#### Storage Parameters (Tab Storage)



- *HLQ for backup data sets*
- *HLQ for work data sets*
- *MGMTCLAS (SMS)*
- *STORCLAS (SMS)*
- *DATACLAS (SMS)*
- *Volume count*  
Overwrites the value defined in *DATACLAS*, if necessary.
- *A Number of GDG generations:*  
If a value larger than zero is specified, then the copy and work data sets are allocated as GDGs. For more information about GDG generation, see [Backup \[page 118\]](#).

#### General

- *DB2 run library*  
A library containing the program DSNTIAD
- *Plan of program DSNTIAD*

#### REORG-specific parameter (Tab REORG)

- *Number of parallel jobs*
- *Size of work datasets/unload dataset:*  
Space requirement grows with the number of indexes; default 2, increase if necessary.
- *Which COPY options are to be used:*  
You can specify if a remote copy should be taken, as well.

#### Note

The REORG jobs use data from the table and index monitor.

REORG ALL indexes and tablespaces use the same parameters as a single REORG.

## 6.7.8 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`.

## 6.7.9 Update Statistics (RUNSTATS)

From the DBA Planning Calendar, you can update the statistical information for one or all SAP tablespaces or those tablespaces for which a `RUNSTATS` is recommended.

### Special Considerations for 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

Several tablespaces, whose statistics should not be updated to make the Db2 optimizer favor index-based access (see also [Optimal Access Paths Assurance with RUNSTATS \[page 259\]](#)). These are excluded automatically from any update for all SAP tables and recommended SAP tables.

Suppose you have a special reason to update statistics for a certain tablespace that falls into the above category. In that case, 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.

## Profile Parameters for RUNSTATS

The following profile parameters exist for `RUNSTATS`. You maintain these in the *JOB Profile* screen, which you find as follows:

1. Call transaction `DBACOCKPIT`
2. Choose ► *Jobs* ► *JOB Profile* ►

The *JOB Profile* screen appears with the following options:

### RUNSTATS

- *Number of parallel jobs*  
You can choose the number of jobs to be executed in parallel, speeding up the process.
- ► *Default* ► *Low/medium/high percentage of rows sampled* ►  
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*  
`RUNSTATS` includes the `HISTORY` option. If specified, every time the related catalog tables are inserted or updated, some selected fields are copied to some new catalog history tables. This results in preserving the relevant statistical data across multiple `RUNSTATS` executions and creates a basis for monitoring the database objects growth and assistance in determining optimal allocation quantities. Additionally, you could use the historical data 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. You can set `AGE` from **0** to **32767** days. The utility deletes all records older than the specified number of days. If you select **AGE=0**, `MODIFY STATISTICS` will not be used.

#### Note

Since SAP does not yet use the `HISTORY` option of `RUNSTATS`, the default value of transaction `DBACOCKPIT` ► *RUNSTATS* ► *Age of MODIFY STATISTICS* ► 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 want to use the historical data for reinstating old catalog statistics in case of access path degradation, a value of **30** is sufficient.
- If you want to monitor database objects growth, a value of **365** to **730** days is more appropriate.

## 6.7.10 When to Exclude Tablespaces from RUNSTATS, REORG, and COPY

You might want to exclude tablespaces from the process of updating statistics and raising alerts for several reasons.

### DB2NORUN and DBSTATC

To exclude tablespaces from consideration in Update statistics for all SAP objects (`RUNSTATS ALL`) and `RUNSTATS` on objects needing new statistics (`RUNSTATS ALERTS`), do one of the following:

- Insert the tablespace name and database name into table `DB2NORUN` using transaction `SE16`.
- Insert the tablespace name and database name into table `DB2NORUN` **manually** using `SPUFI`.
- You can also use table `DBSTATC` 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`.

#### Note

Setting the `ACTIVE` flag in table `DBSTATC` only covers `RUNSTATS ALERTS`. Excluding tablespaces, from `RUNSTATS ALL` is only possible via `DB2NORUN`.

#### Caution

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.

### DSNACCOX Exception Table

SAP exploits real-time statistics and the stored procedure `DSNACCOX`. The exception table of `DSNACCOX` will exclude any tablespace or index space in the subsystem for alerts on utility runs.

## 6.7.11 Central DBA Planning Calendar

The Central DBA Planning Calendar creates a single point of control for managing multiple databases in SAP system environments. It contains different types and versions of database systems. Generally, it also enables the administration of non-SAP databases. However, you can only manage Db2 for z/OS databases if they are SAP databases. The Central Planning Calendar can be especially beneficial for managing MCOB landscapes. To use the Central DBA Planning Calendar, call transaction `DBACOCKPIT`, and choose [Jobs](#) [Central 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.

### Refresh Display

To refresh the system's display 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 a background job)
- Schedule as Job (the scheduled jobs run daily at the specified time)

You can monitor the refresh jobs' execution state 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 choose the database systems of the same type you want to schedule this action. This facilitates the scheduling of identical DBA actions on multiple databases. You can update or delete planned activities for remote systems on the Central Planning Calendar.

#### Note

The DBA Planning Calendar of each database is also capable of running on SAP systems that use 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 the central and local DBA Planning Calendar is very similar. The main difference is that the central version is only for viewing DBA actions by systems (from here, you can easily switch to the local version). The local version shows activities 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 activities 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.

For more information about the Central Planning Calendar, see the SAP online documentation.

## 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 doubleclicking 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 choose the database systems of the same type you want to schedule this action. This makes it easier for you to plan identical DBA actions on multiple databases.

## 6.7.12 DBA Planning Calendar

Some database administration tasks are very time-consuming or can only be carried out when they are 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 in your SAP system's DBA Cockpit..

To use the DBA Planning Calendar, choose one of the following:

- Call transaction `DBACOCKPIT` and choose 

- Call transaction DB13.

Actions can be scheduled in advance using background processing. These actions are then executed automatically.

## 6.7.12.1 DBA Planning Calendar: Basic Functions

You can execute the following basic functions in the *DBA Planning Calendar*:

Function	Procedure
Schedule a new action	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 specific time delay (in weeks) to avoid planning repeated activities more than once.  Except for the update of transaction DB02, you can schedule any action either for execution on the z/OS host or upload a corresponding z/OS 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.
Change a scheduled action.	Select an action and choose <i>Edit</i> . You can change all action parameters.
Display results of executed actions	Select an action and choose <i>Action Details</i> . Unsuccessful actions are highlighted in red in the calendar.
Display parameters of an action	Select an action and choose <i>Action Details</i> .
Start an action immediately.	You can choose any action and press the <i>Execute</i> button. A copy of this job will be created and executed immediately.
Delete a scheduled action	Select the action you want to delete. Choose <i>Delete</i> . This function is only possible for future actions.
Restart an aborted action	Position the cursor on the relevant action and choose <i>Execute</i> . Choose a date and time for the restart.

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.

You can also initiate all actions if you choose *Execute* instead of *Add*. The actions then run using background processing. This enables you to execute an action manually if, for example, the planned action was unsuccessful.

## 6.7.12.2 DBA Planning Calendar: Preparation

All database administration tasks must be secure. Therefore, authorization checks must be made for certain operations in the SAP system, the database system, and 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

You must authorize the user must be authorized to run the Db2 utility corresponding to a specific administration task.

You need to have set up the stored procedures according to the instructions *Installing the Db2 Stored Procedures*.

For more information, see the IBM documentation *Db2 12 for z/OS Utility Guide and Reference*.

### z/OS Host

For information about how to set up z/OS specifics, see *Installing the Db2 Stored Procedures*.

#### Note

You must maintain **all** your profile parameters in the [JOB Profile](#) screen:

1. Call transaction `DBACOCKPIT`
2. Choose [Jobs](#) > [JOB Profile](#)
3. Depending on your security setup, you may need to provide a user and password under [Change](#).

## 6.7.13 SAP Db2 Conversion Tool

### 6.7.13.1 General Information

Currently, we support 6 conversion type in the **Conversion Tool**:

- Encryption
- Sep Multitable TS
- Linline LOB
- Fixed Length
- BRF to RRF
- 6 to 10 bytes RBA

With the rollout of Db2 10, IBM introduced the inline LOB feature (short: **Inline LOB**). Implementing Inline LOB for an existing LOB column requires a *Forced Conversion* of the object, during which the object will not be accessible. To avoid this, a generic SAP Db2 Conversion Tool (short: **Conversion Tool**) was designed to handle various kinds of conversions for Db2 on IBM Z using an online REORG with SHRELEVEL CHANGE. For more information about Inline LOB, see [Inline LOB \[page 193\]](#).

Another conversion type available with the Conversion Tool is Fixed Length 1 conversion (short: **Fixed Length**). It is currently available for Db2 9 and Db2 10 in combination with SAP Basis Release 7.03/7.31. This conversion type changes the representation of character columns defined with a length of 1 in the SAP Data Dictionary from variable length to fixed length. For more information about Fixed Length, see [Fixed Length Character 1 \[page 197\]](#).

With the rollout of Db2 11, IBM introduced the Extended RBA/LRSN feature, which extends the Relative Byte Address (RBA) and Log Record Sequence Number (LRSN) from 6 to 10 Bytes.

With the rollout of Db2 9, IBM introduced the Reordered Row Format (RRF) feature, an alternative row format to Basic Row Format (BRF).

For more information, see the IBM documentation *Db2 for z/OS Administration Guides*.

All of these conversion types offer a reduction in CPU consumption and costs. However, they require thorough analysis since they potentially alter many objects and may need multiple online REORGs. Therefore, you should carefully plan any any conversions.

## Use Cases

There are two types of use cases for the Conversion Tool:

- One-time conversions per object (such as Fixed Length, BRF to RRF, or 6 to 10 Byte RBA)  
Fixed Length conversions would typically be done once to convert any columns created before SAP Basis Release 7.03/7.31. Adding additional columns or creating new objects with columns of variable character length of 1 is automatically adjusted by the SAP ABAP Data Dictionary, so there is no need to convert for Fixed Length in those cases.
- Frequent conversions per object (such as Inline LOB)  
Adding new LOB columns to an existing object will not be inlined automatically. It may be advisable to call the Conversion Tool every once to check for new non-inlined LOB columns.

Instead of manually converting and reorganizing single objects, you can easily implement these conversions for more than one object at a time. Also, the Conversion Tool can mass process objects on local and remote systems. A remote function call (RFC) setup needs to be prepared for ABAP systems, which is described in [Prepare the Technical Setup \(Technical Prerequisites\) \[page 170\]](#).



## Note

Any conversions require careful planning and attention to the current technical prerequisites. For more information about these technical prerequisites, see [Prepare the Technical Setup \(Technical Prerequisites\) \[page 170\]](#).

Suppose you start a new Conversion Update Job. In that case, all older recommendation data are deleted, which means that an older mass conversion was not fully executed; this remains unconverted on the DB2 database.

You can never mix two conversions at the same time! But you can later repeat the Conversion Update Job for any conversion type if necessary.

## New DBA Planning Calendar Jobs

The Conversion Tool comes with two new DBA Planning Calendar jobs that correspond with two phases of the conversion process, the analysis phase and the conversion phase.

- The first job is called “Update conversion recommendations” (hereafter: Conversion Update Job) and fetches a certain number of affected objects, and their required attributes needed for the selected conversion type. It stores the analysis results in the control table DB2CONVTOOL. Successfully running this job completes the **analysis phase**.  
The result can be reviewed in the [Job Log](#) of the Conversion Update Job or displayed in the Conversion Tool section of the DBA Cockpit (hereafter: Conversion Viewer), which accommodates a review of the analysis results in more detail. To thoroughly review and possibly manipulate some or all analyzed objects, use Transaction ► [DB2](#) ► [Jobs](#) ► [Recommendation](#) ► [Conversions](#) ►. This phase is called the **review phase**. It is optional and, therefore, can be skipped within the overall conversion process. We recommend using the Conversion Viewer Transaction ► [DB2](#) ► [Jobs](#) ► [Recommendation](#) ► [Conversions](#) ►, especially for first-time users.
- The second job is called “Execute conversion recommendations” (hereafter: Conversion Execute Job) and can be scheduled directly after the Conversion Update Job results are available or after the results’ optional review. This job marks the **conversion phase**. During this phase, conversions and online REORGs are performed for all or a subset of the analyzed objects which are not excluded from conversions or REORGs and which are technically feasible to be converted.  
If an object requires any other changes before conducting the actual conversion, these Collateral Conversions are conducted first. They may include enlarging page size (changing buffer pool), adjusting row format to RRF, or using a Universal Tablespace (UTS) space type. They are a prerequisite for Inline LOB conversions. For more information about Inline LOB prerequisites, see [Inline LOB \[page 193\]](#). Only these Collateral Conversions are triggered by the Conversion Tool during the next Conversion Execute Job. After these collateral changes are finished, you will wexecute the actual conversion (for example, Inline LOB) during the next Conversion Update, and Conversion Execute Job cycle. This means yet another online REORG is required for the affected objects.  
Once a Conversion Execute Job has run successfully, the initial analysis results are no longer visible in the Conversion Viewer since it automatically deletes the content of the control table DB2CONVTOOL. This table only contains the most current analysis results yet to be executed.

Use of the SAP Db2 Conversion Tool involves the following three steps, which are explained in detail within the following sections of this document:

1. Run *Update conversion recommendations* job (Conversion Update Job/analysis phase)
  - To generate the analyzed objects list, use Transaction ► [DB2](#) ► [Jobs](#) ► [DBA Planning Calendar](#) ► [Action Pad](#) ►
  - Select one conversion type from the drop-down list (if applicable) and start the job.
2. Optionally: Review the analyzed objects list (review phase)
  - To access the Conversion Viewer, which allows you to remove or add exclusions as needed for any particular objects, use Transaction ► [DB2](#) ► [Jobs](#) ► [Recommendations](#) ► [Conversions](#) ►.
3. Run *Execute conversion recommendations* job (Conversion Execute Job / conversion phase)
  - To convert and REORG the objects from the object list use Transaction ► [DB2](#) ► [Jobs](#) ► [DBA Planning Calendar](#) ► [Action Pad](#) ►.

#### Note

Since at least one or - in case of any Collateral Conversions for Inline LOB conversions - two online REORGs are an integral part of the whole conversion procedure. The Conversion Execute Jobs need to be scheduled carefully to avoid critical objects being processed during peak-times. Hence, the Conversion Update Job is designed to allow for the limitation of many objects to be analyzed (the initial SAP default is a maximum of 20) and can run the jobs for specific objects only.

We strongly recommend keeping inserts to these objects to a minimum during the REORG phase to avoid COPY-pending status. For more information, see *Chapter 25: REORG TABLESPACE* in the IBM documentation *IBM Utility Guide and Reference*.

## 6.7.13.2 Prepare the Technical Setup (Technical Prerequisites)

You must complete the following technical setup first before using any of the features described below.

### System Prerequisites

For *Inline LOB*: Db2 10 and SAP Basis Release 7.x and higher.

For *Fixed Length*: Db2 9 or Db2 10 and SAP Basis Release 7.03/7.31 and higher.

For basic *RBA* to extended *RBA*: Db2 9 and higher and SAP Basis Release 7.x and higher

For *BRF* to *RRF*: Db2 11 and SAP Basis Release 7.x and higher

## Remote Systems

To use the Conversion Tool remotely, set up the database and as the RFC connection first (the latter for ABAP only). For each remote system, use Transaction [DB2](#) > [DB-Connections, Transaction SM59](#) > [RFC Configuration](#) > [ABAP-Connections](#) > [Create](#) .

## Support Packages/Correction Instructions

The support packages and/or correction instructions in SAP Note [1529521](#) need to be installed in the local and remote destination systems. They activate the inlining of LOBs and the required DB2CONVTOOL control table. To check your support package level, call the menu [System](#) > [Status](#) .

For more information, see the following SAP Notes:

- [2765471](#) - Conversion Tool - Encryption
- [2829660](#) - Enhance Space Monitor with Encryption Info
- [2925598](#) - KPI Action "unencrypted objects"
- [3026995](#) - Conversion Tool - Encryption: indexes of empty DEFINE NO tables Separation of Multitable TS
- [3064516](#) - Conversion Tool - separation of multi-table table spaces (XSAP)

## Defaults

The following defaults apply for the Conversion Tool:

1. The maximum number of objects to be analyzed is initially set by SAP to 20 items. If necessary, adjust this maximum using:
  - For SAP Basis Releases 7.03/7.31 and higher:  
Transaction [DB2](#) > [Jobs](#) > [Job Profile](#) > [Miscellaneous](#) > [Conversion Tool](#) > [Number of objects per analysis](#) and enter a value between 1 and 9999999.
  - For SAP Basis Releases before 7.03/7.31:  
Transaction [DB2](#) > [Configuration](#) > [Job Profile](#) > [Miscellaneous](#) > [Conversion Tool](#) > [Number of objects per analysis](#) and enter a value between 1 and 9999999.

### Note

Regularly review whether the maximum number of objects to be analyzed is too high or too low. If required, you may manually change this number for one particular job in the [Schedule a new action](#) screen by erasing the pre-populated default value and inserting a more appropriate value for that one job. For more information about scheduling a Conversion Update Job, see [Generate the Analyzed Objects List \(Conversion Update Job\) \[page 172\]](#).

2. The maximum number of REORGs to be run in parallel is initially set by SAP to five. If necessary, adjust this maximum using:
  - For SAP Basis Releases 7.03/7.31 and higher:  
Transaction ► [DB2](#) ► [Jobs](#) ► [Job Profile](#) ► [REORG](#) ► [Number of parallel jobs](#) ►  
Enter and *Save* a new value.
  - For SAP Basis Releases before 7.03/7.31:  
Transaction ► [DB2](#) ► [Configuration](#) ► [Job Profile](#) ► [Miscellaneous](#) ► [Conversation Tool](#) ► [Number of object per analysis](#) ►  
Enter and *Save* a new value.

## REORG Prerequisites

Perform a “REORG for one tablespace” before starting the Conversion Execute Job. You may choose any tablespace. If you are unable to start the REORG job, check the parameters “JES held output class”, “DB2 load library”, and “DB2 run library”, in the DBACOCKPIT Job Profile ► [DBACOCKPIT](#) ► [Jobs](#) ► [Job Profile](#) ► [Tab General](#) ►. Repeat the test REORG until at least one REORG has been completed. For more information, see [Convert and REORG Objects from Object List \(Conversion Execute Job\) \[page 177\]](#).

### 6.7.13.3 Generate the Analyzed Objects List (Conversion Update Job)

Before executing a conversion with the Conversion Tool, you must run the corresponding Conversion Update Job in the DBA Planning Calendar. Otherwise, the Conversion Viewer using Transaction ► [DB2](#) ► [Jobs](#) ► [Recommendations](#) ► [Conversions](#) ► displays an empty list of recommendations along with an info message that the current analysis has no data available (see message window). Running the Conversion Execute Job, in this case, would render a job result “No objects processed”.

## Start the Conversion Update Job

To start the Conversion Update Job, use Transaction ► [DB2](#) ► [Jobs](#) ► [DBA Planning Calendar](#) ► [Action Pad](#) ► [Update conversion recommendations](#) ►.

On the [Action Parameters](#) tab, perform the following steps:

1. Choose the desired conversion type available for your (remote) system from the drop-down list (if applicable), for example, Inline LOB for Db2 10 or Fixed Length for SAP Basis Releases 7.03/7.31.
2. Specify whether you want to limit the number of objects to be analyzed at a time using one of the following options:
  - The pre-set maximum number of objects to be analyzed at a time, which is already pre-populated
  - Any other number of objects (1-99999999) to be used for this job only

### ⚠ Caution

Specifying large numbers may take some time during the analysis phase, depending on the number of affected objects. It may cause difficulties when scheduling the Conversion Execute Job with its associated REORGs.

3. Limit the analysis to a subset of objects you specify by object name or object name pattern using wild cards "\*" in the *Object filter* field.
4. Start the job in one of two ways:
  - Click the *Execute Immediately* button for an immediate start of the Conversion Update Job.
  - Specify another point in time in the *Planned Start* date and time fields and click *Add*.
5. Ensure proper sequencing that is Conversion Update Job successfully finishes before the Conversion Execute Job or any other Conversion Update Job starts.

### 📌 Note

For first time users, we recommend limiting the Conversion Update Job to a particular table to evaluate the impact on the system.

The Conversion Update Job is now visible in the *DBA Job Planning Calendar*. If required, refresh the button regularly to update the status color. Once the Conversion Update Job has finished, that is, the job color is green or red, you may want to visit the *Job Log* to review the preliminary results. Double-click the job line in the *DBA Planning Calendar* to open *Display Details of Action* and select the tab *Job Log*.

For more information about the objects, particularly about objects that would not need to be converted, see the Conversion Viewer using Transaction ► *DB2* ► *Jobs* ► *Recommendations* ► *Conversions* ►. If you are not interested in the details, start the Conversion Execute Job described in [Convert and REORG Objects from Object List \(Conversion Execute Job\) \[page 177\]](#).

## 6.7.13.4 Review the Analyzed Objects List (Conversion Viewer)

Once you have successfully finished running a Conversion Update Job, open the Conversion Viewer to review and, if required, manipulate the list of recommended objects. The use of the Conversion Viewer is optional. However, any changes made within this action directly impact the Conversion Execute Job to follow and on future cycles (Update and Conversion Execute).

### Start the Conversion Viewer

To start the Conversion Viewer, use Transaction ► *DB2* ► *Jobs* ► *Recommendations* ► *Conversions* ►.

### 📌 Note

Depending on your Conversion Update Job specifications, this list can still be empty. For example, if you limited the analysis to a particular table, that table does not contain any non-inlined LOBs. In that case, the analysis for Inline LOB would render no results. Another empty list is shown if all 20 (out of

the default maximum 20) analyzed objects cannot be converted and, thus, filtering by [Result Set](#) [Objects to be processed](#) does not render any results. For more information about how to solve this, see [Troubleshooting \[page 179\]](#).

## Selection Criteria

The first screen of the Conversion Viewer is designed to initially only show the [Result Set](#) of objects which will be processed, that will be converted and reorganized during the Conversion Execute Job. To view all objects that were analyzed, you need to change the [Result Set](#) criteria to [All analyzed objects](#) and press the button [Apply Filter](#).

If for example, the screen shows 20 [All analyzed objects](#): Five of these objects are not going to be inlined at all. Likewise, no Collateral Conversions are performed on these five objects either. One of these objects, table ABAPDOCU\_TREE, is deliberately excluded from conversions and/or REORGs. Four others are non-convertible due to space type limitations; the multi-table tablespace (MULTI) first needs to be manually converted to a universal tablespace (UTS) space type. You will inline only the 12 objects without any Collateral Conversions flags during the Conversion Execute Job. The three objects at the bottom of the list will only receive Collateral Conversions. For more information about the Conversion Execute Job, see [Convert and REORG Objects from Object List \(Conversion Execute Job\) \[page 177\]](#).

The [Selection Criteria](#) option [Object name](#) allows for a search by object name or object name patterns using the wild card(s) "\*" within the Conversion Viewer listing. Any combination of [Selection Criteria](#) is technically possible.

In case your screen is not showing all 20 objects at one time, use the scroll bar to scroll down the Conversion Viewer listing. Review and analyze the attributes of these analyzed objects as well. You may also sort or filter the list, such as [Excluded](#) and [NonConv](#) (non-convertible) flags, to review these particular objects.

## List of Objects

For each object in the list, the Conversion Viewer displays the following information besides the object name in the [Object](#) column:

- [Excluded](#) column: Indicates a flag for objects which are deliberately not to be converted. You have to set or remove this flag proactively. For more information about how to change these flags, see the [Modify list of Objects](#). The Conversion Execute Job will not convert these flagged objects even though you analyzed them due to the Conversion Update Job.
- [NonConv](#) and [Result](#) columns: Indicate whether objects are technically not convertible by providing a reason for such a limitation in the [Reason](#) column at the right end of the list. The Conversion Execute Job will not convert these objects either.
- [CvtInline](#) or [CvtFixLen](#) column: Indicates the recommended conversion type, converting to Inline LOB or Fixed Length.

For Fixed Length, Collateral Conversions are not necessary. Consequently, no more columns are visible in the Conversion Viewer in the case of Fixed Length recommendations.

For Inline LOB, the next columns indicate the need for any Collateral Conversions, which are additional conversions needed before performing the recommended conversion type:

- *CvtPgSize* column: Indicate the need to convert the page size to a higher value to accommodate the additional inlined Bytes within the base tablespace  
For more information about the prerequisites for Inline LOB, see [Inline LOB \[page 193\]](#).
- *CvtRowF* column: Converting row format from BRF to the required RRF
- *CvtSpType* column: Converting space type to UTS if the space type is not yet a UTS type  
For more information about the prerequisites for Inline LOB, see [Inline LOB \[page 193\]](#).
- The rest of the columns contain the current and required object attributes.  
For example, current page size (*PgSize* column) versus required page size (*ReqPgSize* column)  
Others are buffer pool, row format, and space type. The values for these minimum requirements are automatically determined for you during each Conversion Update Job.

### ⚠ Caution

There is no history kept of the details shown in the Conversion Viewer screen. If you want to save a snapshot, use the download option, and save the list to a local file.

## Modify List of Objects

You can change the *Exclude* flag or change the Layout of the list to show suppressed columns.

