Administration Guide | PUBLIC
Document Version: 2.8 – 2023-10-17

Database Administration Guide for SAP on IBM Db2 for Linux, UNIX, and Windows

THE BEST RUN SAP

Content

1 Introduction

This guide provides detailed information about the administration of IBM Db2 for Linux, UNIX, and Windows 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. A basic understanding of the fundamental database concepts and an elementary knowledge of SAP system administration are required.

This documentation applies to SAP systems based on SAP NetWeaver 7.0 and higher on IBM Db2 9.7, 10.1, 10.5, 11.1, and 11.5. To avoid double naming, we mostly speak of "the Db2 database" or just "Db2" when all IBM Db2 for Linux, UNIX, and Windows versions are addressed. For more information, see our Naming Conventions [page 10].

> **i Note**
>
> SAP systems running on IBM Db2 10.1 and lower are out of mainstream maintenance.

Database Administration Essentials

Tasks

Read about the main database administration tasks and when to carry them out

User Management and Security

Learn about system users, groups, role-based security concept, and authentication

Configuration

Find out about variables, profile registry, and parameter settings

Backup and Recovery

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

High Availability (HA)

See what HA solutions and cluster management you can use

Performance

Performance monitoring and tuning you should consider

- Administration Tasks [page 13]
- User Management and Security [page 21]
- Configuration [page 37]
- Backup and Recovery [page 77]
- High Availability [page 146]
- Performance Considerations [page 135]

1.1 Document History

> ⚠ Caution
>
> Make sure you have the latest version of this document that you can find at https://help.sap.com/viewer/db6_admin on SAP Help Portal.

The following table provides an overview of the most important document changes:

Document History

Version	Date	Description
2.8	2023-10-17	Added: Virtual Host Name and IP Address [page 155] including subsections Reuse of Host Name and IP Address of Single Server [page 155] and Setup of a New Virtual Host Name and Virtual IP Address [page 156] Updated: What Is the DBSL? [page 172]
2.7.1	2023-03-06	Info: SAP systems running on IBM Db2 10.1 and lower are out of mainstream maintenance. Minor update of Converting Password Storage in an SAP System to the Secure Storage in the File System [page 33]
2.7	2022-10-10	Information about secure storage in the file system (AS ABAP) was added (see User Authentication Concept for AS ABAP [page 30]).
2.6	2022-06-02	• Added: Fix Pack level check, new functions in the DBA Cockpit for EXPLAIN • Updated: Links to IBM documentation • Added: Link to blogpost about native encryption
2.5	2021-09-23	• Added: Information about the Pacemaker cluster software for high availability (see Cluster Management Software [page 150] and Installing Pacemaker with IBM Db2 [page 151]). • Updated: Links to IBM documentation
2.4.1	2020-11-19	Link updates and minor corrections
2.4	2020-08-27	Update due to the release of Db2 11.5 MP4 FP0SAP. Advanced Logspace Management (ALSM) was introduced. For more information about ALSM, see Components of the Db2 Log File Management [page 79].

Version	Date	Description
2.3	2019-11-05	Update due to new Db2 version 11.5
2.2	2018-07-27	With Db2 11.1 Mod 3 FP3 iFix001SAP, there's a new default parameter setting for vendor log archive timeouts, see Configuration of the Db2 Log File Management [page 82] and Other Storage Vendors [page 89]. Sections Updating the Db2 Fix Pack Level [page 159] and Rolling Update of the Db2 Fix Pack Level [page 160] were added.
2.1	2018-01-02	• Information about Db2 V9.1 and V9.5 removed because these Db2 versions are no longer supported as of 12-31-2017. • Links to SAP Service Marketplace replaced by links to the new SAP Help Portal • Graphics update
2.02	2017-08-22	• Change of database name by IBM to *IBM Db2* (formerly IBM DB2 for Linux, UNIX, and Windows), see Naming Conventions [page 10]. • As of Db2 11.1 MP2 FP2: No more lock escalations due to new Db2 registry variable `DB2_AVOID_LOCK_ESCALATION` • Information about database partition groups was added to chapter Tablespaces, Containers, and the File System [page 53]
2.01	2017-03-16	Correction of links
2.0	2017-2-21	Minor updates/corrections: • Configuring Tivoli Storage Manager (TSM) / IBM Spectrum Protect [page 223] (Renaming of IBM Tivoli Storage Manager to IBM Spectrum Protect) • Checking the Database for Consistency [page 123] (new default setting of registry variable as of Db2 11.1 FP1) • Some graphics were updated.
1.90	2016-08-16	Updated version due to new release of Db2 for Linux, UNIX, and Windows Version 11.1
1.80	2015-05-19	General update with content and language corrections For example, the following sections were added or updated: Checking the Database for Consistency [page 123], Locking Concepts [page 129], Locking Mechanisms in an SAP Environment [page 130], Db2 Memory Management Db2 Memory Management [page 41], Using Automatic REORG [page 142], Self-Tuning Memory Management (STMM) [page 44], Redirected Restore [page 125], Updating Statistics Using Automatic RUNSTATS [page 139], and so on.
1.70	2013-11-29	Section Graceful Maintenance Tool (GMT) [page 192] was added.
1.60	2013-07-26	This version has been enhanced with content referring to the newly released version 10.5 of the Db2 database.
1.50	2013-05-10	Update: Information about SAP NetWeaver 7.4 added.

Version	Date	Description
1.40	2012-10-25	Updated Version This version has been enhanced with content referring to the newly released version 10.1 of the Db2 database.
1.30	2011-11-21	Updates: • New role-based security concept • Minor corrections
1.20	2011-04-12	Updated Version • Updates due to new monitoring functions in the DBA Cockpit For more information, see *Performance Considerations* (Performance Considerations [page 135]). • Section on Converting Tables Using Report DB6CONV [page 75] was updated with information about the new version of report DB6CONV. • Separation of duties of datatabase administration users
1.10	2009-12-15	Updated Version This version has been enhanced with content referring to the newly released **version 9.7** of the Db2 database.
1.0	2008-06-06	Initial Version: The content of this guide refers to the database **version 9.1** and **version 9.5** of IBM Db2 for Linux, UNIX, and Windows and is valid for SAP NetWeaver 7.0 and higher and all SAP systems based on these releases.

1.2 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*.

IBM Terminology

The database versions are referred to as follows:

Database Name	Abbreviation
IBM Db2 Version 11.5 for Linux, UNIX, and Windows	Db2 11.5
IBM Db2 Version 11.1 for Linux, UNIX, and Windows	Db2 11.1
IBM Db2 Version 10.5 for Linux, UNIX, and Windows	Db2 10.5
IBM Db2 Version 10.1 for Linux, UNIX, and Windows	Db2 10.1
IBM Db2 Version 9.7 for Linux, UNIX, and Windows	Db2 V9.7

With Db2 11.1, IBM introduced the concept of Modification Packs. A Modification Pack (also referred to as Mod, Mod Pack, or MP) introduces new functions to the Db2 product. For the IBM Db2 Modification Packs and Fix Packs, we mostly use abbreviations such as Db2 11.1 Mod 2 Fix Pack 2, or even shorter, simply Db2 11.1 MP2 FP2.

Renaming: IBM DB2 for Linux, UNIX, and Windows is now IBM Db2

IBM has changed its database name from IBM DB2 for Linux, UNIX, and Windows to simply IBM Db2 (with a lowercase 'b' now in Db2). In older SAP publications, you'll still find the old product name, but in more recent documentation, we'll use the new term, sometimes extended by 'for Linux, UNIX, and Windows' to avoid confusion with other products of the Db2 family, such as Db2 for z/OS or Db2 for i.

Db2 Database Partitioning Feature

The Db2 Database Partitioning Feature is referred to as *DPF*.

IBM Spectrum Protect (Tivoli Storage Manager, TSM)

After the IBM storage product ADSM was renamed to *TSM* (Tivoli Storage Manager), there has been another name change: As of Version 7.1.3, IBM Tivoli Storage Manager is now *IBM Spectrum Protect*. The names are often used interchangeably, but in this document the product is referred to as *TSM*. Note that the directory name `sqllib/adsm` hasn't changed and still contains the old product name.

Other Terminology

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

2 Administration Tools and Tasks

2.1 Administration Tools

You can use the following tools to administer your Db2 database:

- DBA Cockpit
- Db2 Command Line Processor (CLP)

DBA Cockpit

The DBA Cockpit is the preferred tool to use for database administration and monitoring tasks in an SAP environment. It's a platform-independent tool that is part of SAP NetWeaver systems and integrated into SAP Solution Manager. You can run the DBA Cockpit as part of your system administration activities in SAP Solution Manager.

To access the DBA Cockpit, call transaction `DBACOCKPIT` in your SAP system.

> **i** Note
>
> Since the DBA Cockpit is integrated in the SAP system, it is **not** available once the SAP system is offline.