### Excluded Flag

The screen of the Conversion Viewer provides the option to actively set or remove the *Excluded* flag for one or more objects. However, changing this flag affects the current job and any future Conversion Update Jobs and even any other conversion types. For example, you set the *Excluded* flag during an Inline LOB conversion process for an object, and that object is also a candidate for Fixed Length conversion. The next Conversion Update Job for Fixed Length will also respect the Excluded flag set for this object, and the corresponding Conversion Execute Job will not convert the object either.

We do not recommend proactively setting this flag. If you must apply it anyhow, click the empty *Excluded* box on the main Conversion Viewer screen and press the *Save* button. Your choice is updated in the persistent Db2 table `DSNACC.EXCEPT_TBL`, which also stores exclusions for the utility REORG, RUNSTATS, and COPY.

### 📌 Note

For more information about how to check those Excluded objects regularly by visiting their entry in the Maintenance Exceptions table `DSNACC.EXCEPT_TBL`, see [Troubleshooting \[page 179\]](#)

Every conversion type requires at least one REORG, and, therefore, conversions are also prohibited if the REORG exception is set as well. In case you are trying to remove an *Excluded* flag on the Conversion Viewer screen for REORG exempted objects, an appropriate error message in the message window will signal the failure of removing the *Excluded* flag.

Any combination of conversion and REORG maintenance exceptions flags results in an *Excluded* flag in the Conversion Viewer listing. Consequently, in all of these cases, no conversions are performed during the Conversion Execute Job. For more information about how to review and change only conversion exceptions and REORG exceptions, see *Maintenance Exceptions*.

## Suppressed Columns

A few columns are being suppressed in the Conversion Viewer listing, namely schema/qualifier, object type, database name, and space name.

You have the option to view this information for all objects by changing the *Layout* and select the columns you want to see. Insert them into the current *Layout* by clicking on the left arrow and then the bottom's green checkmark.

To review this information for one single object at a time along with additional conversion information, see section [▶ Navigating from Object List ▶ Conversion Details ▶](#).

## Navigate from List of Objects

There are two options to navigate to related screens from the Conversion Viewer, **Conversion Details** and **Edit Exceptions**.

### Conversion Details

To find out more details about an object, click the link behind the object name on the Conversion Viewer screen.

The *Conversion Details* screen provides the following additional information:

1. General info: *Object Name*, *Object Type*, *Space Name*, *Qualifier (schema)* and for Inline LOB only, the values of the Inline LOB prerequisites in case they differ from the current values (Page Size, Space Type, Row Format and technical Reason)
2. Details of objects:  
The details of the object depend on the conversion type.
  - Inline LOB conversion:  
*Column Name*, *Column Type*, *Inline Length* (current base tablespace), *NewInLength* (new inline length in the base tablespace), *Column Length* as well as a listing of the Collateral Conversion values in the *Object Summary* section
  - Fixed Length conversion:  
*Column Name*, *Current Type* (current column type), *ReqColType* (required column type), and *Column Length*.

To navigate back to the Conversion Viewer screen, use the *Back* option.

### Maintenance Exceptions

In the row of an object, click any cell except the *Excluded* and *Object* cells to highlight that cell. Press the *Edit Exceptions* button to review and change the exceptions for this object. You have now left the SAP Db2 Conversion Tool area and reached the *Maintenance Exceptions* panel.

The panel offers the possibility to exclude or include conversions in addition to the existing utility exclusions such as COPY, RUNSTATS, and REORG (partitions).

*Save* any changes before leaving the screen.

To navigate back to the Conversion Viewer, start the screen, use the *Back* option.

If you have altered the conversions or REORG exception settings from flagged to not flagged or vice versa on the *Maintenance Exceptions* panel, the following happens:



1. The persistent Db2 maintenance exceptions table `DSNACC.EXCEPT_TBL` is updated. The changes you saved on the *Maintenance Exceptions* panel will still be there next time you come across any updated objects.
2. The list of objects in the Conversion Viewer is updated, which means the *Excluded* flag is either removed or added. For more information about the *Excluded* flag, see *Modify List of Objects*.

#### Note

Depending on your *Selection Criteria* in the Conversion Viewer, ► *Result Set* ► *Objects to be processed* ►, and this object might not be visible anymore in the list of objects displayed. Change the *Result Set to All analyzed objects* and pressing the *Apply Filter* button to see it again.

3. Any Conversion Execute Job run after these edits will respect your changes.

### 6.7.13.5 Convert and REORG Objects from Object List (Conversion Execute Job)

Once you have finished reviewing the list of recommendations and excluded or included any objects that you do not or do want to convert, you may start the actual conversion phase by running the Conversion Execute Job.

#### Objects requiring Collateral Conversions

If there are any Collateral Conversions needed for one or more objects, regarding page size, row format, and space type during an Inline LOB conversion, they will be processed first. Therefore, only the ALTERs and REORGs for such Collateral Conversions are executed for these objects during the Conversion Execute Job. The initially intended conversion for Inline LOB will not be conducted for these objects until the next Conversion Update, and Conversion Execute Job cycle for Inline LOB.

If, for example, three objects need to be converted to partition-by-Growth Universal Table Space and their page sizes need to be increased for the additional inline length to fit into the base tablespace. In this particular example, switching from BRF to RRF is not required but would also be performed if another set of objects requires such a Collateral Conversion. For more information on the prerequisites for Inline LOB, see [Inline LOB \[page 193\]](#).

Once you perform the next Conversion Update Job for Inline LOB conversion type, these objects will once again appear in your result list with a *ConvInl* flag but no longer any Collateral Conversion flags. The corresponding Conversion Execute Job will then perform the actual Inline LOB conversion to complete this group of objects' conversion process. Hence these objects undergo a total of two REORGs to finish the Inline LOB conversion.

## Objects without any Collateral Conversions

If, for example, twelve objects do not require any Collateral Conversions, they are technically feasible and not excluded from conversions. Hence, their LOB columns are inlined appropriately, and the REORG runs as part of the Conversion Execute Job.

Objects solely recommended for Fixed Length, BR/RRF, and 6 to 10 Byte RBA conversions generally fall into this category as they do not require any Collateral Conversions.

### ⚠ Caution

The Conversion Execute Job is processed in two phases. Phase 1 performs the ALTER statements. If one ALTER fails, all ALTER statements are rolled back. After the successful execution of all ALTER statements, a COMMIT is performed to complete Phase 1. Phase 2 executes the required REORG utility. Once any of these REORGs fail, the failed and not yet executed REORGs are not stored and cannot be triggered again. To prevent such failed REORGs during a conversion execution job, perform a "REORG for one tablespace" as explained in section [Prepare the Technical Setup \(Technical Prerequisites\) \[page 170\]](#) before starting any Conversion Execute Job.

## Start the Conversion Execute Job

### 📌 Note

We strongly recommend keeping inserts to these objects to a minimum during the REORG phase to avoid COPY-pending status. For more information, see *Chapter 25: REORG TABLESPACE* in the IBM documentation *IBM Utility Guide and Reference*.

To reach the Conversion Execute Job screen, use Transaction [DB2](#) > [Jobs](#) > [DBA Planning Calendar](#) > [Action Pad](#) > [Execute conversion recommendations](#) (double-click).

Optionally, you can specify intervals for Conversion Tool jobs. To do so, click on the [Recurrences](#) tab.

### 📌 Note

Keep your local and remote systems in sync according to SAP Note [1529521](#), before using this feature.

To start the Conversion Execute Job, choose one of the following options:

- Press the [Add](#) button to apply the [Planned Start](#) date and time you specified
- Press the [Execute Immediately](#) button.

Monitor the job status and review the [Job Log](#) tab results once the job has changed to colors green or red.

After the Conversion Execute Job has finished, the Conversion Viewer list will be empty until another Conversion Update Job has run. An info message in the message window will inform you that no (recommendation) data is currently available.

You may now prepare and run your next Conversion Update, and Conversion Execute Job cycle. For example, in the case when *Initial Screen of the Conversion Viewer Showing the Results Set Objects to be Converted* and *Conversion Viewer Showing Result Set All Analyzed Objects*, the next job cycle does the following:

- The Conversion Update Job will continue to list the objects with the *Excluded* and *NonConv*, flags and the Conversion Execute Job will still ignore them
- The tree objects which had previously received any Collateral Conversions are now listed again with an Inline LOB conversion flag only, and the Conversion Execute Job will convert these to Inline LOB
- The twelve objects for which you already completed the Inline LOB conversion are no longer recommended and, make room for twelve new objects to be analyzed and possibly converted.

You can see a history of all jobs and their *Job Log* results in the *DBA Planning Calendar* entry of any job. For more information about creating your history files from the Conversion Viewer list, see [Review the Analyzed Objects List \(Conversion Viewer\) \[page 173\]](#).

## 6.7.13.6 Troubleshooting

Solutions to symptoms and issues

Symptom/Issue	Solution
Database connection missing	Set up a database connection using Transaction <a href="#">DB2</a> . To specify DB values, see <a href="#">Prepare the Technical Setup (Technical Prerequisites) [page 170]</a> .
RFC system failure when calling STU3_ADMIN_CHECK_RFC	To create STU3_ADMIN_CHECK_RFC using correction instructions, see SAP Note <a href="#">1529521</a> .
SQL error 204- [IBM][CLI Driver][DB2] SAPS2.DB2CONVTOOL IS AN UNDEFINED NAME	To create the missing table DB2CONVTOOL using correction instructions, see SAP Note <a href="#">1529521</a> .
Cannot determine the inline length for RFC system failure.	To create DB2_GET_LOB_INLINE_LENGTH using correction instructions, see SAP Note <a href="#">1529521</a> .
There is no DSNACC.EXCEPT_TBL.	Create one as described in the <a href="#">Exception Table [page 46]</a> .
Conversion Update Job continues to analyze the same objects and claims 0 objects to be processed in <i>Job Log</i> and Conversion Viewer listing.	Increase the maximum number of objects to be processed simultaneously using instructions from <a href="#">Prepare the Technical Setup (Technical Prerequisites) [page 170]</a> . Restart the Conversion Update Job.  To see the additional objects listed, scroll down on the Conversion Viewer listing or sort it by <i>Excluded</i> and <i>NonConv</i> flags. The objects can be reviewed and executed if technically feasible.

## Symptom/Issue

## Solution

Make sure that a particular object recommended for Inline LOB or Fixed Length conversion is not being converted.

Set the *Excluded* flag for these objects.

Start a Conversion Update Job for the appropriate conversion type and use *Object filter* to specify the object name or object name pattern using wildcards '\*'. Once this particular Conversion Update Job has finished, review the exceptions as described in [Review the Analyzed Objects List \(Conversion Viewer\) \[page 173\]](#) ► [Navigating from List of Object Maintenance Exceptions](#) ►

Set the *Conversions* exception flag and *Save*. Then run regular Conversion Update, and Conversion Execute Job cycles.

Ensure that a particular object that would be eventually recommended for Inline LOB or Fixed Length conversion is being converted during a specific time frame only.

Cancel any other Conversion Update, or Conversion Execute Job cycles planned parallel to this Conversion Update, or Conversion Execute Job cycle.

Start a Conversion Update Job for the appropriate conversion type and use the *Object filter* to specify the object name or object name pattern using wildcards '\*'. Once this particular Conversion Update Job has finished, review the exceptions described in [Review the Analyzed Objects List \(Conversion Viewer\) \[page 173\]](#) ► [Navigating from List of Object Maintenance Exceptions](#) ►.

Remove any conversion or REORG exception flags and *Save*.

Schedule the corresponding Conversion Execute Job, which will process only this object at the time you specify.

In case the object requires Collateral Conversions for Inline LOB, repeat this job cycle one more time to perform the actual Inline LOB conversion.

REORG with AUX YES has ended in COPY-pending status for a partition-bygrowth tablespace with LOB columns.

To create a recovery base and remove COPY-pending status for the LOB tablespaces, run COPY with SCOPE PENDING and a LISTDEF utility control statement that includes the LOB tablespaces.

For more information, see the *Recommendation* in the IBM Documentation *IBM Utility Guide and Reference*.

### Note

We strongly recommend keeping inserts to these objects to a minimum during the REORG phase to avoid COPY-pending status.

For more information, see the *Restriction* in the IBM Documentation *IBM Utility Guide and Reference*.

For more information about the related SAP Notes, see the following Table:

SAP Notes related to the Conversion Tool

Note	Topic
SAP Note <a href="#">1439989</a>	Db2-z/OS:CCMS: DB13 REORG LOB Online + REORG Part. List
SAP Note <a href="#">1452370</a>	Db2-z/OS:v10: Release of Db2 10 for SAP Components (incl. Inline LOB)
SAP Note <a href="#">1492381</a>	Db2-z/OS:v10: Inline lob length
SAP Note <a href="#">1529521</a>	Db2-z/OS:CCMS: Inline LOB Conversion Tool
SAP Note <a href="#">1561861</a>	Db2-z/OS: DDIC corrections: Upgrade to 7.03/7.31 prereq. (incl. Fixed Length)
SAP Note <a href="#">1671517</a>	Db2-z/OS: VARCHAR FOR BIT DATA conversion to Fixed Length
SAP Note <a href="#">2765471</a>	DB2-z/OS:CCMS: Conversion Tool - Encryption
SAP Note <a href="#">2829660</a>	DB2-z/OS:CCMS: Enhance Space Monitor with Encryption Info
SAP Note <a href="#">2925598</a>	DB2-z/OS:CCMS: KPI Action "unencrypted objects"
SAP Note <a href="#">3026995</a>	DB2-z/OS:CCMS: Conversion Tool - Encryption: indexes of empty DEFINE NO tables
SAP Note <a href="#">3064516</a>	DB2-z/OS:CCMS: Conversion Tool - Encryption: indexes of empty DEFINE NO tables

## 6.8 Alerts

### 6.8.1 Alerts

#### DB Alert Router: Introduction

The stored procedure `SAPCL` starts a database alert router process that connects to Db2 via `RRS` and catches Db2 alerts via the `IFI` interface of Db2. It is a separate process because capturing database events is an awaiting task. This process exclusively listens to Db2 alerts.

Each alert router process 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.

#### Implementation Considerations

The following are implementation considerations concerning the DB alert router:

- There are only eight Db2 OP buffers per Db2 subsystem. Monitor programs such as the `SAPCL` allocate OP buffers exclusively. To run the database alert router, you must ensure that at least one OP buffer is unassigned.
- The alert router process reformats the exceptions events data and saves the data in specific tables for later display.

#### Displaying Database Alerts

You can view Db2 alerts in analysis monitors as follows:

1. Call transaction `DBACOCKPIT`
2. Choose *Alerts*
3. Choose an analysis monitor:
  - *Lock Escalations*
  - *Long Running URs*
  - *Deadlocks*
  - *Timeouts*
  - *Active Log Storage*

In the monitors, you can analyze the details of an event. You can select a period and display events that occurred during that time. For more information, choose one of the following entries: *Lock Escalations*, *Deadlocks*, *Timeouts*.

The database alerts become visible immediately in transaction `DBACOCKPIT` under *Alerts*. However, there may be a 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 because new alerts are fed into the CCMS Monitor Set by a regular 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 process catches all 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 process catches all members' exceptions in the same data-sharing group. Also, in an MCOB environment, you get the exception events of all SAP systems.

### Note

Since the exception event transactions do not filter, the exception events' 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.

## Starting the DB Alert Router

When invoked by the SAP system, `SAPCL` starts an alert router process. If a corresponding process has already been started, no new attempts are made to create it. The SAP system (or, more specifically, report `RSDB2_COLLECT_HOURLY`) tries to start the alert router process at the startup of the SAP system and periodically every hour.

You can also start the alert router process manually:

1. Call transaction `DBACOCKPIT`
2. Choose **Configuration** > *SAP Collector Settings* >
3. In the *Alert Router* screen area, choose *Start*

The Db2 trace class 31 is used for the alert router. It is assumed that only the alert router process of `SAPCL` 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 SAP system's alert router, they are stopped. In this way, it is ensured that Db2 traces hang when `SAPCL` crashes do not pile up. Instead, they are cleaned up with the next start of an alert router.

Only one alert router is started to collect events for the complete data sharing group when using data sharing.

To check if the DB alert router is running:

1. Call transaction `DBACOCKPIT`
2. Choose **Performance** > *Thread Activity* >
3. Sort the column *Plan name*

If the alert router is running, a corresponding Db2 thread with a plan name specified for `SAPCL` is displayed.

## DB Alert Settings

The alert router has the following standard settings for each extent alert threshold values:

- Low threshold = 25

- High threshold = 80

### Note

Suppose the Db2 feature to optimize extent sizing is enabled by setting ZPARM `MGEXTSZ` to `YES`, which is highly recommended. In that case, 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 CCMS Monitor Set analysis tool, for example, by double-clicking the node. This leads you to the Extent Monitor. Alternatively, you can reach the Extent Monitor by calling transaction `DBACOCKPIT` and choosing [Space](#) [Database Growth](#) [Extent monitor](#). 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:

- Call transaction `DBACOCKPIT`
- Choose [Configuration](#) [DB Alert Settings](#)
- 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. However, this does not delete the table entries in the history table, which means you can display events for up to 30 days even after the alert has been deleted.

## 6.9 Diagnostics

### 6.9.1 Storage Management

This section contains information about managing storage in your database:

- [Changing the Database Layout \[page 185\]](#)
- [Storage Parameters \[page 198\]](#)
- [Space Management \[page 213\]](#)



## 6.9.1.1 Changing the Database Layout

The following sections provide you with information on how to change the database layout:

- [Creating Tables and Indexes \[page 185\]](#)
- [Automatic Rebuild Index Function \[page 186\]](#)
- [Modifying the Structure of Tables and Indexes \[page 188\]](#)
- [Setting Indexes to NOT PADDED \[page 189\]](#)
- [Moving Tables \[page 192\]](#)
- [Rules for Self-Defined Objects \[page 192\]](#)

### 6.9.1.1.1 Creating Tables and Indexes

#### Tables

Regardless of the Db2 version, default storage parameters are taken from table `TADB2` if they are not hard-coded within the SAP system.

The following sections describe version-specific behavior for Db2 object creation:

#### Implicit Database Layout

Starting with Db2 10 and all SAP releases higher than 6.20, Db2 takes care of creating all necessary objects in Db2 internally to host tables and indexes like database, tablespace, index spaces, and LOB tablespace. Db2 creates all of these objects in the Db2 default storage group `SYSDFLT`. The SAP System only defines the columns of a table, its primary key, and indexes. For every table, Db2 creates a single-table tablespace that exclusively contains this table and carries the same name as the table.

#### Classic Db2 Database Layout with SAP

Before Db2 10 and depending on the SAP release, SAP used the following database layout with Db2:

- Non-buffered tables  
The table is placed 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.

#### Indexes

Within the classic database layout, the default storage parameters of an index are defined in table `IADB2`. Primary indexes are considered to be an integral part of a table. Therefore, they can only be created and

dropped with the associated table. Be aware that primary indexes are linked to their base table with an additional:

```
ALTER TABLE ... ADD PRIMARY KEY (...)
```

## 6.9.1.1.2 Automatic Rebuild Index Function

The automatic rebuild index function described in this section helps your system to avoid long-running `CREATE INDEX` statements. It allows the system to automatically generate secondary indexes with the option `DEFER YES` and a subsequent call of the stored procedure `REBUILD INDEX`. This is much faster than a simple `CREATE INDEX` call.

### Requirements

The automatic rebuild index function is part of all SAP NetWeaver enhancement packages.

For SAP NetWeaver 7.0, it is shipped with Basis Support Package `SAPKB70016` and Kernel Patch Level 156.

For SAP NetWeaver 7.1, it is shipped with Basis Support Package `SAPKB71007` and Kernel Patch Level 100.

Ensure that the `DSNUTILS` stored procedures are set up.

Once the `REBUILD INDEX` function is activated (see section *Use* below), index creation triggered by `DDIC` activation imports and `SAINT/SPAM` activities are handled as follows:

- The index is created with the option `DEFER YES`.
- The utility `REBUILD INDEX` is called using the specified options.
- The `REBUILD INDEX` utility output is written to the developer trace, which you can find in transaction `ST11`.

#### Note

Only if a problem arises is it partially visible within the activation log of the index. If errors occur, you must always analyze the developer trace file of the corresponding work process (file name `dev_w*`).

### Use

If the relevant `kernel` patch and the support package listed in the *Requirements* section above are applied to the system, an additional screen area called *Automatic Rebuild Index* appears within transaction `SE14` [▶ Edit](#) [▶ Storage parameters](#) [▶](#).

The screen area *Automatic Rebuild Index* on the screen *Db2 Storage Attributes: Table* offers the following options:

#### System-wide default

The first row, entitled *System-wide default* helps you administrate the REBUILD INDEX function's system-wide default settings. It contains the following features:

System-wide default

Field / Push Button Name	Description
<a href="#">OK</a>	Shows whether or not the system-wide usage of REBUILD INDEX (secondary indexes only) is activated for all indexes of this system <ul style="list-style-type: none"> <li>DEFAULT is "not used". SAP recommends to "used".</li> </ul>
<a href="#">USED/NOT USED</a>	Choose <a href="#">OK</a> to activate the automatic REBUILD INDEX function for all indexes on the system as a system-wide default.  When you choose <a href="#">OK</a> , a dialog box appears in which you can specify the REBUILD INDEX options. Alternatively, you can select <a href="#">Change</a> options to change them.
<a href="#">Cancel</a>	Choose <a href="#">Cancel</a> to deactivate the automatic REBUILD INDEX function as a system-wide default.
<a href="#">Enforce system-wide</a>	Choose to <a href="#">Enforce system-wide</a> to enforce the currently displayed REBUILD INDEX settings system-wide for all indexes. The settings are: <ul style="list-style-type: none"> <li><a href="#">USED/NOT USED</a></li> <li>Those settings listed under <a href="#">Change</a></li> </ul> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p><b>⚠ Caution</b></p> <p>When you choose <a href="#">Enforce system-wide</a>, all individual settings are switched off.</p> </div>
<a href="#">Change</a>	Choose <a href="#">Change</a> to modify the system-wide REBUILD INDEX options.
<a href="#">Info</a>	Choose <a href="#">Info</a> to display the SAP Note <a href="#">1158203</a> . This SAP Note contains information and updates related to the REBUILD INDEX feature.

### This table's indexes

In the second row, individual REBUILD INDEX settings for the table being displayed can be specified. The following features are available:

This table's indexes

Field / Push Button Name	Description
<a href="#">USED/NOT USED</a>	Shows whether or not REBUILD INDEX is used for all indexes of the selected table

Field / Push Button Name	Description
<i>OK</i>	<p>Choose <i>OK</i> to activate the automatic REBUILD INDEX function for all indexes of the selected table.</p> <p>When you choose <i>OK</i>, a dialog box appears in which you can specify the REBUILD INDEX options. Alternatively, you can select <i>ChangeOptions</i> to change them.</p>
<i>Cancel</i>	Choose <i>Cancel</i> to deactivate the automatic REBUILD INDEX function for all indexes of the selected table.
<i>Use default</i>	<div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 10px;"> <p><b>⚠ Caution</b></p> <p>By choosing the <i>Use default</i>, all previous individual table settings are overwritten by the system-wide default.</p> </div>
<i>Change</i>	Choose <i>Change Options</i> to modify the REBUILD INDEX options for the indexes of the selected table.

### 6.9.1.1.3 Modifying the Structure of 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 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 views on the table are dropped.
2. The original table is renamed by adding the prefix QCM (for example, TESTTAB becomes QCMTESTTAB).
3. All indexes on the table are dropped.
4. The new DB table, including its primary index, is created according to the table's new DDIC definition. 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 (QCM8TESTTAB).
5. The original table contents are transferred to the new table (INSERT / SELECT adjusting types and formats).
6. The original table that has been renamed is dropped (including its database and tablespace if they become empty).
7. The newly created database table is renamed to its last name (from QCM8TESTTAB to TESTTAB).
8. All secondary indexes and views are recreated.

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  
A table conversion is triggered. This is done to ensure the uniqueness of the new primary index.
- Adding a key field  
A `DROP` and re-`CREATE` of the primary index is performed.

## Secondary Indexes

Changes in the definition of secondary indexes are to adding or deleting a key field. Two cases have to be distinguished:

- Non-partitioning indexes  
The modifications result in a `DROP` and subsequent re-`CREATE` of the index using the new key definition.
- Partitioning indexes  
The index is dropped and re-created. Db2 automatically converts the table partitioning from index-controlled to table-controlled.

### 6.9.1.1.4 Setting Indexes to NOT PADDED

#### Use

Db2 index keys for varying-length columns can be of varying-length (`NOT PADDED`) or fixed-length (`PADDED`). Varying-length keys have the following advantages:

- Varying-length key columns usually result in smaller indexes because the index keys use less than the maximum number of bytes defined for the columns.
- Varying-length keys allow index-only access to data. Fixed-length keys for varying-length columns cannot do so.
- Other Db2 family members have varying-length index keys, so providing this capability increases compatibility with the Db2 family of databases.

#### → Recommendation

SAP and IBM recommend that you convert all indexes within an SAP system to be `NOT PADDED`.

## Prerequisites

To convert all indexes within an SAP system, you must fulfill the following requirements:

- Ensure that the `REBUILD` utility can be run using stored procedures and that the parameter `NUMTCB` in the WLM application environment is set to `NUMTCB=1`.
- Your log file directory (`DIR_GLOBAL`) must have more than 2 GB of space. `DIR_GLOBAL` is located in your system at the following paths:  
UNIX: `/usr/sap/<SID>/SYS/global`  
Windows: `\\<sapglobalhost>\sapmnt\<SID>\SYS\global`

## Procedure

1. To access the tool that automatically detects and adjusts indexes with padded columns, call one of the following:
  - Transaction `DB2_IXFIX`
  - Transaction `DBACOCKPIT`. In the navigation tree, choose **► Diagnostics ► NOT PADDED Check/ Convers. (DB2\_IXFIX) ▾**
2. On the selection screen that appears, specify the following input parameters:
  - *Table name*  
Limits the search for indexes to the name range specified
  - *Maximum number of hits*  
Limits the number of hits
  - *Max. number of parallel procs*  
Limits the number of parallel processes (otherwise, all available batch processes will be utilized by the adjustment process).
  - *Only indexes without datasets*  
If this box is selected, the tool only selects indexes that can be altered without a subsequent `REBUILD`.
3. Choose **► Program ► Execute ▾**.
4. Initiate index adjustment.  
The output list provides the following functions:
  - *Select all*  
Select all listed indexes
  - *Deselect all*  
Deselect all selections
  - *Fix online*  
Start online adjustment of indexes ( $\leq 10$  indexes)
  - *Schedule fix*  
Schedules up to 4 parallel batch jobs (only if that many batch processes are available)
  - *Log overview*  
Displays an overview of all logs related to the transaction
  - *Background jobs*  
Displays all batch jobs related to the transaction
  - *Goto->Database utility*

Go to transaction SE14

#### → Recommendation

1. Start with the online-adjustment of some smaller indexes to check whether the procedure works flawlessly.
2. Then choose a larger number of indexes and initiate a batch adjustment.

5. Start conversion of selected indexes.

#### ⚠ Caution

While converting, the tool performs `ALTER INDEX ... NOT PADDED` and `REBUILD INDEX`, which temporarily sets the index to status `REBUILD PENDING`. In most cases, the associated database table is not available for productive use during that process. Therefore, you should schedule the conversion of a larger number of indexes when there is no or very little productive activity on the SAP system.

To start the conversion:

1. Call one of the following:

- Transaction `DB2_IXFIX`
- Transaction `DBACOCKPIT`

In the navigation tree, choose **▶ Diagnostics ▶ NOT PADDED Check/Convers. (DB2\_IXFIX) ▶**

2. On the selection screen, specify the range for the *Table name*.
3. Specify the maximum number of hits (up to 99999).
4. Choose **▶ Program ▶ Execute ▶**.
5. Select affected indexes and choose *Schedule fix*.
6. Specify and save the batch parameters.

Depending on the number of indexes that need adjustment and the number of batch jobs that run in parallel, the process may run for several hours. The adjustment of one index takes approximately 1 second.

### Manual conversion

You cannot adjust indexes associated with the following tables automatically because it is crucial for the stability of a running SAP system that these tables can always be accessed. This may not be the case if the conversion process for these indexes fails for some reason:

- #LOB
- DDLOG
- DDNTT
- DDNTF
- DDSTORAGE
- IADB2
- IGDB2
- REPOSRC

The transaction `DB2_IXFIX` lists these indexes with status `CONVERT MANUALLY`. It is possible to adjust these indexes manually by performing the following steps:

1. Execute the SQL statement:

```
ALTER INDEX <CREATOR>."<INDEX>" NOT PADDED
```

2. Perform the utility:

```
REBUILD INDEX (<CREATOR>."<INDEX>") SORTKEYS STATISTICS
```

#### REPORT NO UPDATE ALL KEYCARD

Since the impact of such an adjustment on the SAP system's overall performance will be very small, it is also possible to leave these indexes as is with the `PADDED` attribute.

### 6.9.1.1.5 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 ►** to access the Db2 for z/OS-specific part of transaction `SE14`.
4. Modify the storage parameters of the table and related objects
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.

#### 📌 Note

The SAP system provides a tool to process many tables at a time. For more information, see [Mass Processing \[page 208\]](#).

### 6.9.1.1.6 Rules for Self-Defined Objects

#### ⚠ Caution

The SAP system can 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 classic range-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`)

With Db2's implicit object creation, SAP and IBM recommend staying with the layout described in [Implicit Database Layout \[page 15\]](#) and not creating self-defined tablespaces, databases or stogroups.

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 defined during the installation process. Until SAP release 4.5B, it was generally called `SAPR3`. Later installations employ the `SAP<SAPSID>` creator.
- When creating **stogroups**, use the following named 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 Y\* or Z\* databases.
- 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](#) ►.

### 6.9.1.1.7 Inline LOB

With the rollout of Db2 10, IBM introduced a feature to inline a portion of a LOB into the base tablespace (short: **Inline LOB**). It is implemented with one of the following SQL statements:

1. CREATE TABLE...COLUMN...INLINE LENGTH *integer*...
2. ALTER TABLE...ALTER COLUMN...SET INLINE LENGTH *integer*...  
*integer* specifies for
  - BLOB and CLOB columns: The maximum number of bytes that are stored in the base tablespace for the column
  - DBCLOB column: The maximum number of double-byte characters stored in the base tablespace for the column.

For more information about these SQL statements, see *IBM Documentation for z/OS SQL Reference*.

For more information about the Db2 10 rollout, see SAP Note [1452370](#).

In SAP installations, a large portion of the LOB columns is typically less than 2000 Bytes. Inlined LOBs speed up accessing the LOB data as they might eliminate the need to access the associated auxiliary tablespaces. Thus, the savings in CPU consumption can be substantial, depending on the use of LOBs in your SAP environment.

The steps needed to implement Inline LOB are:

1. ALTER TABLE ... ALTER COLUMN ... SET INLINE LENGTH *integer*
2. REORG TABLESPACE with:
  - SHRLEVEL REFERENCE (see SAP Note [1492381](#))
  - SHRLEVEL CHANGE.

Manually implementing Inline LOB for an existing object required a *Forced Conversion* of the object and rendered the object inaccessible (SHRLEVEL REFERENCE). To avoid this and to allow for convenient mass processing of a larger number of objects, SAP rolls out the generic SAP Db2 Conversion Tool (short: **Conversion Tool**). It handles various kinds of conversions for Db2 on IBM Z, including Inline LOB, while using SHRLEVEL CHANGE.

For more information, see [SAP Db2 Conversion Tool \[page 167\]](#).

## Inline LOB Defaults

The maximum inlined length of a LOB column in a base table space is predefined by SAP to 2000 Bytes. The SAP ABAP Data Dictionary automatically doubles this maximum to 4000 Bytes if the base tablespace contains only one LOB column.

### Note

Special tables like REPOSRC contain larger inline LOBs. For more information, see SAP Note [1492381](#).

## Technical prerequisites

There are several prerequisites for inline LOB which are critical for conversion.

### 1. Row Format

To inline LOBs into the base tablespace, reordered row format (short: RRF) is required. Thus, any object created with BRF needs to be converted to RRF first.

### 2. Space Type

Inline LOB requires a universal tablespace (UTS) space type to accommodate the Inline LOB feature. The following table shows the possible values.

Space Type conversions required for Inline LOB

Starting Space Type	Destination Space Type
Segmented tablespace with one table	Inline LOB conversion requires Partition by Growth UTS. Done by Conversion Tool.
Segmented tablespace with multiple tables	Conversion not possible until you move all tables into Partition-by-Growth UTS manually
Partitioned by Range	Inline LOB conversion requires Range Partitioned UTS. Done by Conversion Tool.
Range Partitioned UTS	Ready for Inline LOB.
Partition-by-Growth UTS	Ready for Inline LOB.

### 3. Page Size

The page size may need to be increased to fit the additional Inline LOB Bytes into the base table space. The following table summarizes of the possible Inline LOB values and their implications on the page size.

Inline LOB values per method used and their implications on the page size

Method used	1st LOB	2nd LOB	3rd LOB	and so on...	Page Size implications
CREATE object with <b>one</b> LOB	<=4000	n/a	n/a	n/a	Medium:  To handle the in-lined Bytes in the base tablespace, the page size must be large enough to fit all records.
Transaction SE11 or transport/support package/upgrade (ALTER): adding LOBs	<=4000	no inline	no inline	no inline	Small:  For each non-in-lined LOB column, the page size needs to accommodate Bytes for their references to the auxiliary LOB tablespace.
Conversion Tool (ALTER + online REORG): adding LOBs	<=4000	<=2000	<=2000	<=2000	Large:  To handle the in-lined Bytes in the base tablespace, the page size must be large enough to fit all records
Transaction SE14 ( <i>Forced Conversion</i> ): adding LOBs	<=2000	<=2000	<=2000	<=2000	Large:  To handle the in-lined Bytes in the base tablespace, the page size must be large enough to fit all records

Method used	1st LOB	2nd LOB	3rd LOB	and so on...	Page Size implications
CREATE object with <b>multiple</b> LOBs	<=2000	<=2000	<=2000	<=2000	Large: To handle the in-lined Bytes in the base tablespace, the page size must be large enough to fit all records

## Inline LOB during a CREATE statement

During the creation of a new object with Db2 10, LOB columns are inlined as follows:

1. 2000 Bytes each if there is more than one LOB column, but less than 2000 Bytes per LOB column to fit all records into the maximum page size.
2. 4000 Bytes if there is only one LOB column in the object.

## Additional LOB columns with Transaction SE11

To add LOB columns with Transaction SE11, consider the following rules:

1. Objects created with Db2 10 containing **only one** LOB column:  
Transaction [SE11](#) will not inline these new LOB columns automatically. Use the Conversion Tool to inline these additional LOBs. During this process, the original Inline LOB columns' size will not be reduced to avoid a REORG pending restrictive state. Therefore, the LOB column's original inline length remains 4000 Bytes even if other LOBs are added later. These additional LOBs will be 2000 Bytes or less to ensure that the records fit into the maximum Db2 page size.
2. Objects created with Db2 10 with initially **more than one** LOB column:  
Transaction [SE11](#) will not inline these new LOB columns. Use the Conversion Tool to inline these additional LOB columns with 2000 Bytes or less to ensure that the table records fit into the maximum Db2 page size. In rare cases, this might mean that the first LOB column is 4000 Bytes, the second column is 2000 Bytes, and the following *n* LOB columns are less than 2000 Bytes depending on the total maximum record size.

## DROP/CREATE (with downtime)

Performing a [Forced Conversion](#) of an existing object will reset all LOBs to a maximum of 2000 bytes or less if the records do not fit into the page otherwise. This is also the result even if the object initially contained only one LOB column (4000 Bytes) and later had others added via an ALTER statement (4000 Bytes plus *n* times 2000 Bytes). This option makes the object inaccessible, though.

For instructions on how to manually inline LOBs with a downtime, see SAP Note [1492381](#).

## 6.9.1.1.8 Fixed Length Character 1

Short fix length columns consume less space in the CPU caches and allow you to save CPU cycles for larger SAP workloads that put a heavy strain on it. Also, fixed character columns need less disk space. Since a character field with a length of one typically never gets enlarged, it can be created with a fixed length instead. Therefore, SAP decided to change the representation of character columns defined with a length of 1 in the SAP Data Dictionary from variable length to fixed length.

This feature is called **Fixed Length** and is available with SAP Basis Release 7.03/7.31 in combined with Db2 9 or Db2 10. The following table shows the affected Db2 column types and their Fixed Length equivalents:

Fixed Length conversion results

Regular Db2 Column Type	Fixed Length Db2 Column Type
VARCHAR 1	CHAR 1
VARGRAPHIC 1	GRAPHIC 1
VARCHAR for Bit Data 1	CHAR for Bit Data 1

For more information, see [SAP Note 1671517](#).

The *SAP ABAP Data Dictionary* continues to allow both column types to coexist in a system.

For more information about the rollout of the Fixed Length feature with SAP Basis Release 7.03/7.31, see SAP Note [1561861](#).

## Adding New Columns with a Length of 1

To apply Fixed Length for new columns, SAP has redesigned the SAP ABAP Data Dictionary and R3LOAD utility to convert to Fixed Length automatically. Perform the database check-in Transaction **SE14** **Check Database object** for a particular object to view this converted Db2 column type or query the appropriate Db2 catalogs using Transaction **DB2C**.

## Converting Existing Columns with a Length of 1

Implementing Fixed Length for an existing object with Transaction **SE14** would require a *Forced Conversion* of the object and, would result in downtime for this object. To avoid such downtime and to allow for many objects' convenient mass processing, SAP rolls out the generic SAP Db2 Conversion Tool. It handles various kinds of conversions for Db2 on IBM Z, including Fixed Length.

The Conversion Tool analyzes a particular set of objects to determine whether they contain column types with a variable length of 1 (VARCHAR, VARGRAPHIC, and VARCHAR for Bit Data) and converts them to their Fixed Length equivalents using.

- ALTER TABLE...ALTER COLUMN...
- REORG with SHRLEVEL CHANGE

Exempted from this process are objects which are excluded from any conversions or REORGs.

For more information, see [SAP Db2 Conversion Tool \[page 167\]](#).

## 6.9.1.2 Storage Parameters

The following sections provide you with information on these topics:

- General Overview
- Editing Storage Parameters
- Changing the Source of Storage Parameters
- Special Actions
- Directly Changing Storage Attributes
- Moving Tables to Existing Tablespaces
- Isolating and Combining Tables
- Partitioning Tables
- Handling Large Tables
- Mass Processing
- Incremental Conversion
- Troubleshooting

### 6.9.1.2.1 General Overview

Transaction SE14 provides a means to modify and save storage parameters of tables, indexes, and tablespaces.

To access the initial screen, call transaction SE14 and choose ► *Edit* ► *Storage parameter* ► (for tables/tablespaces) or ► *Edit* ► *Indexes* ► *Storage parameter (for indexes)* ►

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 database's current situation 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 and IADB2, and IGDB2; others are hard-coded (for example, 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 target Db2 objects' storage parameters. If the table does not exist, default values are taken. For indexes, the algorithm is similar. Saved storage parameters are always preferred to the database's current status, which is preferred to the default values.

## 6.9.1.2.2 Editing Storage Parameters

To edit storage parameters, call transaction SE14.

### Displaying Table Attributes

To display table storage attributes:

#### 1. Choose **► Edit ► Storage Parameters ►**.

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)
- Partitioning 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 / 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 only has a primary index.

You can also directly access the screen that displays index parameters:

1. Call transaction SE14 and specify the table name.
2. Choose [Continue](#)  
A list with all indexes is displayed.
3. Choose [Indexes](#)
4. Select the index
5. Select [OK](#).
6. Choose ► [Goto](#) ► [Storage parameter](#) ►.

## 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.
4. Choose ► [Goto](#) ► [Back](#) ► to return to the initial screen.



## 6.9.1.2.3 Switching Between the Sources of the Storage Parameters

You can retrieve storage parameters of tables and indexes from different sources:

### Default Storage Parameters

Choose ► [Source](#) ► [Default](#) ▾.

These are typically used if no special parameters are needed.

#### ⓘ Note

If the table already exist, all previous settings will be deleted once the default settings are saved.

### Current Status in Database

Choose ► [Source](#) ► [Status in database](#) ▾.

If the table does not exist, default parameters are show (source: `COMP`). The system also displays a size estimate of the table (pagesize x number of pages determined by `RUNSTATS`) if `RUNSTATS` results are available.

### Saved Storage Parameters

Choose ► [Source](#) ► [Saved](#) ▾.

If no parameters have been saved, the database's status is displayed (source: `DBS`).

## 6.9.1.2.4 Special Actions

### Triggering Table Conversions

1. On-screen *Db2 Storage Attributes: Table* choose: ► [Source](#) ► [Saved](#) ▾.  
Check the storage parameters that have been saved for this table.
2. To trigger the table conversion, call transaction `SE14`, and choose ► [Goto](#) ► [Force conversion](#) ▾.

## Displaying Current Background jobs

Within transaction SE14, you can start the following background jobs:

- Table conversion (see the 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](#) ►.

All available action logs are displayed.

## Incremental Conversion

On-screen *Db2 Storage Attributes: Table*, choose ► [Goto](#) ► [Incremental conversion](#) ►.

For more information about how to employ this feature, see [Incremental Conversion \[page 211\]](#).

### 6.9.1.2.5 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, the bufferpool. The actions are slightly different for tablespaces and indexes.

## Index

1. Call transaction SE14 and specify the table name.
2. Choose [Continue](#)
3. Choose [Indexes...](#), specify the index and choose [Storage parameters](#).
4. Choose ► [Attributes](#) ► [Display <-> Change](#) ►.
5. Modify index parameters if needed

6. Choose [Attributes](#) [Save; Attributes](#) [ALTER](#).

## 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](#) [ALTER](#).

### Note

For both indexes and tablespaces, the SAP System subsequently performs SQL statements of type `ALTER INDEX` or `ALTER TABLESPACE`.

For most storage parameters, you need to run the Db2 utility `REORG` and adjust the data in the database.

If the system returns the error message:

```
ALTER not possible
```

you cannot adjust the storage attributes with a simple `ALTER` statement. In that case, you have to trigger a conversion with transaction SE14 [Goto](#) [Force conversion](#).

## 6.9.1.2.6 Partitioning Tables

### Partitioning a Table

1. Collect LIMITKEY data (optional).
2. Move table to the partitioned tablespace.

These steps are described in more detail below.

### Collecting LIMITKEY Data

This transaction enables you to collect LIMITKEY data, which can be used to partition a table. It performs an ordered selection on an index's key fields and thereby determines a LIMITKEY setting, which splits the table into parts of equal size. (For performance reasons, you can use only the index fields for collecting LIMITKEY data.)

1. Call transaction SE14 and choose [Storage Parameters](#) [Goto](#) [LIMITKEY data](#).  
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 table's size, 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

1. Choose ► *Goto* ► *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 Range-Partitioned Tablespace

1. Choose *Display*
2. In the screen area *Target Tablespace*, choose *Partitioned*.
3. A dialog box appears. Choose one of the following partitioning options (the subsequent steps depend on the option that you choose):
  - *Range-partitioned: Specify manually*  
Enter the partition key, the number of partitions, and the LIMITKEY values.
  - *Range-partitioned: Use collected LIMITKEY data*  
Display/Collect LIMITKEY data (see subsection above *Collecting LIMITKEY Data*).  
Choose *Use for partitioning* to use the displayed LIMITKEY data to initialize the range-partitioning parameters. Otherwise, choose ► *Goto* ► *Back* ► *No* ►.
  - *Range-partitioned: Upload a file with partitioning infos*  
A dialog appears stating that the format of the file to be uploaded should be as follows:
    - *First row: Partitioning key*
    - *Remaining rows: Ordered ranges (LIMITKEYs) without partition numbers*  
After specifying the file name, the uploaded file is analyzed and used to initialize the partitioning key, number of partitions, and LIMITKEY values.  
Subsequently, a file explorer appears. Choose the file that you wish to upload.
  - *Growth-partitioned*  
By choosing *Growth-partitioned*, the storage attributes for growth-partitioning are initialized. The system employs the following default settings:
    - MAXPARTITIONS 256
    - DSSIZE 64G (as controlled by Db2 system parameters IMPDSSIZE)

4. Adjust tablespace (and index) storage parameters to your needs. In particular, check the LIMITKEY specifications for range-partitioning.
5. Return to the initial screen.
6. Choose ► *Attributes* ► *Save* ►.
7. To initiate the table conversion, call transaction SE14, and choose ► *Extras* ► *Force conversion* ►.

## Changing the LIMITKEY Specifications of a Range-Partitioned Table

1. Choose *Display*
2. Modify the LIMITKEY specifications
3. Choose ► *Attributes* ► *Save* ►.
4. Choose ► *Attributes* ► *Adjust in database* ►. The SAP system executes SQL statements of the form:  

```
ALTER TABLE . . .
```

to adjust the LIMITKEY specifications in the database.
5. 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 of a Range-Partitioned Table

1. Choose *Display*
2. Edit the number of partitions and the LIMITKEY specifications for the new partitions
3. Choose ► *Attributes* ► *Save* ►
4. To initiate the table conversion, choose ► *Attributes* ► *Adjust* ► in database
5. 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 tables LIMITKEY data that has been collected so far is deleted.

## 6.9.1.2.7 Moving Tables to Existing Tablespaces

### Use

#### Note

This section is only relevant to the classic database layout.

### Procedure

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 name of the database and tablespace.
3. Choose ► *Attributes* ► *Save* ▾ to save the target tablespace.
4. To trigger the conversion, call ► *transaction SE14* ► *Force conversion* ▾.

## 6.9.1.2.8 Isolating Tables

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 if tables are kept in a multi-table tablespace in these cases.

### 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, call transaction `SE14` ► *Extras* ► *Force conversion* ▾.

## 6.9.1.2.9 Combining Tables

### Use

#### ⓘ Note

This function is only available with the classic database layout.

Tables in a single-table tablespace, which are only small and rarely accessed may be moved to multi-table tablespaces to simplify database administration.

### Procedure

#### 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, call transaction SE14 ► *Extras* ► *Force conversion* ▾.

## 6.9.1.2.10 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 the following characteristics:

- More than 1 million entries
- 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:
  1. Enter the table name in transaction SE16 and choose ► *Table* ► *Table contents* ▾.
  2. Choose ► *Edit* ▾.
2. Create a quiesce point for the entire subsystem (for example, with `-STOP DB2`). We recommend that you make previous full image copies for all SAP data to accelerate a `RECOVERY`, if necessary. Ensure that afterward, no `write` accesses to the subsystem are carried out – apart from the actions described here.

To partition **transparent tables**, proceed as follows:

#### *Number of entries*

1. In transaction SE14, choose ► *Edit* ► *Storage parameters* ► to modify the table's storage parameters (for more information, see the preceding sections).

### ⚠ Caution

Do not initiate a table conversion!

2. Save the new storage parameters (choose ► *Attributes* ► *Save* ►).
3. Copy the table's contents to a sequential data set using Db2 utilities (for example, DSNTIAUL or REORG with option UNLOAD EXTERNAL).
4. Re-create the table (*Activate and adjust database* with option *Delete data* in transaction SE14). The SAP System uses the saved storage parameters when creating a database, tablespace, table, and indexes.

To partition **cluster or pooled tables**, proceed as follows:

1. Copy the table's contents to a sequential data set using Db2 utilities (for example, DSNTIAUL or REORG with option UNLOAD EXTERNAL).
2. Call transaction SE14.
3. Choose *Delete database table*.
4. Choose *Storage parameters*.
5. Specify and save the storage parameters for the partitioning of the table.
6. On the initial screen of transaction SE14, select *Create database table*.

To **convert a table** in a segmented tablespace to reside in a partition-by-growth universal tablespace (PBG UTS), proceed as follows:

1. Issue ALTER TABLESPACE ..... DSSIZE 64G MAXPARTITIONS 256
2. Run online REORG for this tablespace

To **reload and backup**, proceed as follows:

1. Reload saved table contents into the newly created table (option: LOG NO).
2. As a first backup, carry out a full image copy of the tablespace. This also cancels the Copy Pending.
3. Verify the number of entries using transaction SE16 (see step 1 above).

## 6.9.1.2.11 Mass Processing

### Use

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 input values of RSDB2MAS fall into two categories:

- *Filter*  
Here you specify filter criteria:
  - The *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 a *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 to *Reduce number of databases*, the system recreates single-table tablespaces in existing databases (up to 100 tablespaces according with the naming convention). This helps 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 to *Generate dependent reports after conversion* if you want to avoid bothering the users with a long report generation process after every mass conversion.

### Note

With the implicit database layout, the following options are no longer supported:

- *Multi-tablespace*
- *Tablespace of same type*
- *Reduce number of databases*
- *DEFINE YES*
- *Current DEFINE settings*

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`).

### Note

*Schedule requests* trigger 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 any other user does not access them 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 afterward.

- [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 more information, see the *Incremental Conversion* or 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 canceled) if the conversion of a table fails. In that case, call transaction SE14 and choose [DB requests > Terminated](#) to analyze and resolve the problem.

## Procedure

### Reducing the Number of Data Sets

#### Note

This section is only relevant to the classic database layout.

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 tablespace.

1. Call transaction SA38 and execute report RSDB2MAS.
2. Specify:

Reducing the Number of Data Sets - I

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:
  - Recreate in *tablespace of same type*
  - with *DEFINE NO*
  - 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:

Reducing the Number of Datasets - II

Maximum number of hits 100000

---

Table (empty)

---

Empty YES

---

Isolated in own tablespace NO

---

Buffered by R/3 \*

---

Tablespace data exists \*

---

Index data set exists YES

---

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`. 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.

## 6.9.1.2.12 Incremental Conversion

### Use

In contrast to the standard table conversion within transaction SE14 the incremental table conversion (transaction 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`.

## Procedure

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` worklist.
3. Subsequently, perform the actions described in the online documentation (choose ► *Short description* ► *Ad hoc* ►).

### 6.9.1.2.13 Troubleshooting

If a table conversion fails, there is no data loss. During the conversion process, the original table is renamed from `<TABNAME>` to `QCM<TABNAME>`. The renamed table is only deleted after completing all conversion steps.

The error processing should involve the following steps:

1. Analyze the problem
  1. Call transaction `SE14`. On the initial screen, the following message is displayed:  
*Request: "Adjust"*
  2. Choose *Analyze Adjustment* to analyze the error.
  3. Check the developer trace files (transaction `ST11`) for SQL errors.