The support of database release-specific features in the DBA Cockpit depends on your SAP Basis release and Support Package (SP). For more information, see the *Upgrade Requirements* section in the database upgrade guide for your Db2 version. Our Db2 upgrade guides are available on SAP Help Portal, see References [page 228] in the appendix for more information.

Database release-specific functions of the DBA Cockpit that are referenced in this document are available according to the minimum requirements listed in the above mentioned upgrade guides.

For more information about using the DBA Cockpit, see the separate document *Database Administration Using the DBA Cockpit: IBM Db2 for Linux, UNIX, and Windows* in References [page 228].

Db2 Command Line Processor (CLP)

The Db2 CLP is always available on the database server. For SAP systems that do not use the Db2 Client Connectivity [page 19], the Db2 CLP is also installed on each application server.

For more information, see Command line processor features in the IBM documentation.

2.2 Administration Tasks

Here's an overview of administration tasks that are required to ensure that the database operates well over time:

Initial Setup

- Develop a backup and recovery strategy. For more information, see Developing a backup and recovery strategy in the IBM Db2 documentation.
- If you haven't already done so, enable your database for rollforward recovery [page 77].
- Make sure that your database configuration is correct, which includes the Db2 registry, the database manager configuration, and the database configuration. For more information, see the appropriate sections under Configuration [page 37].
- Familiarize yourself with the periodic tasks that are described in the following, and perform all the steps at least once.

Periodic Tasks

Periodic database administration tasks include recoverability, storage management, performance monitoring, and system health:

Recoverability

- Perform regular database backups using the DBA Planning Calendar of the DBA Cockpit. For more information, see also Database Backup [page 109] and the DBA Cockpit documentation in References [page 228].
- Make sure that log file archiving is enabled and working properly. For more information, see Db2 Log File Management [page 79].

Storage Management

- Make sure that your tablespaces have enough free space available. If the tablespaces are enabled for Db2's automatic storage management or for the automatic resize function, check if there is enough free space in the file systems where the containers of the tablespaces reside.
  For more information, see Tablespaces, Containers, and the File System [page 53].
- The diagnostics log file and the notification log file can grow infinitely. Therefore, you must truncate them where needed. Alternatively, you can enable rotating diagnostic logs using the `DIAGSIZE` parameter. For more information, see Dealing with Growing Diagnostic Data [page 75].

Performance Monitoring

- Configure Db2 automatic maintenance to keep statistics for tables and indexes up-to-date so that the optimizer can choose optimal access plans. For more information, see Updating Statistics for Database Tables [page 138].
- Configure Db2 automatic table maintenance so that regular table and index maintenance operations are executed by Db2. For more information, see Reorganization of Database Objects [page 141].

- Check the monitors for exceptions, for example, overflows, lock escalations, and deadlocks. In addition, check the quality of memory caches of the database, for example, buffer pools, package cache, catalog cache, and dynamic SQL statements. For more information, see the appropriate sections under Performance Considerations [page 135].

System Health

Check the diagnostics log file `db2diag.log` for errors on a regular basis. To do so, you can use the `db2diag` tool. To display a list of all normal and severe errors, enter the following command:

```
db2diag -l error,severe
```

For more information, see Diagnostic Tool db2diag [page 164].

> **i Note**
>
> As of Enhancement Package 2 and Support Package 7 for SAP Netweaver 7.0, you can also analyze the message logs using the message log viewer in the diagnostics area of the DBA Cockpit. For more information, see the DBA Cockpit documentation [page 228].

Emergency Tasks

In the case of an unexpected event, for example, an unexpected database crash, proceed as follows:

1. Do not panic under any circumstances.
2. Get a clear understanding of the nature of the problem. To do so, try to answer the following questions:
   - Can the problem be reproduced?
   - Which components are exactly involved (for example, the operating system, the database itself, SAP system components, client, or server)?
   - Is the system still operable?
3. Check the `db2diag.log` and try to find an indication of the root cause.
4. Try to correct the root cause of the problem.
5. Collect data using the `db2support` tool.
   In most cases, it's sufficient to enter the following command:
   ```
   db2support <output directory> -d <database name> -g -c -s -f
   ```
   A file (`db2support.zip`) is created in the output directory that contains the relevant data for support purposes.
   If you cannot connect to the database, omit option `-c`.
6. If you are not able to solve the problem or if you require clarification about the root cause of the problem, open a customer incident and attach the collected data (see step 5).

Emergency Scenarios

There are some emergency scenarios that you should be able to handle yourself, for example:

- Hardware failure
  If you lose the entire database server, you need to be able to restore the operating system and to recover your database.
- Disk failure
  If you lose all or parts of the disks that contain database files, you need to be able to recover your database.

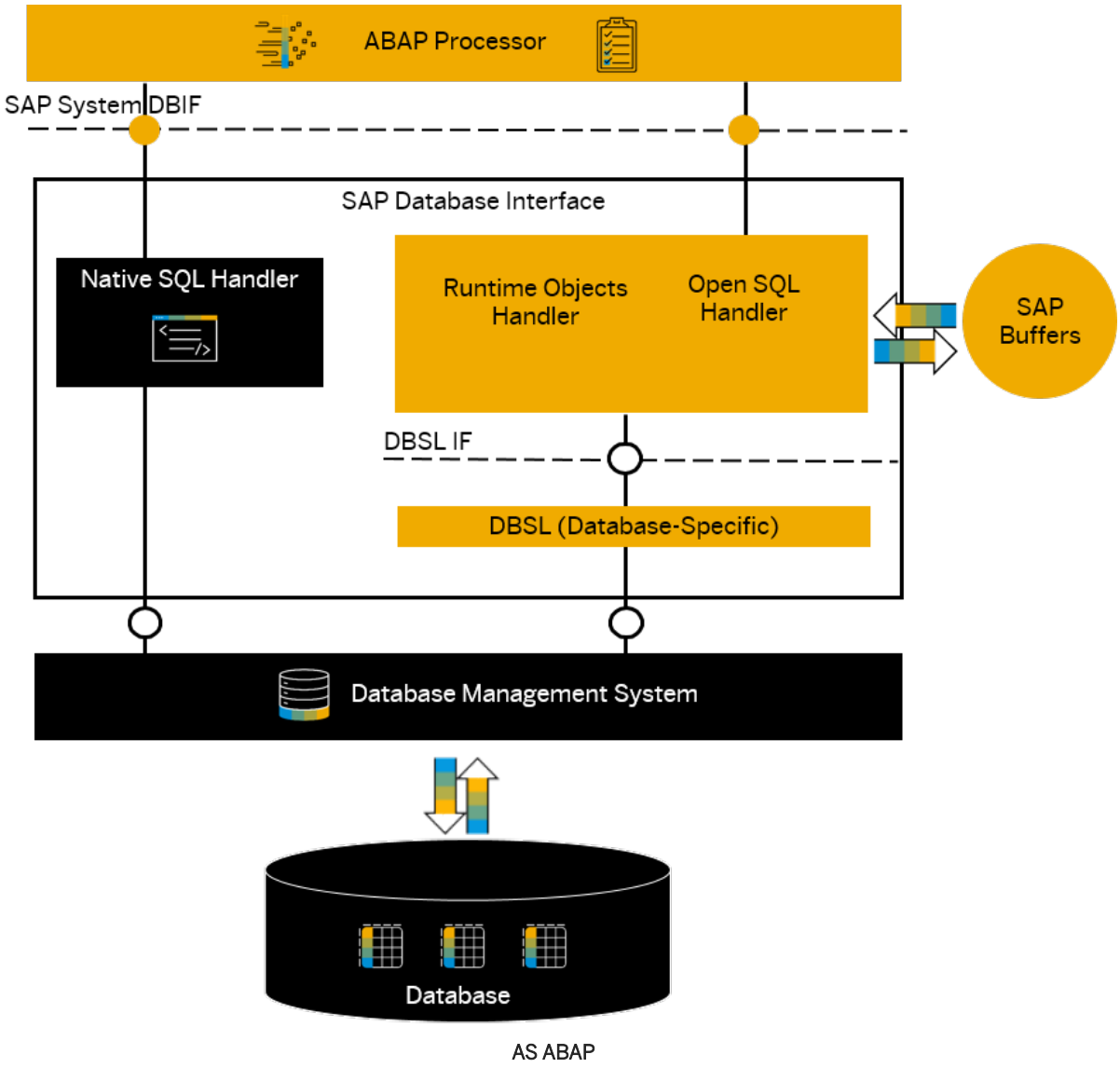
- Recovery from logical errors
  This might become necessary if a table was dropped or rows of a table were deleted by mistake. Make yourself familiar with the concept of a point-in-time recovery of your data.

> **i Note**
>
> The recovery from such emergency scenarios needs planning and training. Make sure that you document the procedure and test it on a regular basis.

3 Architectural Overview

3.1 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 (`dbdb6slib.*`) 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.

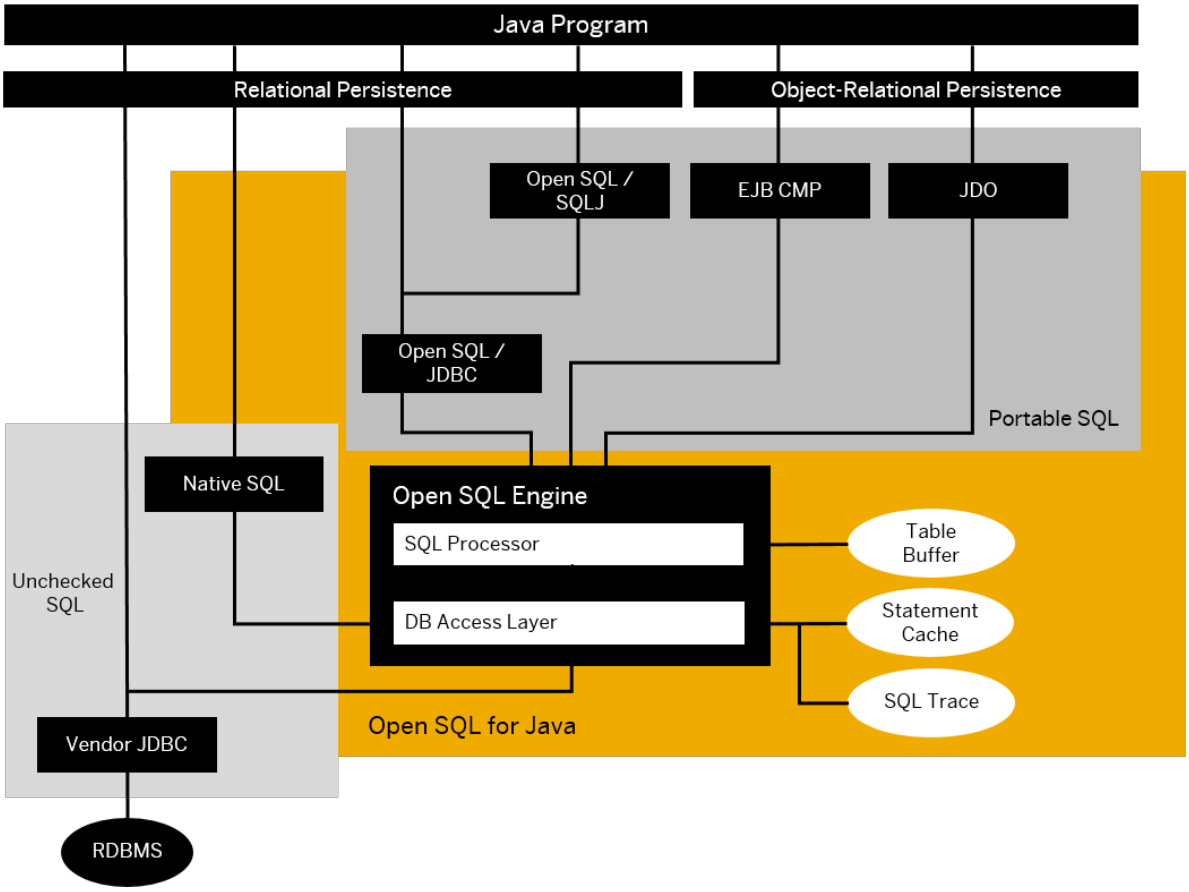
More Information

Db2 Client Connectivity [page 19]

Introduction to Db2 Call Level Interface and ODBC in the IBM documentation.

3.2 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 Java can use various standardized APIs to access the database, for example, JDO, SQLJ, or JPA.

The database interface offers Java applications 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 IBM Data Server Driver for JDBC and SQLJ - a pure Java Type 4 JDBC driver that is based on distributed relational database architecture (DRDA) and uses TCP/IP as network protocol.

3.3 Db2 Components

With regard to Db2 itself, we distinguish between the following components:

- The Db2 software (the Db2 binary code)
  You can have multiple Db2 software installations on one host.

- The database itself, which contains the data and is managed by the DBMS
- The Db2 instance that serves as a layer between the Db2 software and the database
  An instance provides an independent environment where database objects can be manipulated and applications can run.

> **i** Note
>
> In an SAP system installation, there is a one-to-one relation between the Db2 instance and the Db2 database. You can have a local software installation for each SAP system. With this one-to-one relationship between Db2 software and Db2 instance, you can maintain each SAP system independently.

3.4 Db2 Client Connectivity

To connect to the database, the SAP application server requires the following components:

- Db2 CLI driver for the ABAP stack
- JDBC driver for the Java stack
- Database name and connection port for the primary database

The following applies to all SAP releases with SAP Basis 7.0 SP13 and higher running on Db2.

Directory Structure of the Database Client for an SAP System

The Db2 CLI driver and the Db2 JDBC driver files are located in a **shared** directory. Each SAP application server can use the driver files directly from this directory or copy them to a local directory on the application server during startup. This setup simplifies the software maintenance because you have to keep the driver files only in the shared directory.

Since the Db2 CLI driver does not provide its own database catalog, the connection information for the default database of your SAP systems is stored in the file `db2cli.ini`. As of SAP NetWeaver 7.0 SP13 and higher, all SAP systems are installed with this setup by default.

The following figure shows the directory structure of the Db2 client connectivity in a newly installed ABAP+Java system on a UNIX operating system:

- global/db6
  - db2cli.ini
  - jdbc
    - db2jcc.jar (up to Db2 11.1 only)
    - db2jcc4.jar
    - jdbcdriver.lst
  - db2dump
  - <OS>/db6_clidriver, where <OS> is AIX_64, HP11_64 or HPIA64 (up to Db2 10.5 only), LINUXX86_64, LINUXPPC64_64 (up to Db2 10.5 only), or SUNOS_64 (up to Db2 10.5 only)

The Db2 driver files are located in directory `/usr/sap/<SAPSID>/SYS/global/db6`. During the startup of the application server, the Db2 driver files are copied by the utility `sapcpe` to a local directory on the application server, for example, `/usr/sap/<SAPSID>/<Instance Name>/exe`. The call of the utility `sapcpe` can be found in the SAP instance profile.

- `<Instance Name>`
  - log
  - data
  - work
  - exe
    - db6_clidriver
    - db2jcc.jar (up to Db2 11.1)
    - db2jcc4.jar

With this copy mechanism, you can maintain and exchange the Db2 driver files in the shared directory while the application servers are running. The SAP application servers automatically use the new driver files after the next restart.

The main release level of the Db2 CLI driver must match the one of the Db2 software release level on the database server. The Db2 CLI driver can have a Fix Pack level lower than the Fix Pack level that is used on the database server.

During the installation of the database instance, the current SAP installation tool automatically installs the Db2 CLI driver for the operating system of the database server. If you install a new application server, the Db2 CLI driver for this operating system is also automatically added by the installer if it's not already available in the global directory.

For older SAP systems on SAP Basis 7.0, see SAP Note 1091801.

The database shared library (DBSL) looks for the Db2 CLI libraries in the following directories in exactly the sequence given here:

1. If set in the environment: `DB2_CLI_DRIVER_INSTALL_PATH`
2. If it exists: `/usr/sap/<SAPSID>/SYS/global/db6/<OS>/db6_clidriver`
3. In the instance directory specified by the variable `DB2INSTANCE` (for example, `~$DB2INSTANCE/sqllib/lib` on UNIX)
4. In the directories that are specified by the environment settings for the operating system library path

> ⚠ Caution
>
> The Db2 libraries are loaded by the DBSL dynamically. Do **not** include the path to the Db2 libraries in the operating system library path setting. If you do so, you might accidentally overwrite the default search path that is built into the Db2 libraries.

More Information

SAP NetWeaver installation guides on SAP Help Portal

4 User Management and Security

Learn about users and groups in an SAP environment as well as security mechanisms that are used to secure users and passwords.

Note that Db2 requires no user management of its own. Instead, the authentication mechanisms of the operating system are used. Therefore, all users and groups mentioned in this guide are operating system users or groups.

Related Information

SAP System Users and Groups [page 21]
Role-Based Security Concept for Database Users [page 27]
Database Authentication [page 29]
User Authentication Concept for AS ABAP [page 30]
User Authentication Concept for AS Java [page 35]
Other Security Features: SSL, Native Encryption, and LDAP [page 35]

4.1 SAP System Users and Groups

The following tables list the users and groups that are automatically created by the SAP installation tool during the SAP system installation unless they already exist.