2. Continue adjustment

To continue the conversion, you must perform the following steps.

1. 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, a stogroup may be full and additional volumes need to be added. Then `ALTER STOGROUP . . . ADD VOLUME ( . . . )` enables a continuation.
2. Select *Processing Type* in transaction `SE14` and choose *Continue adjustment*. Subsequently, the table conversion continues.

## 6.9.1.3 Space Management

### 6.9.1.3.1 Adding Volume Space

#### Use

##### 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>. You do not need to take immediate measures if a <VOLID\_NEW2>, and so on of the volumes added.

##### Full Volume

As a precaution, you may add additional volumes to the respective stogroups by changing the database layout as follows:

#### Procedure

1. Find all stogroups pointing to the full volume.  
To do this, execute the SQL statement:  

```
SELECT SGNAM 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*.

### 6.9.1.3.2 Data Compression

#### Use

Using data compression can significantly reduce the amount of DASD space necessary to store the data in many cases. Important points are listed below:

To enable data compression, for tablespaces, specify `COMPRESS YES` on `CREATE` or `ALTER TABLESPACE`.

To enable index compression, specify the `COMPRESS YES` option on the `CREATE` or `ALTER INDEX` statement.

The buffer pool of compressed index needs to be larger than 4 KB.

### Note

The default for SAP Unicode systems tablespaces is automatically `COMPRESS YES`. This is set at installation time. The Db2 system parameter `IMPTSCMP` controls whether implicitly created tablespaces are defined with `COMPRESS YES`.

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 significant data import.

### Note

The data is not compressed until a compression dictionary is built, done by the `LOAD` and `REORG` utilities only. They are enabling index compression with Db2 marks an index as advisory reorg-pending. Index compression will take effect when a `REORG INDEX, REBUILD INDEX` or `REORG TABLESPACE` is done.

- 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
  - Slight CPU overhead (in general)
  - Higher P-lock contention (data sharing only)
- Define more free space (`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.

### Note

The free space induced by data compression does not apply to the index spaces because indexes are not compressed.

- Take extra care with the tables with 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`).

In contrast to data compression, Db2 index compression is a software algorithm. While the index resides on disk in compressed form, it is uncompressed when an index page is read from disk into the buffer pool. A 4 KB index page, therefore requires a larger buffer pool page size. To take advantage of space reduction by index compression and ensure good performance, it is best to compress large and reside in the buffer pool for a long time until they are replaced by other pages or written back to disk.

## More Information

For more information, see the IBM Db2 for z/OS documentation.

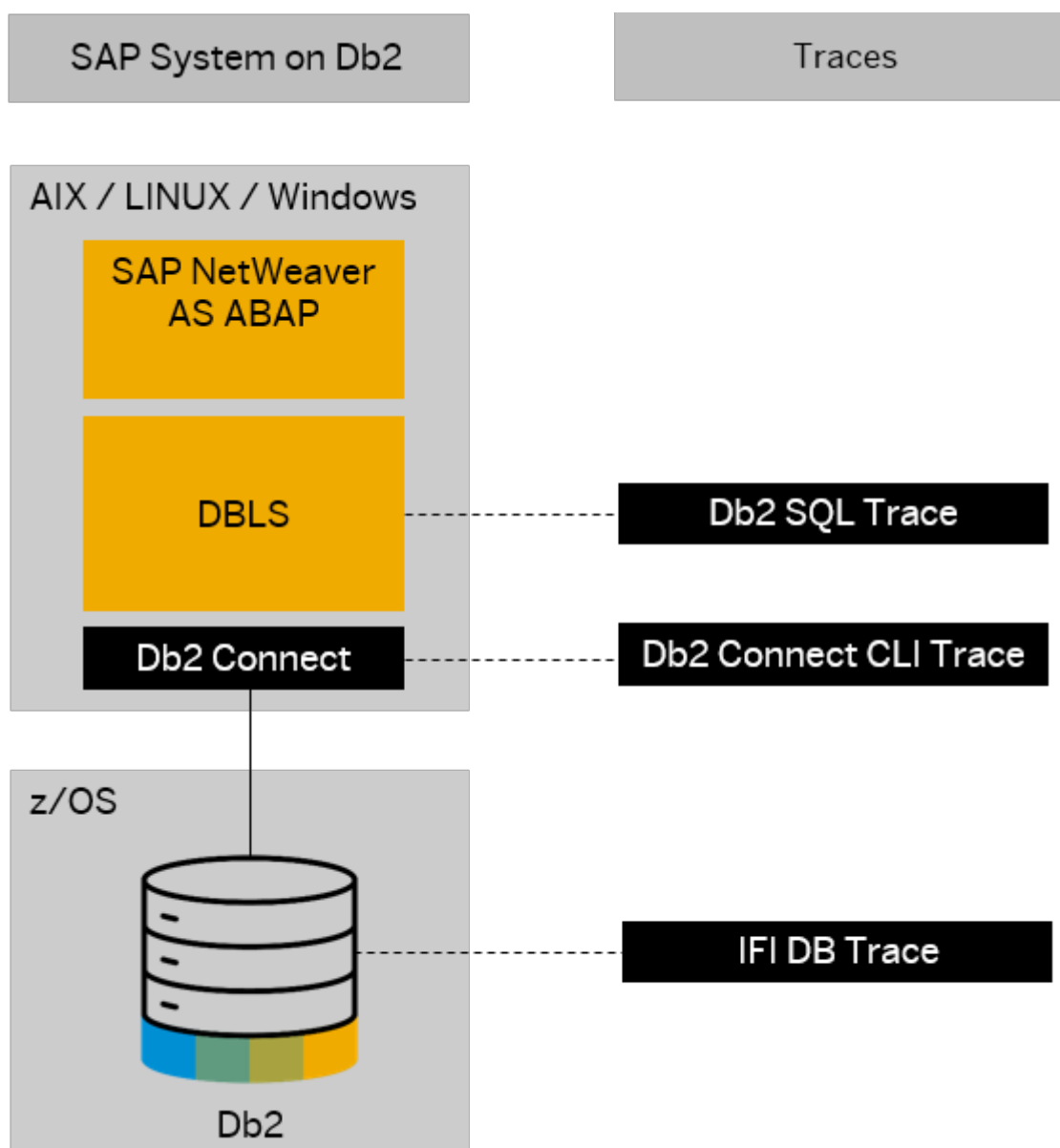
## 6.9.2 Traces

### Tracing Database Communication

The following figure illustrates some of the different traces available in the *DBA Cockpit* and where they are applied in your SAP system on Db2 for z/OS.

#### Caution

Tracing can have a considerable impact on the performance of the entire system. Therefore, please do not use these traces unless it is necessary.



Selected Traces Available in the DBA Cockpit

### DBSL SQL Trace

To display the *DBSL SQL Trace* screen, choose transaction `DBACOCKPIT` ► *Diagnostics* ► *Traces* ► *DBSL SQL Trace* ►.

The DBSL trace allows you to trace CLI statements. Experts should only use it.

The range of the trace levels spans from **0** to **5**:

- Level **0** is off
- Levels **1-5** collect trace information with increasing detail

When you set the trace level from **1-5** and choose *Execute*, the trace data is collected.

Setting the trace level to **0** stops the trace.



## Note

The trace output is written to the developer trace of the work processes.

### Db2 Connect CLI Trace

To display the *Db2 Connect CLI Trace* screen, choose transaction DBACOCKPIT ► *Diagnostics* ► *Traces* ► *Db2 Connect CLI Trace* ►.

With this tool, you can collect CLI trace information and display the CLI traces.

You can turn the CLI trace on either for all work processes or a single work process. If you start the trace by choosing *Start trace*, the SAP profile parameter `db2/db2/cli_trace` is set accordingly.

You can also change the profile parameter `db2/db2/cli_trace` 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 choose *Show trace*.

### IFI DB Trace

To display the *IFI DB Trace* screen, choose transaction DBACOCKPIT ► *Diagnostics* ► *Traces* ► *IFI DB Trace* ►.

With this tool, you can collect on the z/OS host side.

To execute a trace, choose one of the traces available in the *Trace Level* drop-down box and *Start Trace*:

- *SQL TRACE*
- *LOCK TRACE*
- *DATA TRACE*

Choose *ALL WORK PROCESSES* or the specific work process you would like to trace

To view an IFI DB Trace file, select a single work process in the *Work process* pull-down menu and choose *Show trace*.

## Tracing Data for SAPCL

To display the *IFI Data Collector Trace* screen for tracing SAPCL data, choose transaction DBACOCKPIT ► *Diagnostics* ► *Traces* ► *IFI Data Collector Trace* ►.

With this tool, you can collect trace information on the activity of SAPCL and the *Alert Router*.

The range of the trace levels spans from **0** to **6**:

- Level **0** is off
- Levels **1-6** collect trace information with increasing detail

When you set the trace level from **1-6** and choose *Execute*, the trace data is collected.

Setting the trace level to **0** stops the trace.

## 6.9.3 Automatic Start of DB Traces

The start of the relevant traces is the responsibility of the Alert Router. The ABAP system no longer tries to start the traces. For cases when there is no Alert Router, or it does not work, there is a button “Start necessary DB2 traces” on the Collector settings panel. The [SAP Collector Profile \[page 136\]](#) can control the behavior of the Alert Router regarding traces.

### Note

This task is run for remote systems, as well.

## 6.9.4 Db2 Connect Diagnostics

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 the following dialogs of SAP transaction DBACOCKPIT:

- [Db2 Connect Parameter Check](#)
- [DBA Network Statistics](#)
- [DB Ping](#)
- [Db2 Connect CLI Trace](#)

### Db2 Connect Parameter Check

To display the [Db2 Connect Parameter Check](#) screen, choose the transaction DBACOCKPIT ► [Performance](#) ► [Db2 Connect Parameter Check](#) ⌵.

### DBA Network Statistics

With Db2 Connect 9.7 FP3a, CLI Network Statistics have been introduced. To monitor and detect general network problems, a new action has been introduced from within the DBA Cockpit and can be accessed by choosing transaction DBACOCKPIT ► [Performance](#) ► [Db2 Network Diagnostics](#) ⌵.

### Note

The old network statistics, which were only available with the FAT client, have been removed.

Network statistics are stored in table DB2CLINETSTATS, which is created when entering the respective cockpit action if it is not already present. The collection of network statistics can be started/stopped via the DBA Cockpit. The related profile parameter to enable/disable the collection is `db2/db2/net_stats`.

The statistics are being accumulated to fixed intervals (5 minutes), aggregated by the work process, and written by the `db2` at the next commit after the collection interval has ended.

### Note

DBSL 7.20 Patch level 414 or higher is required.

The following metrics are received from the DBSL:

- Server time
- Total round trip elapsed time (network send/receive time and database processing time)
- Number of bytes sent to/received from the database server
- Number of DRDA\* round trips

### Note

Network statistics are only available for ABAP systems.

Two views are provided in the DBA Cockpit:

- Db2 Network Statistics Overview
- Db2 Network Statistics Details

#### Network Statistics Overview

Besides the usual time range filter, the network statistics selection panel provides options to filter by:

- Application server
- Work process type
- Work process number

Also, it is possible to aggregate by the previously named filter options.

#### Network Statistics Details

By clicking one of the overview table links to select a specific interval, you get to the respective interval details. The overview information is shown at the top of the page, and the min and max Samples concerning `avg. network time` is provided. The average values are also displayed.

That means:

- The smallest (min) `avg. network time` during that interval with all related counters in that sample is displayed.
- The average values for that interval are displayed.
- The largest (max) `avg. network time` during that interval with all related counters in that sample is displayed.

## DB Ping

To display the *DB Ping* screen, choose transaction `DBACOCKPIT` ► *Diagnostics* ► *DB Connections* ► *Db2 Ping* ►.

## Db2 Connect CLI Trace

To display the *Db2 Connect CLI* screen, call transaction `DBACOCKPIT` and choose [► Diagnostics ► Traces ► Db2 Connect CLI Trace ►](#).

In the *Db2 Connect CLI Trace* screen, you can check the settings of Db2 Connect and trace activity at the CLI level. The latter is covered in more detail in [Traces \[page 215\]](#).

The current values of the parameters of Db2 Connect that are specific to SAP are compared to the values that are recommended by SAP.

## 6.9.5 Data Sharing Topology

### Use

On occasion, it might not be clear which SAP application server is connected to which Db2 member when data sharing is employed. It is useful to view the data sharing members and their connected SAP application servers in this situation.

### Procedure

To access this information:

1. Call transaction `DBACOCKPIT`
2. Choose [► Diagnostics ► DB Connections ► Data Sharing Topology ►](#)

It provides an overview of the application server connections for every member and provides a break down per work process type down to the individual work process. This is especially helpful, e.g. after a planned Failover has been triggered.

## 6.9.6 Web Dynpro EXPLAIN (Extended Version)

### Use

The extended EXPLAIN uses additional EXPLAIN tables. EXPLAIN tables contain information about SQL statements and functions that run on Db2 for z/OS.

The following tables are used:

#### Sample Code

```
PLAN_TABLE  
DSN_STATEMNT_TABLE
```

```

DSN_FILTER_TABLE
DSN_FUNCTION_TABLE
DSN_DETCOST_TABLE
DSN_PGRANGE_TABLE
DSN_PGROUP_TABLE
DSN_PREDICAT_TABLE
DSN_PTASK_TABLE
DSN_QUERY_TABLE
DSN_SORTKEY_TABLE
DSN_SORT_TABLE
DSN_STRUCT_TABLE
DSN_VIEWREF_TABLE

```

For more information, see the IBM documentation on EXPLAIN tables.

The needed tables are automatically created upon the first execution of the new EXPLAIN and are included in the standard cleanup routine.

## Procedure

1. Call transaction DBACOCKPIT.
2. Choose ► *Diagnostics* ► *EXPLAIN(New Version)* ►
3. Enter the statement using the editor and choose *EXPLAIN*.

On the right-hand side of your screen, you see a graphical representation of the access path. The *Optimized Statement* tab provides the statement which has been adopted by the Db2 Optimizer.

If you click on a node in the access plan graph, several tabs on the left-hand side show information on the statement in general and the currently selected operator.

The following tabs are available:

Tab:	Description:
General	Information on the whole statement
Operator	Details and Db2 cost estimates for the currently selected operator
Catalog Information	<ul style="list-style-type: none"> <li>• The table being accessed by the operator</li> <li>• Available indexes</li> <li>• Information from the Db2 catalog including statistics</li> <li>• SAP maintenance recommendations</li> </ul>

### ⓘ Note

The tab Catalog information is only available for table and index operators.

Tab:	Description:
Predicates	<ul style="list-style-type: none"> <li>• A textual representation of the predicates being evaluated by the selected operator</li> <li>• Information on evaluation stage and filter factor estimates</li> </ul>

With the pushbutton *Optimizer's* help in the EXPLAIN Access Plan, you can decide whether the access path should be generated for parallel execution (degree= *ANY*) or for serial processing (degree = *1*). While parallel execution is chosen; for example, for Business Warehouse queries, serial processing is the default for non BW SAP workloads.

You can also select acceleration options for *EXPLAIN* which correspond to the special registers *CURRENT QUERY ACCELERATION* and *CURRENT GET\_ACCEL\_ARCHIVE*. Per default, acceleration is not considered.

The Pushbutton "Collect Data" allows the export of the whole query scenario in a zip file, which can be used to open an IBM Case for the Db2 optimizer (so-called "Service SQL"). It will open an additional window and initiate a file download to the User's workstation.

## 6.9.7 Catalog Browser (7.02 and Higher)

The catalog describes the database objects, such as tables and indexes. The catalog tables grow as database objects are added to the database. You can query and view the content of the system catalog tables.

### Procedure

1. Call transaction ► *DBACOCKPIT* ► *Diagnostics* ► *Catalog Browser* ►.
2. Enter your statement or select a statement from history.
3. Choose ► *Execute* ►.

The query result is displayed as a table on the tab *Result*. Depending on the query (for example, select of BLOBs), this table can become very wide and exceed the maximum table width. In this case, it is possible to export the table to get the complete information.

You can select how many rows you want to display in the list by entering a value in the field *Rows*.

To clear statements from the history, choose *Clear*.

To improve the readability of statements retrieved from history, it retains the statement formatting (newlines, indentations#).

Since introducing of the new *Catalog Browser* (7.02 and higher), executed statements were only remembered and thus available in the history dropdown during the current session.

A persistent statement history has been made available within the *Catalog Browser* to save statements for future sessions. For this purpose, a new input field *Query Name* has been provided. You can use it to:

- Save statements
- Load statements
- Delete statements

As statements are stored locally (in a database table), it is possible to use the saved statements for different remote systems.

## 6.9.8 Dependency Browser

The Dependency Browser provides you with an overview of the dependencies of most types of objects in the database.

### 6.9.8.1 Explain Dependency

Triggers, advanced triggers, procedures, and some functions execute static SQL queries. These queries might be explainable. For this Feature, the Dependency Browser provides a button in the statements tap if the `stmt` you are investigating in the details screen is. When using the **Explain Stmt** button, a table with all `stmts` in the package is displayed, and one `stmt` must be selected to proceed.

### 6.9.8.2 Selection Criteria

To explore the dependencies of an object, you need to provide the following information to identify the dependencies that you want to look at:

- Object name
- Object type
- Schema
- Direction

In the dropdown list for the *Direction* field, you can select whether you want to investigate the child dependencies or parent dependencies. Expressed, it is a matter of establishing “where the object is used by selecting” *UP* or “which objects are used by selecting” *DOWN*.

Start searching for dependencies of the object by choosing the *Apply* button.

If the database cannot find an object with the attributes provided, you will get an error message.

## 6.9.8.3 Dependency Tree

The Dependency Browser returns the objects and their dependencies in the Dependency Tree: a hierarchy table arranged like a tree. It shows one object per row, with the type and the schema in separate columns. Objects with children appear as nodes, which you can collapse or expand.

### Note

Limitation: Dependencies with objects in other schemas are not displayed until Version [<Version No>](#).

## Multi-Schema Support

A new feature in Version [<VersionNo>](#) is Multi-Schema Support. This feature makes it possible to explore dependencies with objects in other schemas.

## 6.9.8.4 Explore Dependencies

If you want to get more information about one object, you must double-click on the dependency objects. In principle, there are six different types of object, but only two different ways of displaying the details:

- For more information about a table, you can navigate the [Tablespace](#) screen by accessing it from the navigation bar.
- For table views, global variables, table functions, procedures, and scalar functions, you can navigate a screen that you will not see anywhere in the application: The [Dependency Details](#) screen.

### Note

The [Dependency Details](#) screen provides a unique way of getting insights about different database objects such as views. This is one of the great benefits offered by the Dependency Browser.

## Tables

The dependency details for tables are displayed on the same screen as in the [Table Space](#) section in the Navigation Tree. Please see this section of the documentation for more information about how you can get deeper insights into the selected table.

[Dependency Object Details](#) screen allows you to dig deeper into selected dependency objects. The screen consists of an ALV table, providing classical ALV functions like export and sort, and it contains nearly everything the database knows about the view. This information includes:

- What's the output of the view?
- What's the source of the view?



- Who is allowed to use or change the view?

As the table columns of the ALV reflect columns of IBM catalog tables, you can quickly consult IBM documentation to get more information about any column in the table. Further details about views can be found in the IBM documentation of DB2 catalog table `SYSIBM.SYSCOLUMNS`.

## Table Functions, Scalar Functions, Variables, Triggers, Procedures

Like views, it is also possible to get further details about other DB objects: Table Functions, Scalar Functions, Variables, Triggers, and Procedures. See the IBM documentation to get more information about the columns:

- Table Functions: `SYSIBM.SYSPARMS` catalog table
- Scalar Functions: `SYSIBM.SYSPARMS` catalog table
- Variables: `SYSIBM.SYSVARIABLES` catalog table
- Triggers: `SYSIBM.SYSTRIGGERS` catalog table
- Sequences: `SYSIBM.SYSSEQUENCES` catalog table
- Procedures: `SYSIBM.SYSPARMS` catalog table

### 6.9.8.5 Create Statement

A special feature of the Dependency Browser is the [Show Create Statement](#) button, which you can choose to navigate to the [Create Statement for](#) screen. This screen's layout is as intuitive in its design as it is useful in terms of the information it provides. The original create statement created the database object.

### 6.9.9 Sysplex Monitor

The Sysplex Monitor in the DBA Cockpit provides a quick sysplex-wide overview of the machine configuration and utilization. It provides information about the virtualization configuration and performance counters for the database host. The RMF Monitor III Data Portal for z/OS is available in a separate browser window.

#### Procedure


To start the Sysplex Monitor, proceed as follows:

1. Choose **DBA Cockpit** > **Diagnostics** > **Sysplex Monitor**.
2. Choose the CPU for which you would like to have further information. To start the RMF Monitor III Data Portal for z/OS, choose the toolbar's pushbutton **RMF Data Portal**.

## Configuration

The RMF Data Portal host and port have to be configured.

The configuration has to be done once for every Sysplex to be monitored.





Choose [► Configuration > OS Monitoring Settings](#) . The configuration panel shows the Sysplex the currently selected database subsystem belongs to.

### Configuring Data Collection for the Operating System Monitor (Transaction OS07)

You can use the operating system monitor (transaction OS07) to monitor the system resources that the operating system provides. In this way, you can identify the cause of performance problems.

#### Procedure

To enable the RMF III Data Portal for z/OS to report data to the operating system monitor, proceed as follows:

1. Select [► Configuration > OS Monitoring Settings](#) .
2. Choose [► Enable OS07n data collection](#) . The checkbox [► Enable OS07n data collection](#)  activates the data collection, and storage in the central performance history (RZ20) every five minutes. The data can be made accessible to the operating system monitor. If you choose [► Enable OS07n data collection](#)  an icon appears in the toolbar. Choose the icon "Manually start/restart data collection" (F8) to manually start the data collection process.

#### → Recommendation

Due to resource requirements, the data collection is enabled on a central monitoring system, for example, the Solution Manager.

### Configuring Sysplex Monitoring highly available with RMF Distributed Data Server (RMF DDS)

#### Background on RMF DDS

- **RMF Distributed Data Server (DDS)**  
Applications that want to access sysplex-wide performance data.  
This application can retrieve that data from a single data server on one system in the sysplex.
- **Starting the Distributed Data Server**  
The Distributed Data Server (DDS) provides the ability to serve multiple clients in a single-server address space. To have the Distributed Data Server address space managed by RMF, you automatically start it by using the DDS option. This option ensures that you always have one active instance of the Distributed Data Server within your sysplex. As soon as the RMF Sysplex Data Server recognizes the DDS option on any system in the sysplex, the Distributed Data Server is started on the RMF leading system. The leading system is a system with an active Monitor III gatherer and the highest z/OS release. You can use the `MASTER/NOMASTER parmLib` option to define a system to be or not to be a leading system candidate. If another system becomes the leading system, the Distributed Data Server is automatically restarted on this system.

For more information about RMF and how to configure RMF DDS, see the *IBM RMF User's Guide*.

#### Procedure

- **With Tivoli System Automation SA z/OS**

Exploit the latest \*SAPSRV SA z/OS policy. *Start/stop* of RMF DDS will then be managed automatically by SA z/OS.

- **Without Tivoli System Automation SA z/OS**

Follow the RMF description from *IBM RMF User's Guide* to *start/stop* RMF DDS.

In addition to the automated *Start/stop* of RMF DDS, you will need a VIPA that moves with the RMF DDS if it moves to another system in the sysplex. In this example, we use a non-disruptive distributed VIPA ( DVIPA ):

- **Setup DVIPA**

In your TCPIP profile, you need to add a VIPA range definition like this.