SAP System Users

User	Description
`db2<dbsid>`	**Database administrator** This user is the Db2 instance owner and SAP database administrator and has `SYSADM` , `SECADM`, and `DBADM` authorization.
`<sapsid>adm`	**SAP system administrator** This user is authorized to start and stop the SAP system and the Db2 Database Manager (instance). `<sapsid>adm` has the Db2 authorizations `DBADM` and `SYSCTRL` that are required by Db2-specific monitoring functions that were started by SAP application server functions.

User	Description
• `sapr3` SAP systems that were first installed with release 4.6D or lower • `sap<sapsid>` SAP systems based on AS ABAP Kernel Release 6.10 or higher and additional MCOD 4.6D SAP systems • `sap<sapsid>db` SAP systems based on AS Java • `<Other user name>` SAP systems created by a system rename or system copy	**Database connect user** This user owns all SAP database objects (tables, indexes, and views) and additionally has the `SYSMON` authorization. All database connection and instance access operations for an SAP application server are performed with this user. This user is **only** created on SAP systems on which the SAP system database has been installed (not on remote application servers). To determine the database connect user, check the environment of the `<sapsid>adm` user: • If the environment variable `dbs_db6_user` exists, it contains the name of the database connect user. • If `dbs_db6_user` does not exist, the environment variable `dbs_db6_schema` contains the name of the database connect user. • If both environment variables do not exist, the name of the database connect user is `sapr3`. The database connect user requires at least the database authorizations `CREATETAB`, `BINDADD`, `CONNECT`, and `IMPLICIT_SCHEMA`. The user also needs access to the SAP system tablespaces belonging to its `<SAPSID>`. By default, access to SAP system tablespaces is granted to `PUBLIC`, that is, tablespaces can be accessed by all users that have `CONNECT` authorizations. **i Note** **Java only:** By default, only the tablespaces `<SAPSID>#DBD`, `<SAPSID>#DBI`, and `<SAPSID>#DBL` are used by the Java stack.

User	Description

Windows only:
`SAPService<SAPSID>`
or
`sapse<SAPSID>`
	SAP service account user
	This user is a virtual user. On Windows, the SAP system is usually started with this user account, but there is no need to log on to the SAP system with it.
	This user account must have the local user authorizations to log on as a service and has to be a member of the local administrator group. The name of this user must be `SAPService<SAPSID>`.
	i Note
	Up to and including SAP NetWeaver 7.0 SR1, user `sapse<sapsid>` or `SAPService<sapsid>` must be a member of group `Administrators`.
	As of SAP NetWeaver 7.0 SR2, the SAP service account user must be a member of the extended security group for Db2 administrators, which can be, for example, `<domain>\DB2ADMNS_<DBSID>`, `<hostname>\DB2ADMNS_<DBSID>`, or `DB2ADMNS` (depending on your system environment.) It needs no longer be a member of the Windows `Administrators` group.

SAP System Groups

Groups	Description
`db<dbsid>adm`	**Database system administration group** This group assigns the `SYSADM` authorization. Each member of this group has the `SYSADM` authorization for the Db2 Database Manager instance. This is the highest level of authorization within the Db2 Database Manager and controls all database objects. User belonging to this group: `db2<dbsid>`
`db<dbsid>ctl`	**Database system control group** This group assigns the `SYSCTRL` authorization. Each member of this group has the `SYSCTRL` authorization for the Db2 Database Manager instance. With `SYSCTRL` authorization, operations affecting system resources are allowed but direct access to data is not allowed. User belonging to this group: `<sapsid>adm` (secondary group)

Groups	Description
db<dbsid>mnt	**Database maintenance group** This group assigns the SYSMAINT authorization. Each member of this group can perform maintenance operations on all databases associated with an instance. Members of this group are not allowed to directly access data, but their authorization includes, for example, updating the database configuration, performing database or tablespace backups, and restoring an existing database.
db<dbsid>mon	**Database monitoring group** For SAP systems based on Kernel 7.20, the db<dbsid>mon group replaces the db<dbsid>mnt group. Users of this group have the authorization to monitor the database. Users belonging to this group: sapr3, sap<sapsid>, and sap<sapsid>db (database connect user) **i Note** The database connect user is also the owner of all database objects and therefore can access them (see also Role-Based Security Concept for Database Users [page 27]).
sapsys	**SAP system administration group** Each member of this group can act as an SAP system administrator. User belonging to this group: <sapsid>adm (primary group)
Windows only: SAP_<SAPSID>_GlobalAdmin	**Domain-level SAP system administration group** This group is used for grouping the SAP system administrators. The sole function of a global group is to gather users together at domain level so that they can be placed in the appropriate local groups. The members of this group are the domain users <sapsid>adm and sapse<sapsid>. The group SAP_<SAPSID>_GlobalAdmin is only used when the SAP system belongs to a Windows domain. The group SAP_<SAPSID>_GlobalAdmin is not required for a local installation.
Windows only: SAP_<SAPSID>_LocalAdmin	**Local group on an application server** Only local groups are created and maintained on an application server. A local group can only be given authorizations for the system where it is located. If the system is part of the domain, the local group can contain users and global groups from the domain.

Groups	Description
Windows only: DB2ADMNS or <domain>\DB2ADMNS_<DBSID> or <hostname>\DB2ADMNS_<DBSID> (for installations with local users)	**Extended security group for Db2 administrators** The following users must be members of this group: • db2<dbsid> • SAPservice<sapsid>
Windows only: DB2USERS or <domain>\DB2USERS_<DBSID> or <hostname>\DB2USERS_<DBSID> (for installations with local users)	**Extended security group for Db2 users**

4.1.1 Access Authorizations for Db2 Directories and Database-Related Files

UNIX: Access Authorizations for Directories and Files

Db2 Directory or File	Access Privilege in Octal Form	Owner	Group
Home directory of user db2<dbsid> (/db2/<DBSID> or /db2/db2<dbsid>)	755	db2<dbsid>	db<dbsid>adm
Database software installation path: /db2/<dbsid>/db2_software This is the default installation path after a fresh install. It may change later if a new software version is installed. You can display the path using **db2level** as user db2sid.	755	root	Primary group of user root depends on your operating system
/db2/>DBSID>/log_dir	750	db2<dbsid>	db2<dbsid>adm
/db2/<DBSID>/db2dump	750	db2<dbsid>	db2<dbsid>adm

Db2 Directory or File	Access Privilege in Octal Form	Owner	Group

Directories where database containers can be located:
`/db2/<DBSID>/sapdata*`
`/db2/<SAPSID>/sapdata*`
`/db2/<DBSID>/saptemp*`
	750	`db2<dbsid>`	`db2<dbsid>adm`

`/usr/sap/<SAPSID>/SYS/global/dscdb6.conf`	640	`<sapsid>adm`	`sapsys`

LINUX: Access Authorizations for Directories and Files

Directory	Access Privilege	Owner	For User or Group

`<drive>:\db2<dbsid>`	Full Control	Administrator	`SAP_<SAPSID>_LocalAdmin`, System, `DB2USERS`
			or
			`<domain>\DB2USERS_<DBSID>`, `DB2ADMNS`
			or
			`<domain>\DB2ADMNS_<DBSID>`

`<drive>:\db2<dbsid>\db2_software`
(database software automatically installed by the SAP installer)
	Full Control	Administrator	`DB2USERS`
			or
			`<domain>\DB2USERS_<DBSID>`, `DB2ADMNS`
			or
			`<domain>\DB2ADMNS_<DBSID>`

Either of the following:
• `<drive>:\db2<dbsid>\DB2<DBSID>` (database software automatically installed by the SAP installer)
• `<drive>:\<path_to_db2_software>\DB2<DBSID>` (database software was manually installed because automatic installation had not been integrated yet in the SAP installation tool)
	Full Control	Administrator	`DB2USERS`
			or
			`<domain>\DB2USERS_<DBSID>`, `DB2ADMNS`
			or
			`<domain>\DB2ADMNS_<DBSID>`

Directory	Access Privilege	Owner	For User or Group

Either of the following:
• `<drive>:\db2\<DBSID>\DB2<DBSID>` (database software automatically installed by the SAP installer)
• `<drive>:\DB2<DBSID>` (database software was manually installed because automatic installation had not been integrated yet in the SAP installation tool)
	Full Control	Administrator	`DB2ADMNS`
			or
			`<domain>\DB2ADMNS_<DBSID>`

`<drive>:\db2`	Full Control	Administrator	Everyone
`<drive>:\db2\<dbsid>\log_dir`	Full Control	Administrator	`Db2<dbsid>`, System
`<drive>:\db2\<dbsid>\db2dump`	Full Control	Administrator	`SAP_<SAPSID>_LocalAdmin`, System

4.2 Role-Based Security Concept for Database Users

As of SAP NetWeaver 7.0 on Db2 9.7 or higher, you can use database roles to restrict user authorizations on IBM Db2 according to organizational tasks. This is particularly relevant for the following use cases:

- **Restriction of user authorizations**
  You can exclude the administration user from access to application data (“separation of duties”). You identify database administration duties and provide each individual database administrator with their own user ID and with an authorization as minimal as possible to complete his or her daily tasks. You can reuse and adapt the default roles shipped by SAP to perform these tasks.
- **Change tracking**
  If you create individual users for all database administrators, this allows you to track the changes performed by database administrators on individual user account level.

For more information, see the article Role-Based Security Concept for Database Users on IBM Db2 for Linux, UNIX, and Windows on SAP Community.

Related Information

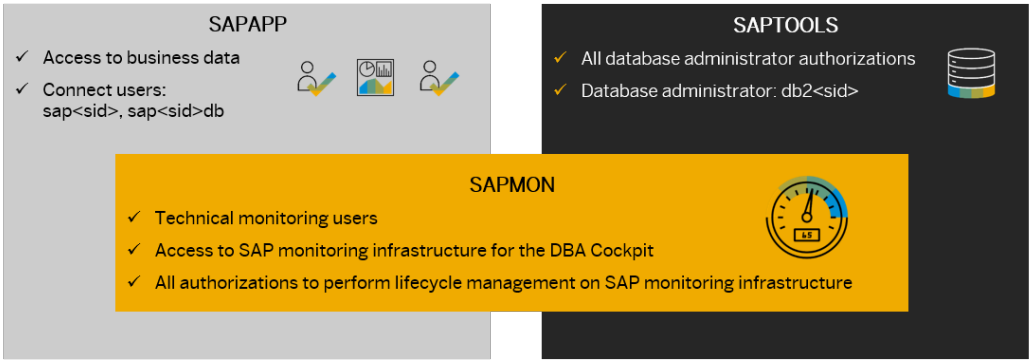
Database Roles for SAP System Environments [page 28]
Activating the Role-Based Security Concept [page 28]

4.2.1 Database Roles for SAP System Environments

The following database roles are available for SAP systems:

- **SAPAPP role**
  The SAPAPP role is the role for business applications. By default, it is assigned to all connect users, including the connect users of business applications.
- **SAPMON role**
  The SAPMON role is designed for monitoring. It has all authorizations for the monitoring APIs provided by Db2. In addition, users with this role have all authorizations to set up and maintain the monitoring infrastructure provided by SAP, especially the DBA Cockpit.
- **SAPTOOLS role**
  For database administrators, the SAPTOOLS role is available. It has the most powerful authorizations to perform all administrative tasks.

The SAPMON role is included in the SAPAPP and SAPTOOLS roles.



SAPAPP, SAPTOOLS, and SAPMON Roles

> **i Note**
>
> The <sapsid>adm user is not assigned to any of the new database roles. The <sapsid>adm user still belongs to the database group SYSCTRL, so administrators with this user can start and stop the database server.

4.2.2 Activating the Role-Based Security Concept

Use

Automatic Activation of the Role-Based Security Concept for SAP Systems

As of SAP NetWeaver 7.0 SR3 and IBM Db2 9.7, all new SAP system installations work with the new role-based security concept. The SAP installer creates the roles automatically and does not assign any single user authorizations anymore. The installer also assigns the SAP default users to their appropriate database roles.

Manual Activation of the Role-Based Security Concept

If you need to activate the role-based security concept manually, you can use the `db6_update_db.[sh|bat]` tool (see SAP Note 1365982). For example, after you have performed an SAP system upgrade, you can also run the `db6_update_db` script to get the same default setup.

When you execute the `db6_update_db` tool, it does not automatically change the authorization concept. It always enforces the concept for which the database is configured. This means that if you have a traditional system with user-specific authorizations, it does not convert the authorization concept of the system into an authorization concept based on database roles. If you have a system configured for database roles, it ensures that all SAP-specific authorizations are assigned based on database roles. After a Fix Pack update or a Db2 version upgrade, the `db6_update_db` tool can be used to repair the authorizations. It always enforces the database authorization concept that is currently active.

Procedure

To enable the role-based security concept for an existing database manually, call the `db6_update_db` script as follows:

- UNIX:
  ```
  db6_update_db.sh –d <dbsid> -enable_roles
  ```
- Windows:
  ```
  db6_update_db.bat –d <dbsid> -enable_roles
  ```

4.3 Database Authentication

The Db2 database is always installed with one of the following database parameters:

- `Authentication = SERVER`
  The user ID and password provided on connect or attach are verified by Db2 using operating system services on the database server.
- `Authentication = SERVER_ENCRYPT`
  This parameter provides a higher level of security since passwords are sent encrypted across the network. We recommend that you use this setting. It is supported by all currently supported database versions.

Authenticating a User in a Windows Environment

To authenticate a user, Db2 searches the Windows security database in the following sequence:

1. It searches for the user in the local security database on the database server.

2. If the user is not found, Db2 searches in the security database of the Primary Domain Controller in the current domain.
3. If the user is still not found, Db2 searches in the security databases of all trusted domains until either the user is located or all the security databases have been searched.

Db2 provides a new registry variable that determines where Db2 searches for the following groups:

- `SYSADM_GROUP`
- `SYSCTRL_GROUP`
- `SYSMAINT_GROUP`

To enable Db2 to identify all groups correctly, the registry variable `DB2_GRP_LOOKUP` has to be set to `TOKEN`.This is automatically covered by the `DB2_WORKLOAD=SAP` setting (see section *Aggregate Registry Variable DB2_WORKLOAD* in DB2 Profile Registry [page 39].

4.4 User Authentication Concept for AS ABAP

SAP systems on an IBM Db2 database use a connect user and its password in the operating system to connect to the database of the application server ABAP. Most SAP executables use the DBSL to open such connections (for example, disp+work, R3trans, tp). However, there are also some standalone tools like `db6util`. These programs need a way to retrieve the operating system password from a secure location.

Since the SAP system must connect to the database without asking for a password, the user ID and password for the user `sap<sapsid>` are centrally stored and maintained in a password storage file. There are two options for storing password files: the old `dscdb6.conf` file and the newer ABAP secure storage in the file system (for SAP NetWeaver systems 7.5 and higher as of SL Toolset 1.0 SP36).

Both can be accessed from all application and database servers that use NFS (UNIX) or Windows shares and are protected against unauthorized access using file-system access authorizations. User passwords are stored in encrypted form.

dscdb6.conf File

The password length is limited to 16 characters.

The `dscdb6.conf` file is stored in the following directory:

Operating System Platform	Directory
UNIX	`/usr/sap/<SAPSID>/SYS/global/dscdb6.conf`
Windows	`\\%DSCDB6HOME%\sapmnt\<SAPSID>\SYS\global\dscdb6.conf`

> **i Note**
>
> In a Windows-only environment, the environment variable `<DSCDB6HOME>` contains the name of the server where the global directory is located. Typically, this server is the primary application server.

In a system environment where the database server is running on an operating system other than Windows, `<DSCDB6HOME>` should contain the name of the server where you can access the file or path of the password storage using the path listed above.

Secure Storage in the File System (AS ABAP)

Secure storage in the file system of the AS ABAP is available for SAP systems 7.50 and higher running on IBM Db2 for Linux, UNIX, and Windows. It's used for systems that are installed using SL Toolset 1.0 SP 36 and higher. The minimum SAP Kernel requirement is 7.49.

The password length is limited by the DBSL layer and can be up to 64 characters. In the ABAP secure storage, the name of the database connect user and its password are stored.

The ABAP secure storage is located in the following directory:

Operating System Platform	Directory
UNIX	`/sapmnt/<SAPSID>/global/security/rsecssfs`
Windows	`\\<global host>\sapmnt\<SAPSID>\SYS\global\security\rsecssfs`

Related Information

Managing Passwords (dscdb6.conf) [page 31]
Managing Passwords (Secure Storage) [page 32]

4.4.1 Managing Passwords (dscdb6.conf)

Learn how you can manage passwords in the file `dscdb6.conf` using the command-line tool `dscdb6up`.

Context

> **i Note**
>
> Managing passwords using the file `dscdb6.conf` is only relevant for AS ABAP systems that were installed using software provisioning manager 1.0 SP 35 and lower. As of SP 36, SAP secure storage is available (see User Authentication Concept for AS ABAP [page 30]).

If you inadvertently deleted or destroyed the file `dscdb6.conf`, or if you updated the operating system password of the database connect user, you can re-create it by using the command-line tool `dscdb6up`.

You can update the passwords of the ABAP connect user (by default, `sap<sapsid>`) as follows:

Procedure

1. Log on to the database server as user `<sapsid>adm`.
2. On the command line, enter the following command:
   `dscdb6up -create <password of the ABAP connect user> none`

   > **i** Note
   >
   > Previously, the file `dscdb6.conf` was also used to store the password of the `<sapsid>adm` user. This password is no longer used. Because you still must provide two values to fulfill the command syntax of `dscdb6up`, you can use any value for the `<sapsid>adm` password, such as the value **none**.