```
VIPARANGE DEFINE MOVEABLE NONDISRUPTIVE 255.255.255.248 9.152.20.176
```

within a `VIPADYNAMIC / ENDVIPADYNAMIC` statement.

Publish the DVIPA in your DNS with an entry like this: `9.152.20.177 <DDS name>` where `<DDS name>` is your choice.

Combined with `DOMAINORIGIN` from `/etc/resolv.conf` to the full qualified name for the RMF DDS.

You will need that in a later step in the `SAP DBACOCKPIT` configuration. So with that done, the RMF DDS can be reached by it's IP address, by it's short name `<DDS name>`, and by its long name `<DDS name.DOMAINORIGIN>`.


- **Bind together the DVIPA with the RMF DDS.**

In your TCPIP profile, you need to add a port bind statement. RMF DDS listens on `port 8803`. You will need something like:

```
8803 TCP RMFDDS01 BIND 9.152.20.177 ; VIPA FOR RMF DDS FOR SAP SYSPLEX MON.
```

To activate the changes in the TCPIP profile (DVIPA definition and bind), you have to restart TCP on each LPAR in the Sysplex.

## Configure SAP DBACOCKPIT

1.  [Logon](#) to your SAP GUI and call transaction DBACOCKPIT

2.  [Goto](#)  [Configuration](#)  [OS Monitoring Settings](#) 

3.  [Edit](#) Parameters for RMF Distributed Data Server (DDS)

Hostname = `<DDS name>` or IP address or `<DDS name.DOMAINORIGIN>`

Port = 8803

4.  [Check](#) Enable OS07n data collection.

5.  [Save](#).

A test connection will be executed. "Test connect successful" message window appears.

### Verification

Use SA z/OS to *stop/start* the APLs RMF and RMFGAT *and/or* cancel the RMF, RMFGAT and RMFDDS01 address spaces manually. During your tests, you can use SAP DBACOCKPIT Sysplex Monitor or RMF Data Portal on `<DDS name:8803>` to verify that performance data is continuously available.

### Result

Any time you stop or cancel RMF or RMFGAT or RMFDDS01, the RMF DDS will be stopped, and restarted on the RMF Master Lpar, whichever that is at that time. The VIPA will automatically move along with RMF DDS. Sysplex Monitor data is continuously available.

### More Information

- RMF Monitor III Data Portal for z/OS (For more information, see the IBM documentation.)
- SAP Notes:
  - [1308393](#): Enabling the Sysplex Monitor
  - [1084019](#): OSO7N: New Operating System Monitor

## 6.9.10 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.

However, you can create, change, and delete your Db2 commands with the authorization profiles S\_DB2\_EXPC or S\_DB2\_DBADM. You can also 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 \[page 126\]](#) in *Basic Operations*.

### Displaying the Command-List

To display the *Command list* screen, call transaction DBACOCKPIT and choose **► Diagnostics ► Db2 Commands ▾**.

### Maintaining Db2 Commands

Executing a Db2 command from the SAP system with *Db2 Commands* produces the same result as completed from the z/OS system log.

The following functions are available:

Functions for Maintaining Db2 Commands

Function	Procedure
<a href="#">Create</a>	Choose <a href="#">Create</a> . Make the necessary entries and choose <a href="#">Save</a> . <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"><p><b>Note</b></p><p>You can execute the new command in the <a href="#">Create Command</a> screen <b>without</b> saving it.</p></div>
<a href="#">Change</a>	Select command and choose <a href="#">Change</a> . Make your changes and choose <a href="#">Save</a> . You may also delete a command here. <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"><p><b>Note</b></p><p>You can execute the new command in the <a href="#">Create Command</a> screen <b>without</b> saving it.</p></div>
<a href="#">Display</a>	Select command and choose <a href="#">Display</a> .
<a href="#">Execute</a>	Select command and choose <a href="#">Execute</a> . The next screen displays the result of the command.

## 6.9.11 EXPLAIN

### Db2 EXPLAIN: Introduction

The EXPLAIN function uses Db2 EXPLAIN, which uses a special table, PLAN\_TABLE, that stores the EXPLAIN function results. If there is not a PLAN\_TABLE, the EXPLAIN function creates the following tables:

- Plantable: <SCHEMA> . PLAN\_TABLE
- Statement table: <SCHEMA> . DSN\_STATEMENT\_TABLE

These two tables reside in implicitly created UTS tablespaces.

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 STMTOKEN) from the Db2 statement cache. The advantage of selecting the statement ID or statement token is that EXPLAIN describes the access path currently stored for the statement cache statement. 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 when the catalog statistics may be different from the time when you prepared the statement.

## EXPLAIN Usage within an SAP System

There are two ways to access EXPLAIN:

### Enter the SQL statement manually:

- “Manual Option 1”
  1. Call transaction DBACOCKPIT
  2. Choose ► *Diagnostics* ► *Explain* ▾
  3. Enter the statement manually using the editor and choose *Explain*.
- “Manual Option 2”
  1. Call transaction ST05
  2. Choose *Analyze (Explain) an SQL Statement to Be Entered*
  3. In the menu, choose ► *Goto* ► *Editor for Entering SQL Statement* ▾.
  4. Enter the statement manually using the editor and choose *Explain*.

### Select an SQL statement directly from a list to be explained:

1. Call transaction DBACOCKPIT
2. Choose *Performance*
3. Choose *Thread Activity* or *Statement Cache*  
Select an SQL statement from the generated list of statements.
4. Choose *Details* or *SQL Statement*, depending on the context
5. Choose *Explain SQL statement*

For more information, see the sections [Thread Activity \[page 105\]](#) or [Cached Statement Statistics \[page 99\]](#).

If a statement **entered manually** in ST05 is explained, SAP passes the statement text to EXPLAIN. Statements from the SQL trace in transaction ST05 are usually defined using the option STMTCACHE STMTOKEN.

If a statement is **selected directly from a list** of SQL statements, SAP uses the EXPLAIN option STMTCACHE STMTID.

As these two options only work if a statement is still in the dynamic statement cache of Db2, the SAP system detects such situations. It automatically invokes Db2 EXPLAIN, providing the statement text. On the bottom of the panel that displays EXPLAIN's result, it is indicated if the displayed access path 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, a free random number is chosen. For the other options, Db2 inserts the statement ID as a query number in the PLAN\_TABLE. If a statement is explained using the statement token, the STMTOKEN field of the PLAN\_TABLE also contains the statement's token. The SAP system always deletes the rows in the PLAN\_TABLE that would interfere with a statement to be explained.

There are two views on the PLAN\_TABLE:

- Standard view (PLAN\_TABLE data presented in a hierarchical view)
- Expert view (the content of the PLAN\_TABLE)

### Note

If `EXPLAIN` is invoked with the statement text, you can dynamically turn on parallelism to explain each SQL statement's access path.

## Troubleshooting

If you have problems using `EXPLAIN` within the SAP system (duplicate records, invalid `SELECT` statement when reading the `PLAN_TABLE`), SQL errors while invoking Explain, the `<SCHEMA>.PLAN_TABLE` is probably not defined correctly.

You can solve this problem by dropping the `PLAN_TABLE` and/or the other Db2 `EXPLAIN_TABLES`. Then retry the `EXPLAIN` within the SAP system.

### Note

The classic explanation will only recreate the `PLAN_TABLE` and the `DSN_STATEMENT_TABLE` automatically. In case of persistent problems with wrong `EXPLAIN` table formats, make sure to have the current SAP Note [1731268](#) implemented in the system.

With SAP Note [2266420](#), SAP introduces the usage of Db2 supplied Stored Procedure `ADMIN_EXPLAIN_MAINT` to maintain the `EXPLAIN` tables. From this point on, SAP relies on the Stored Procedure to create `EXPLAIN_TABLES` matching the current Db2 release and service level.

## SAP NetWeaver 7.02, 7.30 and Higher

[Web Dynpro EXPLAIN \(Extended Version\) \[page 220\]](#)

### 6.9.12 PTF Status

Since the PTF status of a Db2 subsystem is not displayed, and helps you keep your system at the latest maintenance level, the database monitor provides a diagnostic function.

You can access this function as follows:

1. Call transaction `DBACOCKPIT`
2. Choose **► Diagnostics ► PTF Status ►**

In the background, the Db2 utility `DIAGNOSIS` is called via the stored procedure `DSNUTILS`.

The resulting module entry point list called *Diagnose Display MEPL* is displayed in raw format.

## 6.9.12.1 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 Notes [81737](#) and [364109](#)
3. Determination of the status of all required PTFs within the z/OS system
4. The output of missing PTFs and/or **Function Module IDs (FMID)**

## 6.9.12.2 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.
- SAP Note [364109](#) lists all PTFs related to the latest Db2 put levels.

## 6.9.12.3 Technical Details

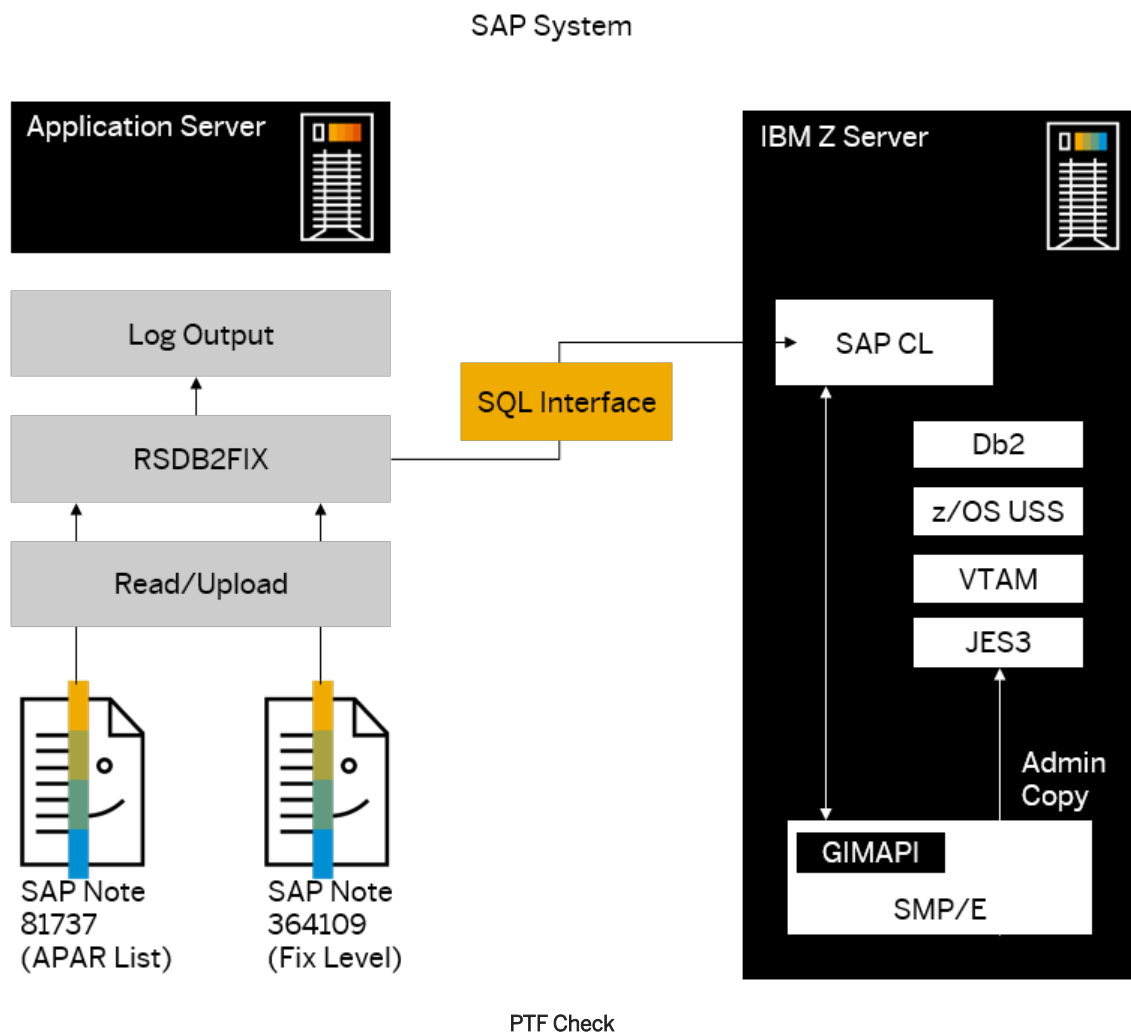
The automated PTF check is based on the assumption that customers administer all z/OS software components using IBM's **System Modification Program Extended (SMP/E)**. This z/OS program keeps a record of all changes (for example, PTFs) to function modules in the **Consolidated Software Inventory (CSI)**.

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

The stored procedure **SAPCL** – which is delivered with the SAP software – can connect to the **GIMAPI** interface and forward the SMP/E data to the SAP System. The PTF check itself is performed by report **RSDB2FIX** and runs on the SAP System.



The technical details of the PTF check are illustrated in the following figure:



There are two sources of input for the check tool:

- SAP Note [364109](#) contains the current fix level which is a list of all PTFs related to the latest Db2 put levels.
- SAP Note [81737](#) lists all PTFs required for Db2 (on top of the put levels), and all other z/OS-related products.

Both SAP Notes are formatted in such a way that they can be used directly as input for the check report. If a valid RFC connection exists, the tool is able to retrieve the most recent versions of both SAP Notes [81737](#) and [364109](#) from the SAP Support Portal. Alternatively, you can also download them to your PC and use the download files as input for `RSDB2FIX`.

The SAP systems kernel release as well as the versions of the Db2 and z/OS software used are determined by the PTF check tool. This information combined with the uploaded PTF information (SAP Notes [81737](#) and [364109](#)) is subsequently used by `RSDB2FIX` to retrieve a list of required PTFs and FMIDs.

Finally, the tool queries the status of each required FMID and PTF employing `SAPCL` 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.

## 6.9.12.4 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 precisely the software's status that is running.
2. Update the check report `RSDB2FIX` in your SAP system if necessary. For more information, see SAP Note [183311](#).
3. Setup the stored procedure `SAPCL`  
For more information, see [Installing Stored Procedure SAPCL \[page 128\]](#).
4. Establish a connection to SAPNet – R/3 Frontend. (Optional)
  1. Log on to the check system and call transaction `OSS1`.
  2. Choose **► Parameters ► Technical Settings ►**.
  3. Specify and save *Logon* settings.
  4. Choose *Logon*, specify group `1_PUBLIC`, and check whether the connection works.


## 6.9.12.5 Performing the Check

Once you have completed the preparations described in section [PTF Check Setup \[page 234\]](#), you can perform the PTF check.

1. If the SAP system is not able to connect to SAPNet, you need to transfer SAP Notes [81737](#) and [364109](#) to your PC:
  1. Display the English version of SAP Note [81737](#) in the SAP Support Portal at <http://support.sap.com/notes>
  2. Choose *Download*.
  3. Repeat steps a and b for SAP Note [364109](#).
  4. Start the SAP Download Manager and download the two SAP Notes to your PC.  
The default file names are `NOTE_0000081737` and `NOTE_0000364109`.
2. To access the PTF check tool, call transaction `DBACOCKPIT` and choose **► Diagnostics ► Missing PTFs ►**.  
The input screen of the check report `RSDB2FIX` appears.
3. Enter the following input values:

Fix Check: Input Screen Values

File Name or Setting	Input Value/Description
<a href="#">SAP Note 81737</a> (Required if the SAPOSS connection does not exist.)	The full path name of the file on the PC that contains SAP Note <a href="#">81737</a>

File Name or Setting	Input Value/Description
<i>Fix level file (Optional)</i>	The full path name of the file on the PC that contains SAP Note <a href="#">364109</a> 
<i>Log Name</i>	Name of the log to which the output is written. The pattern &R3&, &DATE&, and &TIME& are substituted by the name of the SAP system, the date, and the time.
<i>SAPNet</i>	SAPNet RFC destination. The default connection <i>SAPOSS</i> is created by using transaction OSS1 (see <a href="#">PTF Check Setup [page 234]</a> ). You can access the transaction directly by choosing the <i>Online Service System (OSS1)</i> .
<i>SMP/E Settings</i>	<p>The input depends on how SMP/E is configured in your environment. Specify the <i>data set</i> and the <i>zone</i> for at least one <i>CSI library</i>. The sample input in the <i>Fix Check: Input Screen</i> graphic represents an environment where all the Db2 function modules are administered in data set SYS1.DB2810.CSI and zone TDB281, whereas the remaining software components (z/OS USS, JES3, VTAM, and so on) are kept in data set SYS4.S160524.CSI and zone T160524.</p> <p>You cannot check GLOBAL or DLIB zones.</p>

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). You can easily repeat the PTF check.
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.
7. Execute report *RSDB2FIX* online. The resulting log is displayed directly after the check.
8. You can display all result logs by choosing *Display logs* within the initial selection screen of *RSDB2FIX*.

## 6.9.12.6 Output Analysis

The *RSDB2FIX* program reports all errors, warnings and check results 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.
  - *The following PTFs are missing.*  
A list providing information on the missing PTFs and their associated FMID and APAR is given. FMID and APAR order the list.  
Check the status of these missing PTFs. Maybe they are only needed under specific 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.
  - *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 because none of the associated FMIDs is active in the given SMP/E settings. The APAR may refers to a product that is not installed in your environment. For example, this is 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 return `RSDB2FIX`.

## 6.9.12.7 PTF Check Recommendations

### Scheduling PTF Checks Regularly

Consider defining a background job within transaction `SM36` that checks all your systems regularly, for instance, once a month. Each target system 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 systems SAP system release. Therefore, use `RSDB2FIX` before upgrading a system to check whether all PTFs required for the target release are applied.

## 6.9.12.8 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 the current version. See SAP Note 183311 for details.*  
A hot package is provided.
- Error message: *Version of SAP Note 0081737 is outdated. Please use the 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.

## 6.9.13 DB2 Message Log

Procedure

The *DB2 Message Log* displays the DB2 console messages of the monitored DB2 subsystem provided by IFCID 197. Therefore the DB2 Messages can be accessed via the DBA Cockpit instead of logging on to ISPF (3270).

It provides the option to filter by a specific time range. As the message texts are truncated in the overview table (due to the 128 characters limitation of ALV displays) a detailed viewer was introduced to display the complete message text. You can access the detailed view by clicking one of the links in the overview table to select a specific entry.

For more information, see [1327401](#).

1. Call transaction DBACOCKPIT.
2. Choose ► *Diagnostics* ► *DB2 Message Log* ▾.

## 6.9.14 z/OS System Log

You can display the *z/OS System Log* by calling transaction DBACOCKPIT and choosing the ► *Diagnostics* ► *z/OS System Log* ▾.

The *z/OS System Log Viewer* can show the z/OS System Log by the z/OS image running the current Db2 instance. If a Sysplex is used, the z/OS System Log from all participating z/OS Images is listed.

The content is similar to the output of the SD;LOG TSO-command.

### Prerequisites

The *z/OS System Log Viewer* is based on the Stored Procedure ADMIN\_INFO\_SYSLOG.

To successfully execute the Stored Procedure

- the procedure needs to run within an address space established by WLM with TCB=1
- the user calling this procedure needs to have access to the SDSF ISFEXEC TSO-command MAS
- z/OS Version 1 Release 12 or higher is required.

The following *Selection Criteria* are available:

- *From Date/Time*: Specify the starting date and time (default: 5 minutes ago)
- *To Date/Time*: Specify the ending date and time (default: current time)
- *Filter*: Enter a search string (default: empty)
- *z/OS LPAR*: To select all Sysplex members, use the drop-down option "ALL". Otherwise select one of the LPARs hosting an instance of the monitored Db2 data sharing group system. (Default in case of data sharing group is the data sharing member, the application server is connected to)
- *Maximal entries*: maximum number of log entries to be listed (default: up to 2000 entries)

To change the default settings, specify new criteria, and press the *Apply Criteria* button.

### Note

By using the *F8* button, you can to display the content of the last five minutes. The entered *Selection Criteria* are being applied.

The following table summarizes the columns displayed in the output list.

Column Name	Column Definition	Possible Entries
MSG	Record Type (the first character) and Request Type (second character)	Record Types: <ul style="list-style-type: none"> <li>• n: Normal Line Message</li> <li>• m: Multiline Message (To see complete message, double click on <a href="#">Date or Time</a>.)</li> <li>• x: Unknown Record</li> </ul> Request Type: <ul style="list-style-type: none"> <li>• c: Command</li> <li>• r: Response to command</li> </ul>
SSID	z/OS LPAR Name	
Date	Date of the log entry	
Time	Time of the log entry	
Routing Code	Routing Code	
Origin	Origin of log message	For example CONSOLE, Subsystem Name
User Flags	User Flags	
Message	Message optional with ID	Freeform field filled by log generator

As the message texts are truncated in the overview table (due to the 128 characters limitation of ALV displays), a detailed viewer was introduced to display the complete message text. This is especially helpful when monitoring multi-line-messages. You can access the detailed view by clicking one of the links in the overview table in order to select a specific entry.

# 7 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 hardware, software, operational or environmental error or outage. These processes are a crucial factor in system availability and reliability. They need careful assessment of their requirements, understanding of the processes, and skillful planning, development, and procedures.

Some of these processes are performed automatically by Db2 without outside intervention, such as recovering the SAP database to its consistent state just before a z/OS system crash or 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 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.

### Note

An SAP database includes all the tablespaces, indexes, and Db2 catalog and directory entries (practically all the catalog and directory tablespaces and indexes) pertinent to the SAP system. With MCOB, the set of SAP databases that share a Db2 subsystem or data sharing group form an MCOB landscape. From the viewpoint of operational and semantic integrity, an SAP database or MCOB 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 of when the entire SAP database or MCOB landscape might need to be recovered. A recovery to the current state is 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 MCOB 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.
  - Consider producing multiple backup copies.
  - Keep back-level backups to extend the interval within which it is possible to perform a prior point in time recovery. This also reduces the impact of backup data sets that might be damaged (inconsistent).
  - 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).

The following sections present a summary of backup and recovery options used in the SAP system environment.

For more information, see also the following IBM documentation:

- *Db2 for z/OS Administration Guide section Operation and Recovery*
- *Db2 for z/OS Utility Guide and Reference*
- *Db2 for z/OS Command Reference*
- *Db2 for z/OS Data Sharing: Planning and Administration*

The *Casebook 2017 Edition: Tightly Integrated Db2 Backup, Recovery and Cloning for SAP Environments* provides detailed hands-on explanations for various typical backup, recovery, and cloning use cases with SAP.

The IBM documentation *Business Continuity for SAP on IBM Z* discusses disaster recovery that goes hand in hand with the Db2 backup and recovery approach discussed here.

## 7.1 Backup

The appropriate backup procedure is a critical 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. All changed pages were reflected on disk before the backup was taken, and the recovery time. 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 the one hand and the recovery speed. The shorter the log apply phase, the faster the recovery. Frequent backups of all of the data pages containing committed rows provide the most rapid recoveries. However, this has a higher cost in resources and impedes the system's



concurrent activity to the extent that it might not be acceptable. There are two main types of backup. In this documentation, these are referred to as **online backup** and **offline backup**. Since SAP solutions on IBM Z typically cannot tolerate much downtime, we focus on the online backup approach.

## 7.2 Online Backup

An online backup of an object (tablespace, partition, index, or volume) is a copy of the object. During this 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:

- Volume-level
- Object-level

In SAP environments, volume-level online backups are generally recommended as the primary backup approach as it is more automated and requires fewer CPU cycles on IBM Z.

### Volume-Level Online Backup

Volume-level backups are copies of the volumes on which Db2 objects reside.

As these backups are created at the disk level, they consume negligible CPU resources on IBM Z.

This type of backup requires the availability of disk subsystems capable of generating high-speed volume copies, such as IBM DS8000, EMC, or HDS.

You can capture volume-level backups in two ways:

- Under the control of Db2
- Outside the control of Db2 while suspending Db2 log activity

The Db2 utility `BACKUP SYSTEM` invokes z/OS DFSMSHsm fast replication services to take volume-level backups of all volumes on which Db2 objects reside.

This method provides a very efficient and non-disruptive means of taking online backups of the entire SAP database.

The `Backup System` utility is positioned as a single point of control to manage virtually all aspects of backups.

A prerequisite for this method is to have a disk subsystem that supports 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, you can use normal Db2 restart recovery to recover the system to the point in time of the backup copy. This kind of recovery is speedy but is limited to the issues when backups were taken.

Also, the `DUMP` option allows you to dump backups to tape additionally. Using the `DUMPCCLASS` parameter, you can control the DFSMSHsm dump class used for dump processing. The `DUMPOONLY` option enables you to dump an existing backup from disk to tape. The new options, `ESTABLISH FCINCREMENTAL`, and `END FCINCREMENTAL` serves to take incremental FlashCopy data-only backup. When the `BACKUP SYSTEM` is invoked with the `ESTABLISH FCINCREMENTAL` option, the incremental FlashCopy relationship is established. This creates a full backup. At this point, change recording is started within the disk subsystem. Every subsequent invocation of `BACKUP SYSTEM`, without any option, results in copying the changed tracks from the source volume to the initial full FlashCopy backup. Incremental FlashCopy hence overrides the initial backup. `BACKUP SYSTEM` with the `END FCINCREMENTAL` option takes the last incremental FlashCopy backup and completes the incremental FlashCopy relationships. FlashCopy backups dumped to tape are always full backups on tape - even if the backup on disk is incremental. The advantages of incremental FlashCopy backups are that they require less time to physically copy the volumes and that the I/O impact on the disk subsystem is minimized.

### Procedures for Making Volume-Level Backups

The procedure for making volume-level backups under the control of Db2 is to invoke

```
BACKUP SYSTEM
```

To prepare the backups so that you can recover subsequent single tables from the volume-level backup, you should exploit the z/OS DFSMSHsm copy catalog feature (new with z/OS 1.11) by updating the configuration of the DFSMSHsm copy pools used by Db2. This enables recovery at table level from any system-level backups to any point in time after system-level backup is allowed – even if the table was later reorganized or moved to another volume by native storage means. However, keep in mind that there needs to be sufficient free space on the volume, where a dataset resides at the backup time to enable the dataset to be recovered from the volume-level backup.

You can take volume-level backups outside the control of Db2 and couple with the Db2 feature to suspend and resume Db2 logging activity; this approach is also very efficient on-demand. However, it is more complex and requires stopping the Db2 log.

The procedure for making volume-level backups outside of Db2 is outlined below:

1. 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 other 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.

#### Note

Although the `SET LOG SUSPEND` command initiates a checkpoint, it does not wait for all the data pages to be written out. Thus, you can improve subsequent recoveries' performance by minimizing the number of data pages that are not written out when the disk volume copies are taken. You can do this by triggering a checkpoint five to ten minutes earlier. An on-demand checkpoint is triggered by issuing the Db2 command `SET LOG LOGLOAD(0)`.

2. Take disk volume copies of all the volumes containing Db2 user and system data  
You should do this step 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. You can use the system backup obtained by this for starting

Db2. This start will be performed in the same way as Db2 restart after an abnormal system termination. You will roll the inflight recovery units, which brings the SAP database to a consistent state.

3. 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.

## Object-Level Online Backup

Object-level 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 backup type. The utility generates backups of Db2 tablespaces, partitions, and indexes.

The following table lists other useful `COPY` options:

Option	Description
FULL	Specifies whether a full or incremental image copy is to be created. The total image copy is a copy of only those data pages that have been changed since the last backup.
CHANGELIMIT	It 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 <code>CHANGELIMIT</code> specification, where only one value is specified, to avoid the situation where no full image copy exists.
COPYDDN RECOVERYDDN	Allows you to create up to four identical copies of the table-space
CHECKPAGE	Checks page consistency within a page and makes subsequent <code>DSN1COPY</code> checks superfluous.  Note that <code>CHECKPAGE</code> does not do the checks performed by <code>CHECK INDEX</code> and <code>DSN1CHKR</code> (for tablespaces that contain internal links).
PARALLEL	Can significantly improve <code>COPY</code> performance by parallel copying of objects specified in the same <code>COPY</code> control statement

Option	Description
FLASHCOPY	<p>With this option, the COPY utility calls dataset-level FlashCopy to copy the data set. As a result, the copy process does not require IBM Z CPU cycles and is performed very quickly. However, FlashCopy image copies are always full. There are two types: FLASHCOPY(YES) takes a dataset-level FlashCopy as is. Therefore it can contain uncommitted data that will be rolled back during recovery. The other type is FLASHCOPY(CONSISTENT). In this case, the COPY utility first takes a dataset-level FlashCopy and then backs out uncommitted data from the FlashCopy backups. You can combine this with running the Db2 utilities in SHRLEVEL CHANGE mode, resulting in a consistent backup being created from an online backup.</p>

## 7.3 Backup Procedure Recommendations

For each SAP system installation, you should choose a backup procedure optimal to its particular needs and conditions.

### Recommendations for a Backup Procedure

The following sections 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.

#### Note

Producing regular online backups is a mandatory step of any backup procedure.

### Make an Offline Backup Before an SAP System Installation, Upgrade, etc. (Mandatory)

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.

## Set the Frequency of Regular Online Backups (Mandatory)

For any SAP system, catalog and directory tablespace regularly create online backups. How often you should take the backup and whether it is full or incremental, depends on the tablespace's change rate.

If you take traditional image copies as a starting point, we recommend that you run the backup jobs every 1-2 days and specify `CHANGELIMIT(10)`. It would help if you relied on the image copy recommendations of stored procedure `DSNACCOX` to identify the objects that need new incremental or full image copies.

### Note

With the `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 `FULL(YES)` or `CHANGELIMIT(0)`. Also, consider running the `MERGECOPY` utility that consolidates a complete backup and many incremental backups into a new, more recent full backup.

You can significantly improve the index recovery time if the recovery is based on the index copy rather than on the index rebuild. Therefore, it is recommended to copy at least large indexes as well.

## Use the BACKUP SYSTEM Utility

Where available, use the disk system capabilities of generating fast volume copies, ideally using the `BACKUP SYSTEM` utility to produce fast online system backups.

If produced when there are no long-running, non-committing jobs in the system, these backups provide for a very efficient prior point in time recoveries.

If using `BACKUP SYSTEM` logs, BSDS and associated ICF catalogs need to reside in the same copy pool. Likewise, the Db2 data, catalog/directory, and related ICF catalogs also need to reside in the same copy pool. If you take the volume-level backups outside the control of Db2, 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. This ensures the highest flexibility in recovery scenarios.

The `RECOVER` utility is capable of recovering an individual object based on volume-level backups taken by `BACKUP SYSTEM`. Hence, this feature relieves you from the burden of taking image copies regularly. The only cases in which image copies are still needed is for two reasons:

- Not logged Db2 operations like online `REORG` or `LOAD` utility with the `LOG NO` option. Therefore, it is key to always take inline-image copies for these utilities.
- If the object has moved since the volume-level backup was taken – for example, by online `REORG` or when data sets were moved to different volumes (for instance, to optimize the data set to volume mapping or consolidate volumes). To avoid that Db2 utilities delete and redefine data sets, use the `REUSE` option of the utilities where possible.

Prior to z/OS 1.11, the z/OS DFSMSHsm fast replication services, which Db2 relies on, could not recover individual data sets from volume-level backups whenever the data sets were moved to a different volume in the meantime. The reason was that the fast replication services consulted the ICF catalogs to determine the relevant volume for a data set. You retrieved this information at the time when you scheduled recovery. To address this gap, z/OS 1.11 introduced the copy catalog feature for the DFSMSHsm fast replication services. They would enable the restoration of individual data sets from the volume-level backup even if you moved them

to a different volume. The only consideration left is that there needs to be sufficient space on the volume on which the data set resided at the backup time. If it turns out that this is not the case, you may move data sets currently on that volume to other volumes and then restart object-level recovery.

## Considerations Regarding the BACKUP SYSTEM Utility

The following are additional considerations if you rely on `BACKUP SYSTEM`.

For housekeeping purposes, we recommend that you do the following:

- Periodically run the `MODIFY RECOVERY` to clean up obsolete entries in `SYSIBM.SYSCOPY` and `SYSIBM.SYSLGRNX`. This utility checks both `SYSIBM.SYSCOPY` and the Db2 bootstrap data set make sure there is either a system-level backup or an image copy available for a tablespace. Only if neither of these exists will the utility then put the tablespace into `COPYP` status.
- You can use the `DB2 REPORT RECOVERY` utility to determine whether there are system-level backups or image copies available for a certain tablespace or index.

### Note

The SAP DBA Cockpit can be used to display this information.

- Set up the automatic dump function of `DFSMSHSM` to scratch expired tape backups
- Volumes should usually not be removed from a copy pool. Otherwise, you will have to restore the volumes that were deleted during recovery manually. The deleted volumes must then be synchronized with `RESTORE SYSTEM`, as well. If volumes are removed, ensure that they are not reused until all the copy pool backup copies that require the volumes have rolled-off.
- Db2 can automatically clean up the CF structures during the first Db2 restart after system-level recovery to facilitate the system-level point-in-time recovery in data sharing environments. This is controlled by the Db2 system parameter `DEL_CFSTRUCTS_ON_RESTART`. For an SAP environment, we generally recommend setting this parameter to `NO` to enable a faster Db2 restart. Before performing a system-level recovery using `RESTORE SYSTEM`, we recommend setting this system parameter to `YES` to clean up the CF structures since they no longer match the Db2 data.

You may take the following periodic checks to make sure all required backups have been successfully taken:

- In `SYSIBM.SYSCOPY`, check whether an (inline) image copy is available for each online `REORG` and `LOAD`.
- Issue the `DFSMSHSM` command `LIST COPYPOOL SELECT (FRSTATE(FAILED))` to check whether a FlashCopy backup did not complete successfully.
- Issue the `DFSMSHSM` command `LIST COPYPOOL SELECT (DUMPSTATE(PARTIAL))` to list the dump versions that did not complete successfully.
- To check the status of the physical background copy of FlashCopy, issue one of the following:
  - TSO `FCQUERY` command
  - HSM `QUERY COPYPOOL` command

### Note

Both commands only show the copy status while the physical background copy is still ongoing.

## Considerations When Primarily Relying on Image Copies

If you do not rely on the `BACKUP SYSTEM` utility, you need to regularly create online image copies for any SAP system, catalog, and directory tablespace.

How often you should take the backup 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 `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. It is strongly recommended that you rely on stored procedure `DSNACCOX` to identify the objects that should be backed up.

For full image copies, you should consider taking advantage of FlashCopy image copy (FCIC). If you specify the option `FLASHCOPY(YES)` or `FLASHCOPY(CONSISTENT)`, the `COPY` utility invokes data set-level FlashCopy to copy the data sets object efficiently. This requires FlashCopy Version 2. This approach's advantages are that no CPU resources on IBM Z are required to copy the data and that the recovery can be faster. You should use `FLASHCOPY(YES)` since this is sufficient to copy the data and require less effort to take backups. During recovery, the uncommitted data is then rolled back. To consistently clone a set of tables, using the `FLASHCOPY(CONSISTENT)` option can be an attractive option.

### Note

With the `CHANGELIMIT` option, you might end up have full backups created often enough, which is not efficient from the recovery point of view. For this reason, make sure you have a complete backup created periodically by specifying `FULL(YES)` or `CHANGELIMIT(0)`. Consider running the `MERGECOPY` utility that consolidates a full backup and many of incremental backups into a new, more recent full backup.

You can significantly improve the index recovery time if the recovery is based on the index copy rather than on the index rebuild. Therefore, it is recommended to copy at least large indexes as well.

## 7.4 Combining DB2 BACKUP SYSTEM, DS8000 FlashCopy, DS8000 Metro Mirror, and GDPS

`BACKUP SYSTEM` is non-disruptive. Therefore the Metro Mirror relationship between the primary and secondary sites does not need to be split. At the primary site, you can take volume-based copies at any time with the `BACKUP SYSTEM` utility. Due to the Metro Mirror secondary status of volumes at the secondary site, you cannot take the copies there. The Copy Pool Backup storage group, which contains the backup target volumes, can be mirrored to the secondary site using Metro Mirror to have the backups available at both the primary and secondary sites.

To seamlessly combine the FlashCopy and Metro Mirror technologies to have a single volume, to be the target of a FlashCopy operation at the primary site and, at the same time, the source of a Metro Mirror relationship. We recommend exploiting the DS8000 Remote Pair FlashCopy (RPFC). This prevents duplex-pending situations for volumes while the physical background copy of FlashCopy is going on. Duplex-pending volumes may prevent Geographically Dispersed Parallel Sysplex (GDPS) from triggering Failover. With RPFC, this issue is resolved. Hence, you can design symmetric primary and secondary locations.

To enforce that all FlashCopy backups are taken with RPFC technology. Simultaneously, the volumes always remain in full-duplex mode. You should specify the Preserve Mirror Required (PMR) option in the Data Facility

Storage Management Subsystem (DFSMSdss). To enforce this at the enterprise level, you can set PMR in the DFSMSdss installation exit `ADRUIXIT`. You must also apply the APAR `0A34143`. For a more granular solution on Db2 for z/OS subsystem granularity, you can set the Db2 system parameter `FLASHCOPY_PPRC`.

## 7.5 Recovery

The Db2 utilities `RECOVER` and `RESTORE SYSTEM` is used to recover data.

`RECOVER` allows you to recover Db2 objects at the following levels:

- tablespaces
- indexes
- partitions
- individual data sets
- individual pages

On the other hand, `RESTORE SYSTEM` 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 (options `TOCOPY`, `TOLASTCOPY`, `TOLASTFULLCOPY`), which does not ensure consistency – for example, this state may contain uncommitted data.
- 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. Recovering this way always establishes consistency.
- 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 created outside of the Db2 control (for example, storage subsystem fast copy capabilities).

### RECOVER Utility

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 applies 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 sufficient for most point in time recoveries.

You can recover indexes 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).

### Recovering Single Object from Volume-Level Backup



The `RECOVER` utility can recover an individual object using a volume-level backup taken by `BACKUP SYSTEM` as a recovery base. This capability allows you to use volume-level backups as the only regular backup mechanism. To enable it, specify **YES** for the Db2 system parameter `SYSTEM_LEVEL_BACKUPS`. The `RECOVER` utility then automatically takes the best backup – be it an image copy or volume-level backup – as a recovery base depending on the available backups and the recovery target point.

## Backward Recovery

Sometimes single SAP tables need to be recovered to a prior point in time, and the last backup is relatively old. For example, this can happen when a faulty transport has been applied that deleted too much data in a table. The recovery can then be slow since Db2 is always used to apply log records. Db2 allows you to perform a point-in-time recovery by rolling back the Db2 log instead of restoring a recovery base and rolling forward. The `BACKOUT` option of the `RECOVER` utility needs to be specified to enable this. The benefits of backward recovery are less downtime for specific scenarios and an additional safety net for recovery that does not require a backup.

## 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. The `FROMDUMP` option allows you to specify that you should consider only backups on tape. You can use the Db2 system parameter `RESTORE_RECOVER_FROMDUMP` to set this option by default. If this option is not specified, `RESTORE SYSTEM` only takes the backup on disk into consideration.

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.

If the backup resides on tape, restore the copy pools by the **DFSMSdss RESTORE** command.

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.

## 7.6 Recovery to the Current State

A recovery to the **current state** is generally less demanding and usually needed more often than a **prior time** recovery point.

A typical example of current state recovery is a DASD volume failure that resulted in a loss of all or some of the volume data. In this case, the procedure 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 are known to Db2 (taken by the `COPY` utility), or you can use those unregistered in Db2 (such as the volume-based backups). The choice depends on the recovery case. For example, suppose the entire volume needs to be replaced. In that case, 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.

For current state recovery, backups are known to Db2 (taken by the `BACKUP SYSTEM` or the `COPY` utility), or you can use those unregistered in Db2 (such as the volume-based backups). The choice depends on the recovery case. For example, suppose the entire volume needs to be replaced. In that case, as the basis for the `RECOVER LOGONLY` process, a volume-based backup 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.

The `RESTORE SYSTEM` utility allows you to perform system-level recovery to the current state by performing a conditional restart in `SYSPITR` mode with no log truncation through the specification of the value `FFFFFFFFFFFF` for `SYSPITR`.

## 7.7 Recovery to a Prior Point in Time

This type of recovery is used to reinstate the SAP database at some previous point in time. You will lose all the changes that occurred after that time, 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

When you might need a prior point in time recovery, a typical situation is an application program logic error that introduced unwanted changes into the system that could not be “reverse engineered”. In some cases, you can avoid the prior point in time recovery and the loss of data associated with it 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 any prior point in time using object-based backups
- Recovery to the state at the time you created a volume-based online backup of the SAP database.
- Recovery to any prior point in time using volume-based backups

### Note

Common to most of these techniques is that you cannot recover the SAP data stored in Db2 (for example, SAP application-based archived data) to the same prior point in time. For SAP ABAP applications, that is not a problem in principle as this data is not considered a recoverable resource from the database perspective. Note that there are SAP Java applications that also store data in the file system, which needs to be recovered to the same point in time. Check the SAP Java documentation for your specific application to find out whether this applies to your case.

## Improving Recovery Performance

You can speed up the methods 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, you should stop the SAP system and access to Db2 either restricted to recover jobs only (by specifying `START DB2 ACCESS(MAINT)`) or wholly 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, before the point when the system got damaged
- Availability of offline backups
- Whether you included index spaces in the backup
- Availability of quiesce points

The recovery methods are described in more detail in the following sections.

### Recovery to Any Prior Point in Time Using Object-Based Online Backups

This recovery method uses the conditional restart technique. It is the least obstructive to everyday operations in creating all the prerequisites for a prior point in the SAP database's time recovery.

The method's main characteristics are that neither offline backups nor quiesce points need to be provided, making it the prime choice in 24 x 7 SAP system environments. It can also bring the system closest to when the SAP database is semantically and operationally consistent.

This recovery method assumes that a set of valid object-based backups (the `COPY` utility took, for example, tablespace, partitions, and index image copies) is available.

### Note

This recovery method assumes that a set of valid object-based backups (the `COPY` utility took, for example, tablespace, partitions, and index image copies) 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 summarize the most important points and considerations your recovery procedure will need to consider.

1. To determine which RBA or LRSN approximates the time  $T$  you want to bring the system back. 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.
  - 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 are 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 is completed) to the currency.  
In addition to these, you need to recover the objects that were `REORGED` 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 you can ignore the target RBA from the consistency viewpoint, but you might want to identify them as well to delete (`AMS DELETE`) the corresponding orphan data sets. You can identify the `REORGED` objects by selecting the matching `SYS COPY` rows before the catalog is recovered to the target RBA.  
You can find the objects that were dropped or created since the target RBA by matching the result of `SELECT` of all the rows from `SYSTABLESPACE` and `SYSINDEXSPACE` with the corresponding underlying data sets in the ICF. It would help if you did the `SELECT` 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 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 an 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 you will not start all the objects in the activated state at the target RBA at the next Db2 start; in other words, it will not go through the normal restart process. However, that `DEFER` does not affect the log during restart; in other words, Db2 still processes the appropriate log range. Still, the logged operations are not applied to the deferred start data sets.  
During the start, you might place some pages in the `LPL/GRECP`. You will remove these pages from the `LPL/GRECP` in 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 an 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* section *Recovering Catalog and Directory Objects* `RECOVER TABLESPACE`.

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 standard restart processing during the subsequent subsystem starts.
6. It is possible that some of the transactions that use an asynchronous update protocol are not fully rolled back after the system recovery, but they can be identified by transaction `SM13`, and appropriate action is 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.

### Note

In some cases that should never be exercised without in-depth expertise in the affected SAP 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, you cannot use the conditional restart method 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.

Using the `TORBA` or the `TLOGPOINT` option of the `RECOVER` utility, you can consistently recover the tablespaces to the desired point in time. The compensating transactions are executed directly in 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 usually started. You will perform this start as in the case of Db2 restart after an abnormal system termination. The inflight recovery units will be rolled back, which brings the SAP database to a consistent state.

After Db2 comes up, you can use AMS to reform the volumes added after the recovery point. This does not affect the 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. You can accomplish this by setting the Db2 system parameter `DEL_CFSSTRUCTS_ON_RESTART` to `YES` or executing 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. However, you must be sensitive to at what time such a volume-based backup is taken. You should avoid doing so during long-running units of recovery. When you use this method, you can only recover the system to this specific point when you took the backups. 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 backup taken by the `BACKUP SYSTEM` utility or recovers by processing the log, assuming that you manually restored volumes. 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 consider.

1. Determine which RBA or LRSN approximates the time to which  $T$  you want to bring the system back. You can directly specify a timestamp using the new `SYSPITR` option of the `DSNJU003` utility. 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.
  - 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  $T$ . 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 when the system is to be recovered. For data sharing systems, repeat this step on all members specifying the same LRSN truncation point.

#### Note

The `ENDLRSN` option can be used in both data sharing and non-data sharing environments. It allows you to set the exact log point. In non-data sharing environments, `ENDLRSN` expects an RBA value.

3. In data sharing, delete CF structures.
4. Start Db2. In data sharing, start all active members.
5. Submit the `RESTORE SYSTEM` utility job.

Unless there are specific reasons, let `RESTORE SYSTEM` automatically choose the best backup as a recovery base from disk and tape.

If the backup has already been manually restored, use the `LOGONLY` option.
6. Recover objects marked in recover pending state and rebuild objects 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. Still, they can be identified by transaction `SM13`, and appropriate action is 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.

## 7.8 Federated Recovery to a Prior Point in Time

If multiple SAP systems logically belong together and there arises the need to perform a point in time recovery due to application errors, all related SAP systems would need to be recovered to ensure that the systems remain consistent. One approach that implicitly allows a federated point in time recovery is consolidating all related SAP systems in the same database using `MCOD`.

If you do not run an `MCOD` configuration, you can still perform a federated point-in-time recovery if all Db2 subsystems run on the same Sysplex and use the time source. Db2 allows you to specify the clause `ENDTIME` for the `DSNJU003` utility, which serves to identify the `RESTORE SYSTEM` utility's recovery target point.

As the timestamp used for the Db2 log entries all rely on the same server time protocol, you will be able to perform a federated recovery to any prior point in time after carrying out the following steps:

1. Specify the same timestamp for `DSNJU003` on all affected Db2 subsystems, which do not need to be part of the same data sharing group
2. Subsequently, run `RESTORE SYSTEM` at each Db2 subsystem or data sharing group.

# 8 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 for 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. You can do this by collecting and storing the performance data over a more extended period, 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 *SAP Planning Guide: Db2 for z/OS*.

The following is a summary of the information contained in the section *Performance Tuning Considerations Before Installing an SAP System* of that document and in [Db2 Setup \[page 22\]](#):

- 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. You should adjust the system parameters that are categorized as recommended initial values based on the site-specific workload.
- Switch off unnecessary traces.

The only trace 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 significant 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*.

- Maintain the recommended SAP profile parameters settings.

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

## 8.1 Setting Optimal SAP Profile Values

Numerous SAP profile parameters are essential 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 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:

#### Sample Code

```
rsdb/max_blocking_factor = 20
rsdb/max_in_blocking_factor = 100
rsdb/min_blocking_factor = 3
rsdb/min_in_blocking_factor = 3
rsdb/prefer_fix_blocking = 0
rsdb/prefer_union_all = 1
rsdb/prefer_in_itab_opt = 1
```

The SAP profile parameters are contained in `/usr/sap/<SID>/SYS/profile`.

For more information, see SAP Note [2254216](#).

## 8.2 Monitoring and Tuning of SAP CDS Queries

SAP Core Data Services (CDS) typically involves SQL `SELECT` statements on a stack of database views and table `UDFs`, including joins and aggregations. To facilitate the monitoring and tuning of these queries, the SAP DBA Cockpit provides various tooling, such as:

- DDL Dependency Browser
- SQL Explain
- Query-specific Db2 Catalog Statistics
- SQL Statement-Cache Statistics

To investigate the performance of CDS queries, proceed as follows:

- To understand CDS queries better, the DDL Dependency Browser is a great starting point. By entering the top-level view or table `UDF`, used in the `SELECT` statement text, you can see all dependent objects underneath the top-level object. This gives you an idea about the relationships and joint conditions between tables. Also, by double-clicking on the name, you can see the complete `CREATE` statements of all objects.
- To ensure that the Db2 query optimizer has suitable Db2 Catalog Statistics for the CDS query, you should run it once. The Db2 12 optimizer can understand the query and determine whether specific statistics are missing. If so, it generates or extends the `Runstats` profile for this table. This profile is kept in the Db2 table `SYSIBM.SYSSTATFEEDBACK`. Then you should run `Runstats` and take advantage of the profile via option `USE PROFILE` according to SAP Note [2306718](#).
- SQL Explain information complements the DDL Dependency Browser insights by showing the access path that the Db2 query optimizer chooses for the query. The SQL Explain panel in the DBA Cockpit also shows you the number of records in each involved table. This is often useful information as it allows you to focus on the processing of these large tables.

It would be best to focus on the joins that involve the largest tables and check whether indexes are used for the join conditions. You should also consider the predicates in upper-level joins and in the main `SELECT`

statement that apply to these tables. You should usually use these predicates for matching index access if they are selective.

- A useful technique to get to the root of understanding the significant contributors to a CDS query's overall elapsed time is to extract a subquery and execute it independently. You can accomplish this by displaying relevant view definitions in the DDL Dependency Brower and by adding upper-level predicates if needed.
- For queries that involve considerable aggregation processing via `GROUP BY`, it can be beneficial to enable query parallelism in Db2 for these queries. To enable query parallelism, you can define a rule in the SAP optimizer profile according to SAP Note [2270401](#). The optimizer profile guideline needs to be `DEGREE ANY`.
- Finally, it would help if you looked at the `SQL Statement-Cache Statistics` for the CDS query. This provides insights on the selectivity of the clauses via the ratio of rows read and rows processed. The amount of synchronous I/O is also provided. If there is a noticeable I/O wait time, increase the size of the Db2 buffer pools that are used by the query. You should also check the number of Getpages of the query and determine whether it is aggregation processing or due to an inefficient access path. If the RDS limit was exceeded, consider increasing the Db2 `ZPARM MAXRBLK`.

## 8.3 Periodically Recycling SAP Work Processes

We recommended recycling SAP work processes periodically to reduce the amount of real storage accumulated by the Db2 threads that serve SAP work processes. This deallocates the Db2 threads, which frees the accumulated storage, and allocates new threads.

### Note

For database connections that use CLI failover with `KeepDynamic(YES)`, the DDF component of the Db2 server automatically recycles the Db2 server threads that correspond to these database connections at an hourly interval (see Db2 APAR PK69339). Therefore, it is not necessary to set `rdisp/wp_auto_restart` with CLI failover.

With CLI failover, if the periodic recycle of Db2 threads is not desirable, you can proceed as follows:

- Issue the following Db2 commands, which are introduced with Db2 12. These commands change the recycle frequency to 1200 minutes:
  - `MODIFY DDF KDRMUSED(1200)`
  - `MODIFY DDF KDRMIDDLE(1200)`
- Set `rdisp/wp_auto_restart = 71000`. With this configuration, the SAP work processes are recycled before DDF recycles 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 the 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 is reset. The corresponding Db2 thread is deallocated (freeing up all the accumulated storage), and then a new thread is allocated.

### Note

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

- If the work process is idle, than it is possible that it does not get restarted regularly. This can happen only at roll-out time. To address this problem, you should use the `rdisp/nowtime` parameter. It is also a timer given in seconds. Every `nowtime` interval, the SAP dispatcher sends a signal to the work processes that are not active at the time. Such a process checks the `wp_auto_restart` timer and, if expired, gets restarted. For performance, reasons ensure that the `nowtime` value is higher than the value for `wp_auto_restart`.
- The parameter value depends on the real storage usage 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 = 71000  
rdisp/nowtime = 71500
```

### Note

A planned Failover initiated from the DBA cockpit works NOT if you use `wp_auto_restart`. You may:

- Initiate a planned Failover by issuing a Stop DDF for the member to be freed
- Change upfront the `db2dsdriver.cfg` and add the AppServer to a list with its primary member not listed first.