   `dscdb6up` updates the content of the `dscdb6.conf` file with the new passwords in encrypted format. The operating system passwords aren't changed by `dscdb6up`.
3. Update the operating system password on the database server accordingly using operating system tools. Note that in a multipartition environment with multiple hosts, the password needs to be updated on every host.

More Information

dscdb6up - Tool to Set and Update Passwords [page 218]

4.4.2 Managing Passwords (Secure Storage)

If secure storage is used, use the command line tool `rsecssfx` to manage passwords for connect users.

Prerequisites

You need an SAP system based on SAP Kernel 7.49 and higher. You have installed the system using SL Toolset 1.0 SP 36 or higher, or you have manually converted the password storage to secure storage (see Converting Password Storage in an SAP System to the Secure Storage in the File System [page 33]).

Context

When you update the operating system password of the connect user, you must also update the stored password in the secure storage in the file system using the command line tool `rsecssfx`, which is delivered as part of the kernel executable archive.

Procedure

1. Log on to the database server as user `<sapsid>adm`.
2. On the command line, enter the following command:

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

   `rsecssfx` updates the content of the secure storage in the file system with the new passwords in encrypted format. The operating system passwords are not changed by `rsecssfx`.
3. Update the operating system password on the database server accordingly using operating system tools.

   > **i** Note
   >
   > In a multi-partition environment with multiple hosts, you must update the password on every host.

Related Information

rsecssfx - Tool to Create and Update Secure Storage in the File System [page 219]

4.4.3 Converting Password Storage in an SAP System to the Secure Storage in the File System

You can manually convert a previously installed SAP system with the `dscdb6.conf` password storage to the secure storage in the file system. You may want to do this, for example, if you want to use passwords longer than the 16-character limit that is available with `dscdb6.conf`.

Prerequisites

For AS ABAP systems running on IBM Db2 for Linux, UNIX, and Windows, you can use secure storage in the file system to store the database connect user name and password. You need at least Kernel version 7.49 to be able to use the secure storage.

Context

As of software provisioning manager 1.0 SP 36, AS ABAP systems running on IBM Db2 for Linux, UNIX, and Windows are automatically installed with secure storage as password storage. If you have existing SAP systems that were installed using software provisioning manager 1.0 SP 35 or lower, these systems still use the file `dscdb6.conf`. You can manually convert these systems to the secure storage in the file system, provided they meet the prerequisites.

Procedure

1. Log on to the database server as user `<sapsid>adm`.
2. Make sure the following environment variables are set:

   ```
   SAPSYSTEMNAME = <SAPSID>
   ```

   UNIX/Linux:

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

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

   Windows:

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

   ```
   RSEC_SSFS_KEYPATH=\\<global
   host>\sapmnt\<SAPSID>\SYS\global\security\rsecssfs\key
   ```
3. In the login environment of the `<sapsid>adm` user, change the value of the environment variable `rsdb_ssfs_connect` from 0 to 1:

   ```
   rsdb_ssfs_connect = 1
   ```

   > **i** Note
   >
   > **UNIX/Linux**: You can find the environment variable in the `.sapenv*` login scripts.
4. Create the two mandatory entries in the secure storage:

   ```
   rsecssfx put DB_CONNECT/DEFAULT_DB_USER sap<dbsid> -plain
   ```

   ```
   rsecssfx put DB_CONNECT/DEFAULT_DB_PASSWORD <password>
   ```
5. Edit the default profile and change `rsdb/ssfs_connect` from 0 to 1:

   ```
   rsdb/ssfs_connect = 1
   ```
6. Remove or rename the `dscdb6.conf` file in the global directory.
7. Log out and log in again as user `<sapsid>adm`.
8. Check the connection:

   ```
   R3trans -x
   ```

Related Information

User Authentication Concept for AS ABAP [page 30]

4.5 User Authentication Concept for AS Java

SAP's secure storage service of the AS Java is used to store the password of the database connect user `sap<sapsid>db`. This service uses the triple DES (Data Encryption Standard) algorithm together with a secret key from the key storage service to encrypt the password.

To change the password in the secure storage service, follow the procedure described in Security Aspects for Database Connections on SAP Help Portal, for example, for SAP NetWeaver 7.5.

See also SAP Note 1033993 (section *Changing Data Sources*).

Changing the password in the secure storage service does not change the password on operating system level.

> ⚠ Caution
>
> The password on operating system level and the one that is maintained in the secure storage service must be the same. Otherwise, the AS Java will not start successfully.

4.6 Other Security Features: SSL, Native Encryption, and LDAP

Secure SSL Connections

You can encrypt the communication between the application server and the database manager using the SSL protocol. For more information, see Setting up Secure SSL Connections Between SAP Application Server ABAP and an IBM Db2 Database on SAP Community or SAP Note 2385640. As of SL Toolset 1.0 SPS 26 (Software Provisioning Manager 1.0 SP 26), you can set up SSL connections for ABAP systems during SAP system installation or system copy using the installation wizard.

Db2 Native Encryption

You can use encryption to secure your database files on disk. As of Db2 10.5 FP5 and higher, you can use Db2 native encryption. For more information, see Running an SAP NetWeaver Application Server on Db2 for LUW with the IBM Db2 Encryption Technology on SAP Community.

As of SL Toolset 1.0 SPS 26 (Software Provisioning Manager 1.0 SP 26), you can set up Db2 native encryption during SAP system installation or system copy using the installation wizard. For more information about

enabling encryption as part of an SAP system installation or system copy, see the SAP blog post on SAP Community and the *Planning Your Encryption Strategy* chapters in your relevant SAP installation guide.

Transparent LDAP Authentication

Db2 transparently supports the authentication of users that are stored on an LDAP server.

5 Configuration

Every database needs to be configured for the environment where it's running. You configure IBM Db2 using the following variables/parameters:

- Environment variables at operating system level
- Db2 profile registry at operating system and Db2 instance level
- Database manager (DBM) configuration file at instance level
- Database (DB) configuration file at database level

Various parts of the SAP system itself (for example, the ABAP kernel, DBSL, AS Java, and tools like the transport control program`tp`) also have configuration parameters that are relevant for the database:

- Environment variables at operating system level
- SAP profile at SAP instance level

Read the following sections to learn more about the configuration settings at the various levels.

Related Information

Environment Variables [page 37]
Db2 Profile Registry [page 39]
Database Manager and Database Configuration [page 40]

5.1 Environment Variables

The following environment variables are set for the users `<sapsid>adm` and `db2<dbsid>` if not noted otherwise. The variables are set during the installation of your SAP system and you can change them manually.

Db2 Environment Variables

Environment Variable	Value	Location

`DB2INSTANCE`	Name of the Db2 instance (`db2<dbsid>`)	**UNIX:**
		`~/.dbenv_<hostname>.csh`
		and
		`~/.dbenv_<hostname>.sh`
		Windows:
		User environment

Environment Variable	Value	Location
DB2DBDFT	Name of the SAP database (default value: <DBSID>)	Same location as for variable DB2INSTANCE as described in this table
INSTHOME	**UNIX:** Home directory of user db2<dbsid> **Windows:** Default value for a single-partition system: <drive>:\DB2<DBSID> Default value for a multi-partition system: \\%DSCDB6HOME%\db2<dbsid>	Same location as for variable DB2INSTANCE as described in this table

SAP Environment Variables

Environment Variable	Value	Location
SAPSYSTEMNAME	<SAPSID>	**UNIX:** ~/.sapenv_<hostname> .csh and ~/.sapenv_<hostname> .sh **Windows:** User environment
dbs_db6_schema	Name of the database schema and database connect user if the schema name equals the user name	Same location as for variable SAPSYSTEMNAME as described in this table
dbs_db6_user	Name of the database connect user if database schema name does not equal database connect user name	Same location as for variable SAPSYSTEMNAME as described in this table
dbms_type	SAP short form for the database platform, for example, db6 equals DB2 for Linux, UNIX, and Windows	Same location as for variable SAPSYSTEMNAME as described in this table
DB2_CLI_DRIVER_INSTALL_PATH	If this variable is set to a directory, the default path for the CLI driver installation is overwritten.	Does not need to be set If you need to change this variable, use the same location as for the DB2INSTANCE variable.

Additional Environment Variables for Windows only

Environment Variable	Value
DSCDB6HOME	Name of the host that shares the password file `dscdb6.conf`
SAPMNT	`<drive>:\usr\sap\<SAPSID>`
SAPEXE	`<drive>:\usr\sap\<SAPSID>\SYS\exe\run`

5.2 Db2 Profile Registry

The Db2 profile registry is a central repository for Db2-specific configuration variables. It is not related to the Windows registry on Windows platforms.

You can display and set registry variables using the `db2set` command.