After this action, wait until `wp_auto_restart` has hit again (or restart the AppServer).

## 8.4 Customizing the SAP Objects Topology

After installing an SAP system component, a large number of datasets backing thousands of related Db2 tablespaces and index spaces. To prevent contention on the volumes that accommodate heavily accessed datasets, you should take advantage of IBM HyperPAV (Parallel Access Volumes) and Multiple Allegiance or equivalent functionality from other suppliers to address these hot spots. This is particularly important if you are using DFSMS Extended Address Volumes (EAV). To get an overview of heavily accessed tables from SAP, call transaction ST10 to determine the access frequency and pattern on a per-table basis.

## 8.5 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 many Db2 catalog tables. Db2 provides the `RUNSTATS` utility that collects the necessary statistics and updates the catalog.

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. These considerations have been incorporated in the automated `RUNSTATS` mechanism provided in the DBA Planning Calendar of the SAP DBA Cockpit.

For more information about the `RUNSTATS` and other Db2 utilities referenced here, see the IBM documentation *Db2 12 for z/OS Utility Guide and Reference*.

## 8.5.1 When `RUNSTATS` Is Due

Outdated statistics are among 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 due to 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 selecting of the access path. Also, you should prevent `RUNSTATS` from kicking in during the SAP application peak hours. The `RUNSTATS` utility is usually executed on the special purpose zIIP processor.

Also, 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 impacts on the amount of available storage, the restart after an abnormal termination, and shutdown times. The automated `RUNSTATS` of the DBA Cockpit can be set up so that `RUNSTATS` is only collected during known hours of little workload. Also, you can integrate it with external schedulers to make sure that it starts when a nightly SAP batch job finishes (see SAP Note [1264471](#)).

### Determining Tables Needing New Statistics: Db2 Real-Time Statistics

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 `SYSDIBM.SYSTABLESPACESTATS` and `SYSDIBM.SYSINDEXSPACESTATS`. The Db2 stored procedure `DSNACCOX` analyzes these tables and — based on different criteria — deduces a list of tables that should be equipped with new statistics. The SAP system exploits `DSNACCOX` hourly and provides the list of tables for `RUNSTATS` in the CCMS Monitor Set (transaction `RZ20`). Also, the automated Runstats job *Update statistics of recommended objects* in the DBA Planning

Calendar (transaction `DBACOCKPIT` ► [Jobs](#) ► [DBA Planning Calendar](#) ►) works on the recommended tables `DSNACCOX`. As Db2 keeps track of the database changes and formulates the recommendations, `DSNACCOX` and the CCMS Monitor Set provide a very precise and efficient way to determine the tables that need new statistics.

## When to Schedule RUNSTATS

You should schedule `RUNSTATS`:

- **As soon as convenient for 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 system tables. `RUNSTATS` should be run at this time for catalog tables as well because there is a large number of new database objects.
- **During a table import (if the table is concurrently accessed) it is essential to run `RUNSTATS` in the first 15%-25% of the estimated runtime.**

For instance, batch input includes queries and inserts, and the queries need current statistics to use optimal access paths.
- **For 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 tablespaces and indexes that have just been recovered**
- **For 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 you never executed `RUNSTATS` 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 containing a table referenced by these cached statements. Starting with Db2 12, `RUNSTATS` only invalidates statements in the statement cache if the `INVALIDATECACHE` clause is used. 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.

### Note

Creating a new index invalidates the cached statements associated with the base table of the index.

The SAP BW system automatically schedules `RUNSTATS` with appropriate options for star schema tables when necessary. For example, after new data has been loaded.

## 8.5.2 RUNSTATS Options to be Used

In general, the following RUNSTATS specifications are recommended for the SAP system environment. SAP automated Runstats collect these statistics by default.

- To collect and update catalog statistics for all the tables in a tablespace:

### ↳ Sample Code

```
RUNSTATS
  TABLESPACE <tablespace name>
  TABLE (ALL)
  SAMPLE
  INDEX(ALL)
  USE PROFILE
  SHRLEVEL(CHANGE)
  INVALIDATECACHE YES
```

Keep in mind that the KEYCARD option is deprecated and is always automatically used by RUNSTATS. Be aware that the USE PROFILE option does not require that a profile exists. If no profile exists for a given tablespace, then RUNSTATS ignores the USE PROFILE option.

- To collect and update catalog statistics for a single table in a given tablespace:

### ↳ Sample Code

```
RUNSTATS
  TABLESPACE <tablespace name>
  TABLE(table name)
  SAMPLE
  INDEX (<index1 name> KEYCARD, <index2 name> KEYCARD, ...)
  SHRLEVEL(CHANGE)
```

Be aware that the KEYCARD option is deprecated and is always automatically used by Runstats.

- To collect and update catalog statistics for a given index:

### ↳ Sample Code

```
RUNSTATS
  INDEX (<index name> KEYCARD)
  SHRLEVEL(CHANGE)
```

- To collect and update catalog statistics for a LOB tablespace:

### ↳ Sample Code

```
RUNSTATS
  TABLESPACE <name of auxiliary tablespace>
  SHRLEVEL(CHANGE)
  INDEX(ALL)
```

- To invalidate a cached statement that references a table in <tablespace name>:

### ↳ Sample Code

```
RUNSTATS
  TABLESPACE <tablespace name>
```

```
SHRLEVEL (CHANGE)
INVALIDATECACHE YES
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, 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 ten most frequently occurring values of the first column of an index. To automatically collect these additional statistics for a specific table, you can define a `RUNSTATS` profile in the Db2 catalog table `SYSIBM.SYSTABLES_PROFILES` for the table. The `RUNSTATS` option `SET PROFILE` would need to be run once to create such a profile, which is then considered for future `RUNSTATS` calls.

As of Db2 12, the Db2 query optimizer generates profiles based on the individual query workload on the tables. So they are tailored to the workload and hence should ensure that suitable statistics are collected the next time `RUNSTATS` is executed.

## 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, you can always use frequency distribution for these queries. However, SAP usually 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 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 they provide 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.

### Note

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

For more information about the SQL statements that require additional treatment also with `REOPT(ONCE)`, see SAP Note [1008334](#)

## 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.

As of Db2 12, the inline statistics function of `REORG` also supports `USE PROFILE`. This allows you to collect the full set of statistics every time you run `REORG`. Hence you can avoid running the `RUNSTATS` utility afterward.

## 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. You must define the Scheduler Work Area (SWA) above the 16 MB line. For more information, see the *Planning Guide for SAP on IBM Db2 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*.

## 8.5.3 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*.

### Special Considerations for Volatile Tables

Special considerations also apply to volatile tables. If `RUNSTATS` is run on an 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 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 heavy lock contention, including deadlocks.

## Db2 System Parameter NPGTHRSR

Fortunately, we can avoid most of these problems by telling the *optimizer* to use a heuristic approach instead of cost-based optimization if the tables are small (for example, less than ten pages) or empty. For such a table, you should select the following access path.

If there is an index with at least one matching column for given statements' predicates, you should avoid a tablespace scan. If there is more than one such index, choose the one with the largest matching columns. If there is still more than one qualifying, choose the one that provides ordering (if applicable). You should select neither list prefetch nor sequential prefetch.

The way to achieve this is simple: You only need to set the Db2 system parameter `NPGTHRSR`. Its value is taken into account during access path selection. For a given table, if `NPAGES` is less than the `NPGTHRSR` value and not -1, index access for the table will be preferred over a tablespace scan. The recommended value is 10.

As of Db2 12, `NPGTHRSR` also applies to tablespaces with `NPAGES = -1` so that it covers objects for which stats have never been collected so far.

### Note

Volatile tables are addressed by the system parameter `NPGTHRSR` and that SAP special tables are addressed by the Db2 table attribute `VOLATILE`. For more information, see [Access Path Considerations for Special SAP Tables \[page 266\]](#).

## Index Probing and Use Of Real-Time Statistics

Whenever the Db2 catalog statistics indicate an extreme situation regarding a specific query on a table, not one row matches the query predicates. Db2 10 introduces a logic to probe an adequate index to validate the statistics. According to the catalog statistics, this approach is an additional safety net that prevents the *optimizer* from choosing an access path that would be suitable. This is, in reality, not good because the statistics are outdated or incorrect. This logic also allows the *optimizer* to rely on real-time statistics data in case no catalog statistics are available.

## 8.5.4 Access Path Considerations for Special SAP Tables

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**

The query returns a complete list of these tables:

### Sample Code

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

- **ABAP Export/Import Tables**

These tables are also called ABAP clusters. The query returns a complete list of these tables:

### Sample Code

```
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)
      ))
```

- **DDIC Tables**

- DD03L
- DD08L

## Purpose

You must access all of these special tables in a particular way to minimize deadlock occurrences and optimize their performance. Their optimal access path should not be cost-based (like for many 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. There is one exception to this: the tables in this category with more than one index, such as TRFC and QRFC tables, which are somehow laxer. For these tables, the access paths as imposed by the Db2 parameter `NPGTHRS` are appropriate.

## Solution

Ensure 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. For the more relaxed TRFC and QRFC tables, setting the `VOLATILE` attribute means that the `NPGTHRS` rules for accessing these tables are permanently enabled, and this is good.

## 8.5.5 Access Optimization to VBHDR, VBMOD, and VBDATA

### 8.5.5.1 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 essential to ensure optimal performance for the statements that refer to the tables. There are three significant areas in tuning the VB protocol tables:

- Optimal Access Paths Assurance for Table Access Statements
- Assigning the VB Protocol Tables to Dedicated Buffer Pools
- Partitioning VB Protocol Tables

### 8.5.5.2 Optimal Access Paths Assurance for Table Access Statements

Among other requirements listed in [Access Path Considerations for Special SAP Tables \[page 266\]](#), you must ensure that only one index is defined for each VB protocol table (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:

```
SELECT      TBNAME, NAME, UNIQUEROLE
```

```

FROM      SYSIBM.SYSINDEXES
WHERE     TBCREATOR=' <SCHEMA>'
AND       TBNAME IN ( 'VBMOD', 'VBHDR', 'VBDATA' )
ORDER BY  TBNAME, NAME;

```

The index name suffix can be identified (it is 0 for primary index) and by the `UNIQUERULE` value equal to "P".

## 8.5.5.3 Assigning the VB Protocol Tables to Dedicated Buffer Pools

### Use

When you install the SAP system, all SAP tables, (including the VB protocol table `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.

#### Note

Some of the steps are also required for partitioning the VB protocol tables. For more information, see [Partitioning VB Protocol Tables \[page 269\]](#).

### 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.

#### Note

The transport `KDOK000668` is automatically applied during the upgrade to SAP NetWeaver 7.0 and higher. This transport needs to be applied manually after a new SAP system has been installed. If the history of the system includes such an upgrade, you can skip this step.

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 [Range-Partitioning Tables \[page 203\]](#).
5. Assign the tablespaces and indexes to a new buffer pool, for example, BP4. This step can also be executed using transactions SE14. For more information, see [Directly Changing Storage Attributes \[page 202\]](#).
6. Set the buffer pool parameters:

```
ALTER BUFFERPOOL(BP4) VPSIZE(1000) VPSEQT(10) DWQT(70) VDWQT(50)
```

## Result

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 \[page 280\]](#).

## 8.5.5.4 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 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 \[page 267\]](#) and [Assigning the VB Protocol Tables to Dedicated Buffer Pools \[page 268\]](#) have been performed.

### Procedure

#### ⚠ Caution

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 tablespaces. For more information, see [Assigning the VB Protocol Tables to Dedicated Buffer Pools \[page 268\]](#).
3. Partition the VB protocol tables  
Partitioning is actively supported by transaction SE14, but since `VBHDR`, `VBMOD`, and `VBDATA` are multiplex tables, you cannot convert them. Consequently, you have to proceed as follows:
  1. Call transaction SE14.
  2. Choose *Delete database table*.

3. Choose *Storage parameters*.
4. Specify and save the storage parameters for the partitioning of the table. As partitioning keys, use VBDATA's first column VBKEY. This column encompasses the IP address of the application server in hexadecimal code.  
If you prefer to use SAP application server's hostname as a partitioning key rather than the IP address, set the SAP profile parameter `rdisp/vk_key_use_hostname` to 1. If you have long hostnames, you need to set the profile parameter `rdisp/vb_key_long_hostname` to 1 additionally. **SAP Note 191191** describes these profile parameters. If you use virtual hostnames, make sure to select the profile parameters `SAPLOCALHOST` and `SAPLOCAHOSTFULL`.

### Example

There are four application servers for the dialog processes.

Their IP addresses are:

#### Sample Code

```
155.56.94.121 (hex 9B.38.5E.79)
155.56.94.122 (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:

#### Sample Code

```
155.56.94.121 -> '9B385E7911...'
155.56.94.122 -> '9B385E7A11...'
155.56.94.123 -> '9B385E7B11...'
155.56.94.124 -> '9B385E7C11...'
```

Accordingly, it would help if you used the following attributes for the partitioning:

#### Sample Code

```
PART 1 VALUES ('9B385E7911')
PART 2 VALUES ('9B385E7A11')
PART 3 VALUES ('9B385E7B11')
PART 4 VALUES (X'FF')
```

5. Also, specify buffer pool BP4 for each VB tablespace and index.  
On the initial screen of transaction SE14, select *Create database table*.
4. Set the buffer pool parameters:

#### Sample Code

```
ALTER BUFFERPOOL(BP4) VPSIZE(1000) VPSEQT(10) DWQT(70) VDWQT (50)
```

5. Change the SAP profile:

### Sample Code

```
dynp/trans_id_format=2
```

6. The following recommendations apply to data sharing environments only:
- Establish good affinity between VB table partitions and application servers. For a set of application processes connected to a Db2 data sharing member, define the corresponding update processes on the application servers connected to the same data sharing member. The easiest way to achieve this is to disable the 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:

Profile Parameters - Partitioning VB Protocol Tables

Profile Parameters	Value
<code>rdisp/vb_dispatching</code>	0
<code>rdisp/vb_included_server</code>	no entry (for example, blank)
<code>rdisp/vbname</code>	name of the local application server; also possible replacement variable: \$ ( <code>rdisp/myname</code> )

### Note

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](#).

## 8.6 Clustering Index

Excessive I/O can cause bad transaction response times for specific statements.

When a row is inserted in a table, Db2 tries to place it near the rows with a similar key values for the index known as the **clustering index**. Placing the rows in this manner 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. Also, Db2 can maximize the effects of its sequential prefetch feature.

### How the SAP System Defines the Clustering Index

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 customers

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 the optimal choice.

## Index-Sequential Table Access

A table may be 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`, `>`, `<`), which include only a prefix of an index or for accesses through a non-unique index. You can identify these statements in the [DBACOCKPIT > Statement Cache Overview](#). If they have relatively large average response times and these times are caused by I/O suspensions, you should consider the following recommendations:

- Suppose some statements access a table index-sequentially through an index that is not defined as clustering, and this is a predominant way that table is accessed. In that case, you can improve the performance by specifying that particular index as clustering. To change the clustering index, issue `ALTER INDEX` with option `CLUSTER` on the index want to set 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.
- The `CLUSTER` option can also be changed within the SAP system. For more information, see [Directly Changing Storage Attributes \[page 202\]](#).  
In the real-time statistics table, Db2 keeps track of SQL queries on a table that would benefit from a proper clustering sequence. This information is kept in the real-time statistics field `REORGCLUSTERSENS`. You can use the information in the field to assess the impact of changing the clustering index.
- Be aware that a transport or SAP release upgrade that changes the table's structure in such a way that the table needs to be re-created would reinstate the primary index as the clustering index. In this case, you need to repeat the process of changing the clustering index.

## 8.7 Partitioning Key

By default, SAP relies on the implicit object creation of Db2. This means that every table is contained in a partition-by-growth universal tablespace (PBG UTS). The table can grow virtually unlimited since Db2 automatically adds a further partition whenever the existing partitions run out of space. There is no logical partitioning of the table, however. Instead, the clustering sequence governs into which partition a new record is inserted. There may be reasons to range-partition a table – for example, to define an affinity of table partitions to data sharing members or to have records with a similar property like age in the same partition. This is often the case in SAP BW. With range partitioning, partitioning keys define how table rows are distributed among the partitions of a partitioned tablespace. 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 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.

## Create and Modify Partitioned Tables

The SAP system provides transactions and functions to create and modify partitioned tables. For more information, see [Range-Partitioning Tables \[page 203\]](#).

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.

### Note

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 from increasing in size much faster than others, as this may bring you close to the maximum partition size. The tablespace attribute `DSSIZE` defines the maximum partition size. As `DSSIZE` and the page size of a tablespace affect the maximum number of partitions, you should aim to choose a value for `DSSIZE` that supports the number of required partitions. For example, tablespaces with a page size of 4 KB and `DSSIZE` = 4 GB support up to 4096 partitions. Tablespace with a page size of 4 KB and `DSSIZE` = 64 GB supports only 256 partitions. For more information, see the IBM documentation *Application Programming and SQL Guide*.

Suppose, at a later point in time, and you notice that the selected key ranges are not optimal. In that case, you can modify key ranges or dynamically add partitions, either by using SAP transactions (see [Range-Partitioning Tables \[page 203\]](#)) or by directly issuing an `ALTER TABLE` statement.

## Format of the Partitioning/Update Key

You can set the value of `MAXVALUE` as `LIMITKEY`:

1. Call transaction `SE14`.
2. Maintain the storage parameters as described in section [Range-Partitioning Tables \[page 203\]](#).
3. Enter `MAXVALUE` in the `LIMITKEY` field, which corresponds to the last entry in the *Partition* list.

For more information, see SAP Note [191191](#).

## 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 single Db2 member mainly processes a partition, the locking volume of P-Locks is further reduced.

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

## 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. You can group numerous application servers of an SAP system in transaction `SM61`. When you schedule batch jobs in transaction `SM36`, you can then use these groups 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 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.

## 8.7.1 Partitioning SAP Banking Tables

For best performance of SAP Banking applications, it is crucial that you range-partition their key tables to encourage the periodic end of day or end of month processes to settle accounts. SAP Note [496904](#) describes how you should partition these tables. The report `RBCA_PARTITION_TABLES` helps you in implementing the partitioning schemes. For more information, see SAP Note [1295863](#).

## 8.7.2 Partitioning Financial Tables

In specific financial tables, rows of different years are physically sorted by `BELNR` in financial tables like `BKPF` or `BSIS`. These tables' common trait is that `GJAHR` follows their `BELNR` columns in the primary key. The common trait of these tables is that their `BELNR` columns are followed by `GJAHR` in the primary key. Since `BELNR` is 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.

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 posting of one year. Within each partition, the rows are clustered according to the primary index. This perfectly suits queries that access the posting of a given year only, which should be the usual case.

## 8.8 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`=300
  - `XLKUPDLT`=TARGET
  - `RELCURHL`=YES
  - `EVALUNC`=YES
- Bind options
  - `ISOLATION`=UR
  - `BLOCKING` ALL
  - `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.

## Lock Escalations

Applications that perform massive updates and deletes and do not commit frequently can accumulate many 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 a too 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.

### ❖ Example

If the IRLM private address space's size 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 many updates that now need to be backed out. For example, the application must do a rollback. Consequently, the backout 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 or later, eliminating this limitation and introducing 64-bit support. Furthermore, the Lock Hash Table and the Record List Entries (RLE) sections share the storage allocated to the lock structure. The initial split in terms of relative storage allocation is explicitly specified through the parameter `IRLM HASH` (introduced by RLM 2.1 APAR PQ44114) or controlled by an internal IRLM algorithm. The maximum number of modifying locks related to database update processing that can be concurrently held and propagated to the CF is limited by the RLE section's size. The actual number of modifying 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 information, see the IBM documentation [Coupling Facility Level \(CFLEVEL\) Considerations](#). 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 a limit is reached (or, more precisely, 90 % of the limit values), new modified 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 a long time if many 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, many 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 database's growth 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 away. The problems are

more likely to be encountered with user-written programs, which need to be reviewed to ensure an appropriate commit frequency.

Also, Db2 also provides mechanisms for limiting the number of locks concurrently held: the system parameters `NUMLKUS` and `NUMLKTS`, 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 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 `NUMLKTS` 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.

### Note

`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, 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, suppose `LOCKMAX` is set too low. In that case, an application process that has acquired more than `LOCKMAX` locks could trigger 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. However, 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 many ways to identify these critical transactions and tablespaces:

- **Monitor the maximum number of locks that are held by individual Db2 threads**

The SAP DBA Cockpit 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 descending order and investigate the corresponding work processes for which transactions, reports, and tables have been involved in acquiring and holding many locks. Transaction SM66 displays which tables are being accessed by which work process.

- **Monitor lock escalations**

Whenever an escalation occurs, Db2 issues IFI record 337, and the message DSNI031I is written to the z/OS console. The database alert router of SAPCL catches IFI record 337 and provides this alert in a transaction in the SAP DBA Cockpit. 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 `Statement Cache Overview`.**

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 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 DBA Cockpit exploits IFI record 313, which allows you to identify long-running transactions. For more information, see the [DB Alert Router](#).

Apart from enabling lock escalation, you should ensure that the IRLM and CF are sized to maximize the locks' number 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 modifying locks that can be concurrently held. The split can be explicitly specified using 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 available central storage on the CF, allowing for structure failover from the alternate CF.

### Note

The central storage must back the entire lock structure. Of course, keeping the number of false contentions small (directly affected by the Hash Table size) 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 regularly committing 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 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.

### ❁ 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 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.
- 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 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 **DBACOCKPIT** **Alerts** **Long Running URs**. 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. Suppose 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. For more information, see [Optimal Access Paths Assurance for Table Access Statements \[page 267\]](#) and [Access Optimization to VBHDR, VBMOD, and VBDATA \[page 267\]](#).

**Avoid application server buffering** for tables with frequent updates. That not only adds to the cost of buffer resynchronizations but can also cause heavy lock contentions.

Call the SAP DBA Cockpit for monitoring locking events, including snapshots of current locking conflicts across the system. See the SAP DBA Cockpit to identify timeouts and deadlocks. For more information, see [Monitoring and Performance \[page 81\]](#).

## 8.9 Buffer Pool Tuning Considerations

Buffer pools are among the essential objects in Db2 performance monitoring and tuning. After installing 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. Make sure that there is no real memory paging to AUX storage in your LPARs.

Depending on the size of your workload, you should adjust the sizes of the Db2 buffer pools. If there are too many buffer pools defined in a Db2 subsystem, this leads to fragmentation of the storage. It is unlikely that the peak workload on the different tables and indices co-occur. This means that assigning these objects to a large number of buffer pools would likely result in some buffer pools being idle and other buffer pools being stressed. The basic idea of buffer pool tuning with SAP would rely on the standard buffer pools described earlier. If needed, the sizes of the buffer pools should be increased or decreased. For the few tables and indices that are crucial, you should use dedicated buffer pools. It may not be feasible for very large critical tables or indices to completely back them by a buffer pool. If this is the case, consider using solid-state drives (SSD) as disk drives for these objects to minimize I/O wait time.

For decent performance, you should fine-tune the buffer pool settings. We recommend that you tune the buffer pools used by the tables and indexes of these transactions or batch jobs to ensure the best performance of your most essential transactions or batch jobs in your SAP system. When you focus on the most important objects, you avoid a fragmentation of the storage that is used by the buffer pools and require less tuning efforts.

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 crucial tables and indices in dedicated buffer pools

### → Recommendation

Buffer pool tuning is a 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 the development of new bottlenecks and performance deficiencies, you need to establish a base for performance evaluation. You can do this by collecting and storing the performance data over a more extended period, but most importantly, before and after any tuning activities.

You can use the following data to identify changes in workload and throughput:

- Number of getpages per reporting interval
- Number of buffer updates per reporting interval
- CPU utilization

You can use the following data 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 of appropriate adjustments.

You can use the following data to check if storage is overcommitted:

- Page-ins for reading and write
- z/OS paging activity

You can use the following data to measure the effectiveness of the tuning:

- Overall hit ratio, which is derived as:  
total getpages or total pages read / total getpages  
where total pages read = total synchronous reads +  
pages read by sequential prefetch +  
pages read by dynamic prefetch +  
pages read by list prefetch
- Random hit ratio, which is derived as:  
(random getpages or random synchronous reads) / random getpages
- Read rate, which is derived as:  
(total synchronous reads + all type prefetch reads) per interval
- Buffer updates per pages written
- Pages are written per write I/O, which is derived as:  
pages written / (synchronous writes + asynchronous writes)
- Write rate, which is derived as:  
(synchronous writes + asynchronous writes) / interval
- z/OS DASD and Cache activity ratios

## 8.10 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 the 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 as a starting point. You need to monitor the global cache hit ratio (reported in [Db2 Subsystem Activity](#) monitor, transaction `DBACOCKPIT` ► [Performance](#) ► [Db2 Subsystem Activity](#) ►). If it is consistently lower than 95%, increase `EDMSTMTC` in increments of 20 MB. Do not oversize it because you will overcommit the real storage and increased system paging. Make sure that all application servers use the same user ID to connect to Db2. Namely, the identical SQL statement 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 **12000** (Db2 11) or **15000** (Db2 12). If you are running with a larger number of SAP work processes, you should consider increasing `MAXKEEPD` increments of 6000 to reduce CPU utilization.
- System parameters  
Ensure that the delivered packages and plans used by the SAP system are bound with `KEEPDYNAMIC(YES)`.