- To display all variables in the Db2 registry profile, enter the following command:
  `db2set -all`

  > Example
  >
  > The output looks as follows:
  >
  > ```
  > [e] DB2PATH=C:\Program Files\IBM\SQLLIB\
  > [i] DB2ACCOUNTNAME=PCIBM12\db2admin
  > [i] DB2INSTOWNER=PCIBM12
  > [i] DB2PORTRANGE=60000:60003
  > [i] DB2INSTPROF=C:\PROGRA~1\IBM\SQLLIB
  > [i] DB2COMM=TCPIP
  > [g] DB2_EXTSECURITY=YES
  > [g] DB2SYSTEM=PCIBM12
  > [g] DB2PATH=C:\Program Files\IBM\SQLLIB\
  > [g] DB2INSTDEF=DB2
  > [g] DB2ADMINSERVER=DB2DAS00
  > ```

- To set a variable, enter the following command:
  `db2set <variable>=<value>`

You can set variables in the Db2 profile registry at the following levels:

- Environment level [e]
- Db2 instance level [i]
- Global (for all Db2 instances on the same machine) [g]

> ### Example
>
> To set a variable on instance level (which is recommended), enter the following command:
>
> ```
> db2set -i <variable>=<value>
> ```

Generally, you should avoid setting Db2 registry variables unless explicitly advised by SAP support. The aggregate registry variable setting `DB2_WORKLOAD=SAP` contains all relevant settings for SAP systems.

Aggregate Registry Variable DB2_WORKLOAD

An SAP system requires that the aggregate registry variable `DB2_WORKLOAD` is set to the value *SAP*. An aggregate registry variable contains several other registry variables with specific values under one name. Exactly which registry variables are set by `DB2_WORKLOAD=SAP` depends on the Db2 Fix Pack level and the Db2 release.

`DB2_WORKLOAD=SAP` **implicitly** activates all settings that are important for an SAP system. By default, `DB2_WORKLOAD=SAP` is set at instance-level during the SAP system installation, which is the recommended level. To check that the output does not show any lines that end with `[O]`, use the **`db2set -all`** command.

5.3 Database Manager and Database Configuration

To set the parameters that are relevant for the database manager and database configuration, use the recommendations provided in the relevant SAP Note for your database version:

Database Version	SAP Note Number
Db2 11.5	2751102
Db2 11.1	2303771
Db2 10.5	1851832
Db2 10.1	1692571
Db2 V9.7	1329179

> ### ⚠ Caution
>
> Always make sure that you read the latest version of the relevant SAP Note **before** you set these parameters.

As of Enhancement Package 2 for SAP Netweaver 7.0 with SP7, you can also check and maintain the current parameter settings on the *Parameter Check* screen in the *Configuration* task area of the DBA Cockpit. For more information, see the DBA Cockpit documentation under References [page 228].

6 Db2 Memory Management

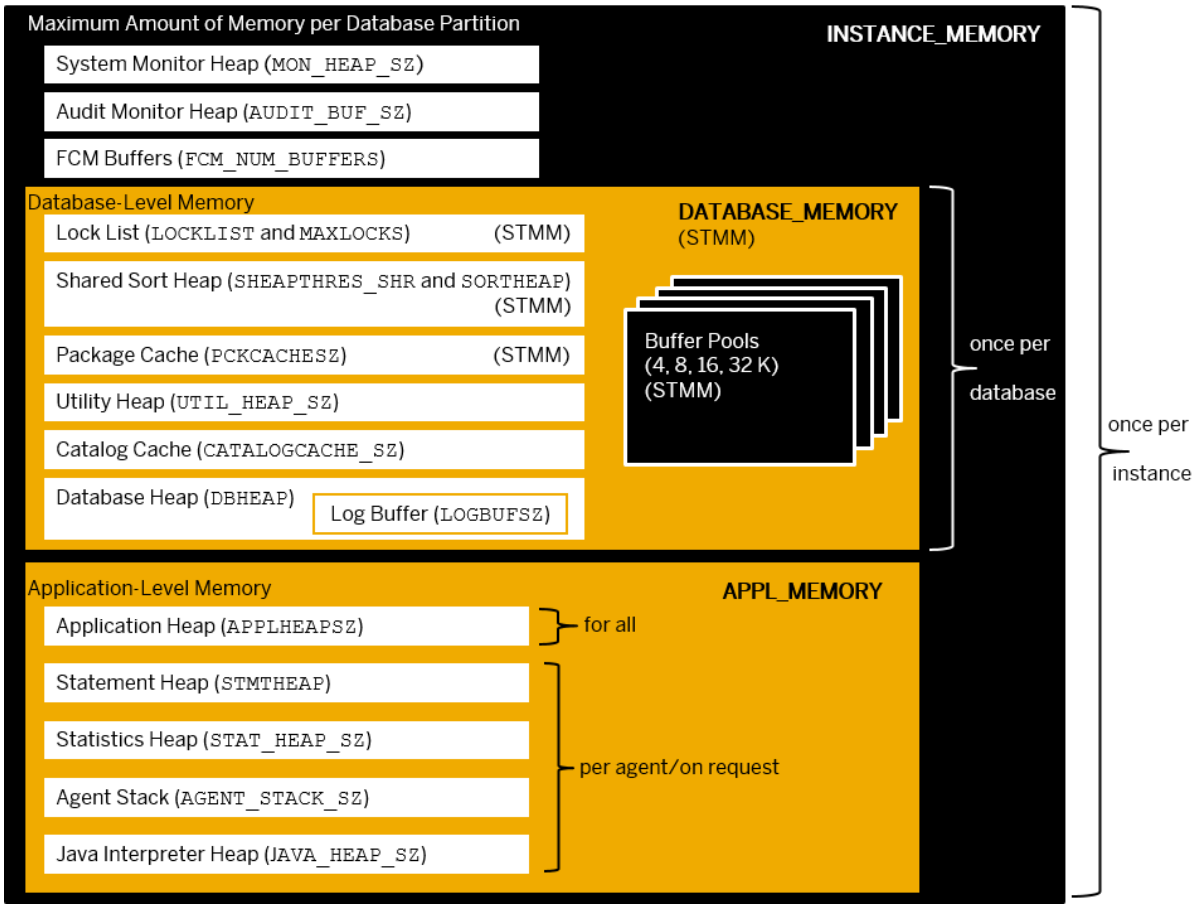
Db2 uses the following types of memory:

- Instance-level memory (memory needed for the Db2 instance)
  Instance-level memory is allocated upon start of the instance (`db2start`) and freed when the instance is stopped. Instance-level memory contains, for example, the system monitor heap and the audit monitor heap.
- Database-level memory (memory needed for every Db2 database in the instance)
  Database-level memory is used by various database-level tasks like caching database objects, execution of SQL statements, backup and restore, locking, and so on. Database-level memory is allocated as shared memory during database activation and freed after database deactivation. It contains, for example, the buffer pools, the database lock list, the shared sort heap, the package cache, and others.
- Application-level memory (memory needed on behalf of an application)

The configuration parameter `INSTANCE_MEMORY` specifies the maximum amount of memory that Db2 can allocate in total, including database-level memory and application-level memory.

The application heap is shared by all Db2 agents. The parameter `APPLHEAPSZ` should be set to *AUTOMATIC* to allow the application heap to grow until the `APPL_MEMORY` limit is reached.

The following figure shows the Db2 memory model:



Db2 Memory Model

Tools for Monitoring Memory Consumption

To monitor the memory consumption of your database, you can use one of the following tools:

- **`db2mtrk`** command on the command line
- Db2 Memory Visualizer (graphical tool)
- DBA Cockpit (SAP tool)

Related Information

Important Database Memory Areas [page 43]
Self-Tuning Memory Management (STMM) [page 44]

6.1 Important Database Memory Areas

The following shared memory areas are of particular interest with regard to database performance and require proper tuning:

Shared Memory Area	Description
Instance memory	The configuration parameter `INSTANCE_MEMORY` determines the total amount of memory that Db2 can allocate. You should set `INSTANCE_MEMORY` to a fixed value. For dedicated database servers (that is, database servers where no other workload runs) up to and including 64 GB of RAM, set `INSTANCE_MEMORY` to 75% of the available memory. For systems with more than 64 GB of RAM, set `INSTANCE_MEMORY` to 85% of the available memory.
Database buffer pools	Usually, this is the largest component within the shared memory area. Here, all regular and index data is manipulated. Database buffer pools also serve as a cache for the data read from disk. If they are too small, the buffer quality - or hit ratio - decreases and more data must be read from disk. As a result, database performance decreases, too. All tablespaces (including the tablespace for the system catalog) use a page size of 16 KB. During the installation of SAP systems only one buffer pool (`IBMDEFAULTBP`) is created and is used by all tablespaces.
Database lock list	This is the area where Db2 stores its locks (for more information, see Locking Concepts [page 129]). If there is not enough space to hold all the locks, a lock escalation occurs. For example, instead of several single rows a complete table receives a lock. Lock escalations lead to a lower level of concurrency and a higher risk of deadlocks. **i Note** With Db2 11.1 MP2 FP2, the Db2 registry variable `DB2_AVOID_LOCK_ESCALATION` is introduced. If it is set to ON, which is the default for SAP systems on Db2, lock escalation is not performed. Instead, the SQL error *SQL0912N* is returned to the application that requested the lock that would normally result in lock escalation. For more information, see Locking Concepts [page 129].
Database shared sort heap	This area is used by Db2, for example, to process hash or merge joins, to deal with in-memory tables, and to do sorts. If the sort operations cannot be performed in the memory area, a sort overflow occurs and a temporary table on the disk is used. This process is more time consuming than an in-memory operation.
Package cache	This area stores access plans, which have already been compiled and optimized, for dynamic SQL statements. The package cache is also located in the shared memory so that the plans can be reused between users or applications. If the package cache is too small, access plans are removed from the cache and the corresponding statements - if they are executed again - need to be recompiled.