## 8.11 Data Sharing Optimization for Different SAP Business Applications

In general, the Db2 data sharing performance and scalability are excellent with SAP Business Applications.

Also, there are specific tuning steps to optimize further data sharing performance that can be accomplished for specific applications:

- The data sharing optimizations for SAP Banking Applications are summarized in [SAP Note 496904](#) 📄.
- The data sharing optimizations for SAP Business Intelligence are summarized in [SAP Note 1239127](#) 📄.
- The data sharing optimizations for SAP PI are summarized in [SAP Note 1260453](#) 📄.

## 8.12 Transaction-Level 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 work units within the application server. It provides very detailed information on both the static and dynamic properties of the SAP system.

**Static properties** do not change during the lifetime of SAP work process 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 used to execute a specific SAP transaction is tagged with Db2 client identifiers filled with information about that transaction. These identifiers contain details of the SAP system that runs 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 manages the Db2 workload using WLM, accounting data can be analyzed, and workload management can be performed at a very fine level.

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, using WLM, a higher priority can be assigned to a crucial SAP transaction or batch job.

## 8.12.1 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 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 employed by this SAP attribute. For more information, see [Static Properties Passed to Db2 \[page 284\]](#) and [Dynamic SAP Properties Passed to Db2 \[page 285\]](#).

When you have this information, you can define WLM classification rules for the WLM qualifier used by this Db2 client identifier.

Mapping Between SAP, Db2 and WLM Identifiers

Client Connection Attribute (ABAP)	Client Info Property Name (J2EE)	WLM Qualifier	IFI Field	Special Register	OMEGAMON XE for Db2 PE Identifier
SQL_ATTR_INFO_ PROGRAMNAME		CI [1]	QWHCCV		CORRNAME
SQL_ATTR_INFO_ ACCTSTR	ClientAccountingInformation	AI [56-143]	QMDASUFY	CURRENT CLIENT_ACCTNG	
		UI [1]	QWHCAID	SESSION_USER	PRMAUTH
SQL_ATTR_INFO_ APPLNAME	ApplicationName	PC [1]	QWHCEUTX	CURRENT CLIENT_APPLNAME	TRANSACT

Client Connection Attribute (ABAP)	Client Info Property Name (J2EE)	WLM Qualifier	IFI Field	Special Register	OMEGAMON XE for Db2 PE Identifier
SQL_ATTR_INFO_WRKSTNNNAME	ClientHostName	SPM [17-34]	QWHCEUWN	CURRENT CLIENT_WRKSTNN AME	WSNAME
SQL_ATTR_INFO_USERID	ClientUser	SPM [1-16]	QWHCEUID	CURRENT CLIENT_USERID	ENDUSER

## 8.12.2 Static Properties Passed to Db2

The following table shows the Db2 client identifiers to which the static SAP attributes are assigned. The SAP application server's work processes pass these attributes to Db2 when they establish connections to Db2.

### Note

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.

Static Properties Passed to Db2 (ABAP)

SAP Attribute	Db2 Connection Attribute (ABAP)	Positions	WLM Qualifier
SAP System ID (SAPSID)	SQL_ATTR_INFO_PROGRAMNAME	1-3	CI [1-3]
Work Process Type	SQL_ATTR_INFO_PROGRAMNAME	4-6	CI [4-6]
Work Process Number	SQL_ATTR_INFO_PROGRAMNAME	7-9	CI [7-9]
Host Name of Application Server	SQL_ATTR_INFO_ACCTSTR	1-32	AI [56-87]
SAP System Number (SAP-SYSTEM)	SQL_ATTR_INFO_ACCTSTR	33-34	AI [88-89]
Work Process ID	SQL_ATTR_INFO_ACCTSTR	35-44	AI [90-99]
Database Connection Name	SQL_ATTR_INFO_ACCTSTR	45-74	AI [100-129]
Database Connection User ID			UI
Indicator for SAP ABAP Workload (constant SAP)	SQL_ATTR_INFO_ACCTSTR	75-77	AI [130-132]

SAP Java applications pass the following attributes:

Static Properties Passed to Db2 (Java)

SAP Attribute for sapjvm 1.6	SAP Attribute for sapjvm 1.4	Client Info Property Name (J2EE)	Position	WLM Qualifier
Host Name of Application Server	Host Name of Application Server	ClientHostname	1-18	SPM [17-34]
Database Connection User ID				UI
Indicator for Java Workload (constant string db2jcc_appli)	Indicator for Java Workload (constant string db2jcc_appli)			CI [1-12]
Host Name of Application Server	Host Name of Application Server	ClientAccountingInformation	1-32	AI [56-87]
00	00	ClientAccountingInformation	33-34	AI [88-89]
Work Process ID	N/A	ClientAccountingInformation	35-44	AI [90-99]
Database Connection User ID	Database Connection User ID	ClientAccountingInformation	64-72	AI [119-127]
Indicator for SAP Java Workload (constant S2E)	Indicator for SAP Java Workload (constant S2E)	ClientAccountingInformation	75-77	AI [130-132]
SAP System ID (SAPSID)	SAP System ID (SAPSID)	ClientAccountingInformation	78-80	AI [133-135]

### 8.12.3 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 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 transaction set at the transaction begins. For SAP batch jobs, the dynamic SAP properties are set at job to begin and do not change during their execution.

The dynamic SAP properties are listed in the tables below:

Dynamic SAP Properties Passed to DB2 (ABAP)

SAP Attribute	Client Connection Attribute (ABAP)	Positions	WLM Qualifier
ABAP Report Name	SQL_ATTR_INFO_APPLNAME	All [1-32]	PC
BI infoProvider '/' BI Report Name	SQL_ATTR_INFO_APPLNAME	All [1-32]	PC
SAP Transaction Code	SQL_ATTR_INFO_WRKSTNNAME	All [1-18]	SPM [17-34]
SAP Batch Job Name	SQL_ATTR_INFO_WRKSTNNAME	All [1-18]	SPM [17-34]
SSAP End User ID	SQL_ATTR_INFO_USERID	1-12	SPM [1-16]

Dynamic SAP Properties Passed to DB2 (Java)

SAP Attribute for sapjvm 1.6	SAP Attribute for sapjvm 1.4	Client Info Property Name (J2EE)	Position	WLM Qualifier
J2EE Application (if available)	N/A	ApplicationName	1-32	PC
J2EE End User (if available)	J2EE end user (if available)	ClientUser	1-16	SPM [1-16]
Java Thread ID	Java Thread Name	ClientAccountingInformation	45-63	AI [100-118]
Thread Type (S/A/ )	N/A	ClientAccountingInformation	81-81	AI [136]
Java Thread Name	Java Thread Name	ClientAccountingInformation	100-179	

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 BW infoprovider and BWI report name's concentration 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.

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 KEEP DYNAMIC (YES), the Db2 threads that serve SAP nevertheless remain active all the time. This behavior is desired as it optimizes performance.

### ⚠ Caution

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 essential 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, resulting in more predictable names 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 names are concatenated by a slash (/).

For function modules called via RFC, the application server uses the same algorithm to determine the ABAP report name to be propagated to Db2 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 `SAMPLE_GROUP`, then the application server would pass `SAPLSAMPLE_GROUP` as ABAP report name to Db2.

During batch job execution, SAP propagates the ABAP report's name 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.

### ℹ Note

The ABAP report `RSM13000` performs all work in update work processes.

## 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 an SAP transaction code to Db2 for them.

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

For OLAP queries from SAP BWI that are executed in dialog processes, the string BW\_REPORT is passed to the Db2 instead of the SAP transaction code. This allows you to identify the 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 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 job's duration. 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. Held cursors, held LOB locators, and active declared global temporary tables avoid accounting and WLM enclaves' intervals to complete at the end of transactions. It would help if you considered this 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 at *Db2 for z/OS Administration Guide*.

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

For IDOCs, the individual message types cannot be deduced since the Db2 accounting data for IDOC processing is fairly coarse-grained.

The following identifiers are passed to Db2 for units of work-processing IDOCs (see SAP Note [995095](#)).

- IDOC message type to Db2 transaction time
- String IDOC to Db2 workstation name
- SAP user ID processing IDOC to Db2 client user ID

By default, ABAP Web Services show up in the Db2 transaction name field as `SAPMHTTP`. You should apply the correction instructions in SAP Note [1368915](#). The following string is then passed to Db2 for the transaction name field:

```
SIW=====>Web Service
```



### Note

The logical SAP hostname is part of the Db2 accounting string.

## 8.12.4 Analysis of Accounting Records

To analyze the accounting records that Db2 generates, you can use a tool like Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS (OMPE). Such a tool allows you to filter out and view individual records and aggregate accounting records for Db2 client identifiers. With this information, you can chargeback costs to individual users or departments.

Suppose you want to create an accounting report that summarizes the Db2 performance data per SAP report name. In that case, you can specify the clause `ORDER (TRANSACTION)` when calling OMPE to generate the report.

The structure that OMPE offers to define groups of users or values is the set name.

Another feature of OMPE concerns the correlation ID field. OMPE allows you to divide this identifier (which contains many different SAP attributes) into the subfields **correlation name** and **correlation number**. OMPE treats these fields like it would any other identifiers. You can modify the correlation ID translation as well.

# 9 Additional Information

## 9.1 Transaction Codes

The following table gives an overview of transaction codes useful in the administration of an SAP system on Db2 for z/OS.

Transaction Codes

Transaction Code	Description
AL08	List of all Logged on Users
AL11	Display SAP-Directories
DBACOCKPIT / DB2 / ST04	Start DBA Cockpit
RZ20	CCMS Monitoring
RZ21	CCMS Monitoring Customizing
SE11	ABAP Dictionary
SE14	Database Utility
SE16	Data Browser
SE17	General Table Display
SE30	ABAP Runtime Analysis
SE38	ABAP Editor
SE80	Object Navigation/ABAP Workbench
SE84	Repository Information System
SM04	Current Users on the Current Server
SM12	Display and Delete SAP Locks (SAP Enqueue locks)
SM13	Update Records
SM21	System Log
SM31	Table Maintenance
SM37	Background Job List

Transaction Code	Description
SM50	Work Process Overview
SM51	Server Overview
SP01	Spool Output Controller
ST02	Tuning Summary
ST03	Workload Analysis and Performance Statistics
ST05	Performance Trace
ST06	Operating System Monitor
ST10	Table Call Statistics
ST11	Developer Traces
ST22	ABAP Dump Analysis
STAD	SAP Statistic
SU01	User Maintenance
SU50	View/Set Own User Data
SU53	Evaluate Authorization Check

## 9.2 Environment Variables and Profile Parameters

The following table lists important environment variables and profile parameters needed in an SAP system on Db2 for z/OS.

Profile parameters use / (slash) while environment variables use \_ (underscore) only.

For additional information, call transaction RZ11, specify the profile parameter, and choose *Display*.

Environment Variables and Profile Parameters

Environment Variable / Profile Parameter	Definition
DB2CODEPAGE	For SAP systems using Unicode, this variable needs to be set to 1208. For SAP systems using ASCII, it needs to be set to 819.

## Environment Variable /

Profile Parameter	Definition
dbms_type dbms/type	Specifies database type (for example: Db2)
dbs_db2_schema dbs/db2/schema	Owner of database objects for this component.
dbs_db2_ssid dbs/db2/ssid	Db2 subsystem id (primary DB).
dbs_db2_use_accounting dbs/db2/use_accounting	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 connected via DRDA. The default value is 0.
dbs_db2_dsn_alias dbs/db2/dsn_alias	Name of the alias in db2dsdriver.cfg for the primary database connection.
dbs/db2/net_stats	Enable/disable CLI driver network statistics The default value is 0 (disabled)
dbs_db2_client_aware dbs/db2/client_aware	If set to 1 the current client is added as comment to the SQL statement to be executed. Due the usage of the REOPT(ONCE) feature the DB optimizer might calculate better access paths in multi-client systems with uneven data distribution across clients. The size of the Dynamic Statement Cache in Db2 should be increased when using this feature. The default value is 0 (disabled)
dbs_db2_autobind dbs/db2/autobind	If Db2 Connect version is upgraded to a new major/minor version or Db2 function level is changed a new collection id is calculated. The required bind is done automatically with the first database connect. The value of 0 disables this feature, default is 1.

Environment Variable /

Profile Parameter	Definition
dbs_db2_locale dbs/db2/locale	Unicode standards used by the SAP kernel might change between releases and require adjustments in Db2 and database communication. Possible values:  Unset: kernel default is used  0: UNI_SIMPLE is used  6: UNI_60 is used (Db2 Unicode 6.0, APAR PH19888)  9: UNI_90 is used (DB2 Unicode 9.0, APAR PH47187)
dbs_db2_user dbs/db2/user	Database Connect User.  Required parameter.
DIR_LIBRARY	The full pathname of the directory that contains Database Interface shared library
SAPDBHOST	TCP/IP address of database host
SAPSYSTEM	SAP system name
SAPSYSTEMNAME	
SQL_TRACE dbs_db2_sql_trace dbs/db2/sql_trace	Switches on and set the granularity of the SQL trace
dbs_db2_cli_trace dbs/db2/cli_trace	The SAP profile parameter <code>dbs/db2/cli_trace</code> controls the CLI trace. 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 work process is concatenated with the string 1, .  For example, if the CLI trace is turned on for the work process 12 only, <code>dbs/db2/cli_trace</code> would be set to 1, 12. The profile parameter <code>dbs/db2/cli_trace</code> 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 <code>p&lt;operating system process ID&gt; t1.cli</code> .
dbs_db2_cli_trace_dir dbs/db2/cli_trace_dir	Defines the target directory for the CLI trace files.

## 9.3 Update Task Troubleshooting

### Use

This section describes how you prevent the update task from being stopped when Db2 SQL code -904 occurs.

### Features

If the update task encounters a Db2 SQL code -904 with the message “resource unavailable”, it is stopped. The code usually indicates that you need to consult your DBA.

However, for some reason, codes do not require stopping the update task since the resource is only temporarily unavailable.

List of known reason codes that do not require the update task to be stopped:

Reason Code	Description
00C90096	NUMLKUS exceeded
00E7000F	Deadlock
00C20031	Latch timeout
00C20255	No P-lock
00C900A5	Temporary file error

You can set a profile parameter with additional reason codes from new releases that do not require the update task to be stopped.

### Integration

You set the profile parameter `db2/db2/nupd_stop` with a list of the additional reason codes you have chosen not to require that the update task be stopped.

### Example

```
db2/db2/nupd_stop = <reason1>, <reason2>, ..., <reasonN>
```

### Syntax Conventions

- Every reason code must be exactly eight characters long.
- A maximum of 20 reason codes is supported.

## 9.4 SAP MDM on Db2 for z/OS

This section describes the environment and parameters that you need when using SAP MDM with Db2 z/OS.

### Unicode

SAP MDM is always Unicode (UTF8)

### Naming Conventions

- All objects in an MDM repository with the name <MDMR> and belong to an SQL schema called <MDMR>\_M<XXX>.
- The STOGROUP of all MDM objects is MDM. It is created when the first repository is created. When you initialize Db2 for SAP MDM, the STOGROUP SAPMDM is created with the VCAT of STOGROUP SYSDEFLT. Alternatively, you can create the STOGROUP yourself.
- All MDM tables are located in databases and tablespaces with the name MDM<NN><MM> . MDMX<NN><MM> where <NN> is a pair of alphanumeric characters, and <MM> a two-digit number. The part of the name <NN> is the same for all objects in a repository.
- The Collection ID is SAPMDMU. The U stands for Unicode.
- The SQLID is SAPMDM.

### MDM Tablespaces and Indexes

The following tables contain the values of the index and tablespace attributes for Db2 in conjunction with MDM.

Index Attributes

Padded Clause	NOT PADDED
FREEPAGE	20
PCTFREE	16
GBPCACHE	CHANGED

DEFINE	ON
BUFFERPOOL	BP2
CLOSE	YES
DEFER	NO
COPY	YES
PIECESIZE	2097152 K
STOGROUP	SAPMDM
Tablespace Attributes	
FREEPAGE	20
PCTFREE	16
GBCACHE	CHANGED
DEFINE	NO
BUFFERPOOL	BP32K
LOCKSIZE	ROW
LOCKMAX	1000000
CLOSE	YES
COMPRESS	YES
MAXROWS	255
SEGSIZE	20
CCSID	UNICODE
STOGROUP	SAPMDM

- MDM adds columns to the tables very frequently. To avoid unnecessary downtime, tablespaces are created with BUFFERPOOL BP32K. The tablespaces are created with COMPRESS=YES.
- The indexes are created in BUFFERPOOL BP2.
- LOB Objects are created with the special register `CURRENT RULES` set to the value `STD`. This implies that the auxiliary tables are created in the database of the base table `SAPMDM<XX>`, and within a Db2 generated tablespace with the **default buffer pool for user data** `ZPARM TBSBPOOL`. The indexes on the auxiliary tables are created with the **default buffer pool for user indexes** `ZPARM IDXBPOOL`.



## **BIND and GRANT**

- The Collection ID `SAPMDMU` has the same attributes as a Collection ID used for the ABAP stack.
- With the GRANT, the same permissions are given as for an ABAP stack with the schema `SAPMDM`.

# A Appendix

## A.1 References

The following section provides information about other sources of information that are useful when installing, configuring, or maintaining your SAP system on Db2 for z/OS.

### → Recommendation

For central access to **all** our documentation, use our [SAP on IBM Db2 for z/OS](#) overview page on SAP Help Portal.

SAP and IBM Documentation

Type of Documentation	Links
Business Continuity for SAP on IBM Z	<a href="https://www.ibm.com/docs/en/bcfsoz">https://www.ibm.com/docs/en/bcfsoz</a>
Db2 12 for z/OS	<a href="https://www.ibm.com/docs/en/db2-for-zos/12">https://www.ibm.com/docs/en/db2-for-zos/12</a>
Db2 13 for z/OS Utility Guide and Reference	<a href="https://www.ibm.com/support/pages/apar/PH4718">https://www.ibm.com/support/pages/apar/PH4718</a>
Db2 Connect 11.5 Documentation	<a href="https://www.ibm.com/docs/en/db2/11.5?topic=db2-connect-overview">https://www.ibm.com/docs/en/db2/11.5?topic=db2-connect-overview</a>
DFSORT Tuning Guide	<a href="https://www.ibm.com/docs/en/zos/2.5.0?topic=dfsorzos-tuning-guide">https://www.ibm.com/docs/en/zos/2.5.0?topic=dfsorzos-tuning-guide</a>
Implementation guides for SAP NetWeaver systems on IBM Db2 for z/OS	<a href="https://support.sap.com/sitoolset">https://support.sap.com/sitoolset</a> >> <b>System Provisioning</b> > <b>System Provisioning Scenarios</b> > Choose >> <b>Install a System using Software Provisioning Manager</b> for installation guides >> <b>Copy a System using Software Provisioning Manager</b> for system copy ( <i>Homogeneous and Heterogeneous System Copy</i> ) guides and >> <b>Rename a System using Software Provisioning Manager</b> for system rename guides
Limits in Db2 for z/OS	<a href="https://www.ibm.com/docs/en/db2-for-zos/13?topic=sql-limits-in-db2-zos">https://www.ibm.com/docs/en/db2-for-zos/13?topic=sql-limits-in-db2-zos</a>
Planning Guide for SAP on IBM Db2 for z/OS	<a href="https://help.sap.com/docs/db2-for-zos/planning-guide">https://help.sap.com/docs/db2-for-zos/planning-guide</a>
SAP on Db2 for z/OS Community	<a href="https://community.sap.com/topics/db2-for-zos">https://community.sap.com/topics/db2-for-zos</a>

### 📌 Note

Contains documentation such as guides for installation, system copy, and SAP system rename (related to the available SAP NetWeaver shipments)

Type of Documentation	Links
SAP Notes	<a href="https://support.sap.com/en/my-support/knowledge-base.html">https://support.sap.com/en/my-support/knowledge-base.html</a>
Security Guide for SAP on IBM Db2 for z/OS	<a href="https://help.sap.com/docs/db2-for-zos/security-guide">https://help.sap.com/docs/db2-for-zos/security-guide</a>
TLS with Client Certificate Authentication for SAP Application Server Connections to Db2 on IBM Z	<a href="https://www.sap.com/documents/2020/10/90ca5a5f-b37d-0010-87a3-c30de2ffd8ff.html">https://www.sap.com/documents/2020/10/90ca5a5f-b37d-0010-87a3-c30de2ffd8ff.html</a>

The following SAP notes are useful when installing, configuring, or maintaining your SAP system on Db2 for z/OS. These SAP notes contain the most recent information on the installation, as well as corrections to the installation documentation.

SAP Note Number	Title
<a href="#">81737</a>	Db2-z/OS: APAR List
<a href="#">183311</a>	Db2/390: Automated PTF Check
<a href="#">913109</a>	Db2-z/OS: SAPCL Patch Collection

Make sure that you have the up-to-date version of each SAP note, which you can find at <https://support.sap.com/en/my-support/knowledge-base.html>.

For more information about **DBA Cockpit with Db2 12**, see SAP note [2302525](#) and the following SAP notes:

SAP Note	Titel
<a href="#">2239553</a>	Db2-z/OS:CCMS: Installation Parameter Settings for Db2 12
<a href="#">2242321</a>	Db2-z/OS:CCMS: Enable 'Installation Parameters' for Db2 12
<a href="#">2251508</a>	Db2-z/OS:CCMS: Db2 12 REORG Mapping Table Changes
<a href="#">2261867</a>	Db2-z/OS:CCMS: Db2 12 RUNSTATS Invalidatecache - changed default behavior
<a href="#">2265560</a>	Db2-z/OS:CCMS: V12 Enablement: DB02 - Lift Partition Limits (PBR RPN)
<a href="#">2266420</a>	Db2-z/OS:CCMS: V12 Enablement: Explain Table Changes
<a href="#">2267378</a>	Db2-z/OS:CCMS: V12 Enablement: Statement Cache
<a href="#">2268434</a>	Db2-z/OS:CCMS: Db2 12 Extended RID Field
<a href="#">2270448</a>	Db2-z/OS:V12: DDIC enhancements

SAP Note	Titel
<a href="#">2284249</a>	Db2-z/OS:CCMS: V12 Enablement: DB02 KPIs - New limit calculations
<a href="#">2302525</a>	Db2-z/OS:v12:CCMS: DBA Cockpit with Db2 12
<a href="#">2302997</a>	Db2-z/OS:v12: Release of Db2 12 for SAP Components
<a href="#">2303000</a>	Db2-z/OS:v12: MDM support of Db2 12
<a href="#">2303010</a>	Db2-z/OS:v12: OEM installation Db2 12
<a href="#">2303020</a>	Db2 z/OS:v12: Upgrade with Db2 12
<a href="#">2303027</a>	Db2-z/OS:v12: Prereqs & preparations for Db2 12
<a href="#">2303029</a>	Db2-z/OS:v12: Migration to Db2 12
<a href="#">2303045</a>	DB2-z/OS:v12: Installation & system copy with DB2 12
<a href="#">2306718</a>	Db2-z/OS:CCMS: Db2 12 REORG TS STATISTICS TABLE with USE PROFILE
<a href="#">2332446</a>	Db2-z/OS:CCMS: V12 Enablement: Virtual and Real Storage
<a href="#">2345842</a>	Db2-z/OS:CCMS: Db2 12 new columns within DSNACCOX
<a href="#">2441303</a>	Db2-z/OS:CCMS: Db2 12 Readiness Check and CLI Failover
<a href="#">2461282</a>	Db2-z/OS:CCMS: Db2 12 Continuous Delivery Levels

For more information about **DBA Cockpit with Db2 13**, see SAP Note [3152992](#) and the following SAP notes:

SAP Note Number	Titel
<a href="#">3093470</a>	DB2-z/OS:CCMS: Installation Parameter Settings for DB2 13
<a href="#">3123091</a>	DB2-z/OS:CCMS: Db2 v13 Enablement - Fix RTS Scalability
<a href="#">3141619</a>	DB2-z/OS: BW support of DB2 V13
<a href="#">3152911</a>	Db2-z/OS:v13: Release of Db2 13 for SAP Components
<a href="#">3152939</a>	DB2-z/OS:v13: Installation & system copy with Db2 13
<a href="#">3152955</a>	DB2-z/OS:v13: Migration to Db2 13
<a href="#">3152970</a>	DB2-z/OS:CCMS: Db2 13 Readiness Check

SAP Note Number	Titel
<a href="#">3152980</a>	DB2-z/OS:v13: OEM installation Db2 13
<a href="#">3152985</a>	DB2 z/OS:v13: Upgrade with Db2 13
<a href="#">3152993</a>	DB2-z/OS:v13: Prereqs & preparations for Db2 13
<a href="#">3289060</a>	DB2-z/OS:CCMS: Db2 13 PRDID change

Check out and participate in our [SAP community for IBM Db2](#). Here you'll find blogs, Q&As, whitepapers, videos, and guides. You can also post blogs and questions and search the community archive for more.

For IBM product documentation on your Db2 version, go to the [IBM Documentation](#).



## A.2 Disclaimer

# Important Disclaimers and Legal Information

## Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

## Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

## Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

## Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

## Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.



© 2024 SAP SE or an SAP affiliate company. 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 SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company 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.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.