The Db2 shared memory areas are all in an area that is configured by the Db2 database configuration parameter `DATABASE_MEMORY`.

You can set the `DATABASE_MEMORY` parameter to one of the following values:

- *AUTOMATIC*
  The size of the database memory is adapted to the workload demands of Db2. For this purpose, memory is taken from and returned to the operating system by the database.
  You do not need to configure beforehand how much memory has to be allocated to the database.
- *Fixed value*
  The maximum size of the database memory is set to a fixed numeric value.
- *COMPUTED*
  The size of the database memory is computed based on the sizes of the consumers.

Tuning the Database Shared Memory Areas

The database shared memory areas can be tuned as follows:

- **Manually** by the database administrator
- **Automatically** by using Db2's Self Tuning Memory Management. For more information, see Self-Tuning Memory Management (STMM) [page 44].

6.2 Self-Tuning Memory Management (STMM)

What Is STMM?

Db2's Self-Tuning Memory Management (STMM) lets the Db2 instance automatically set and adjust the database shared memory to a value that improves the overall database performance. The database shared memory consists, for example, of buffer pools, sort heap, lock list, package cache, and catalog cache. STMM adapts quickly to workload shifts that require memory redistribution and it can tune multiple databases and instances on the same machine at the same time.

The overall memory consumption of Db2 is controlled by the database manager configuration parameter `INSTANCE_MEMORY`. If you have a standalone database server (that is, a database server where no other applications run), set `INSTANCE_MEMORY` to 75% of the system memory for systems up to and including 64 GB RAM. For systems with more than 64 GB of RAM, set `INSTANCE_MEMORY` to 85% of the system memory.

Db2 monitors the use of each memory heap and analyzes if the system benefits from a larger heap in one area and then allocates more memory accordingly. STMM automatically balances memory for optimal usage between all consumers that are set to `AUTOMATIC`. Consumers that are set to a fixed value do not participate in STMM tuning and keep their configured size.

Which Memory Consumers Can You Enable for Self-Tuning?

Memory Consumer	Controlled By
Buffer pools	Size parameter of the `ALTER BUFFERPOOL` and `CREATE BUFFERPOOL` statements
Package cache	`pckcachesz` configuration parameter
Lock list	`locklist` and `maxlocks` configuration parameters
Sort heap	`sheapthres_shr` and `sortheap` configuration parameters
Database shared memory	`database_memory` configuration parameter

The tuning behavior of STMM is determined by the database configuration parameter `DATABASE_MEMORY` that can be set to the following values:

- *Fixed value*
  STMM tunes all consumers that are set to *AUTOMATIC* inside the memory area that is determined by the `DATABASE_MEMORY` parameter. The memory area itself does not grow beyond the size that has been configured by the fixed value.
- *AUTOMATIC*
  STMM tunes all consumers that are set to *AUTOMATIC*. If required, memory is taken from and returned to the operating system by the database. As a consequence, the overall shared memory size can grow and shrink according to the memory available on your hardware and the workload of Db2.

  > ⚠ Caution
  >
  > Db2 always returns memory to the operating system if other consumers on your hardware demand more memory. Therefore, in an environment where other consumers continuously demand more memory, Db2 might reduce its memory consumption to the absolute minimum. This can result in poor database performance.

- *COMPUTED*
  The database manager calculates a fixed value for `DATABASE_MEMORY` and allocates the memory at database activation time. STMM does not tune this value. All consumers inside `DATABASE_MEMORY` that are set to *AUTOMATIC* are tuned by STMM.

What You Need to Consider When Using STMM

Tuning the database memory can be a difficult task since the workload on the database can sometimes be unpredictable. A memory configuration with fixed values cannot provide optimal database performance because the values are not flexible and have to be modified manually.

Tuning the database memory requires expert knowledge. To optimize performance, you might have to test various tuning options, which can take weeks, until you achieve the best performance. This process is time-consuming and expensive.

A database with STMM enabled automatically adjusts its memory and achieves the maximum performance without any manual effort.

By default, current versions of the SAP installation tool enable STMM for all memory consumers mentioned above for all SAP systems that are installed with IBM Db2. We recommend that you use STMM.

Up to and including Db2 10.1, in a DPF (database partitioning feature) or pureScale environment where the workload is distributed across several database servers, STMM tuning takes place only on one database server and changes are propagated to all other servers. For recommendations on the usage of STMM in such environments, see SAP Notes 1132282 and 1555903 (section *Self-Tuning Memory Manager*).

As of Db2 10.5, STMM tuning works distributed on Db2 pureScale database servers. This means that STMM determines the optimal memory tuning and adjusts memory distribution for each member individually. In a distributed database server (DPF), STMM can optionally also be configured to use member-individual tuning. This capability is available on request only through an SAP pilot program. For more information, see section *STMM Member-Individual Tuning* in SAP Note 1851853.

6.2.1 Enabling Self-Tuning Memory Management (STMM)

To enable or disable STMM, you use the `SELF_TUNING_MEM` parameter. If this parameter is set to *ON*, the memory tuner dynamically distributes available memory resources as required between memory consumers that are enabled for self-tuning. Since memory is being traded between memory consumers, there must be at least two memory consumers enabled for self-tuning.

To view the current setting of the `self_tuning_mem` parameter, use the **`GET DATABASE CONFIGURATION`** command that specifies the `SHOW DETAIL` parameter. The possible parameter settings are:

Parameter	Setting
`SELF_TUNING_MEM`	*OFF*
`SELF_TUNING_MEM`	*ON (Active)*
`SELF_TUNING_MEM`	*ON (Inactive)*

If the parameter is set to *ON (Active)*, the memory tuner is actively tuning the memory on the database system. If the parameter is set to *ON (Inactive)*, this means that although the parameter is set to *ON*, self-tuning does not occur because there are less than two memory consumers enabled for self-tuning. For information about the use of STMM in multi-partition database environments, see SAP Note 1132282.

Alternatively, you can change the buffer pool, database, and database manager configuration using the functions available in the *Configuration* task area of the DBA Cockpit.

Procedure

1. Log on to the database server as `db2<dbsid>`.
2. To enable all of your buffer pools for STMM tuning, enter the following SQL statement:
   ```
   db2 alter bufferpool <bp_name> size AUTOMATIC
   ```
3. To update your database configuration for all remaining consumers, enter the following commands:
   ```
   db2 update dbm cfg using SHEAPTHRES 0
   db2 update db cfg for <DBSID> using LOCKLIST AUTOMATIC MAXLOCKS AUTOMATIC
   PCKCACHESZ AUTOMATIC SORTHEAP AUTOMATIC SHEAPTHRES_SHR AUTOMATIC
   ```
4. To set `DATABASE_MEMORY` to a fixed value (at least 100000 pages), enter the following command:
   ```
   db2 update db cfg for <DBSID> using DATABASE_MEMORY <value>
   ```
   Depending on your Db2 version and your operating system you can set this value also to *AUTOMATIC*.
5. To enable STMM, enter the following command:
   ```
   db2 update db cfg for <DBSID> using SELF_TUNING_MEM ON
   ```

To disable STMM, enter the following command:

```
db2 update db cfg for <DBSID> using SELF_TUNING_MEM OFF
```

After you have disabled STMM, your database continues to use the settings that were active at the time when STMM tuning was switched off. You can either continue to use these settings or update your configuration to fixed values.

More Information

SAP Note 2303771: Db2 11.1 Standard Parameter Settings

SAP Note 1851832: Db2 10.5 Standard Parameter Settings

SAP Note 1692571: Db2 10.1 Standard Parameter Settings

SAP Note 1329179: Db2 9.7 Standard Parameter Settings

Self-tuning memory overview in the IBM Db2 documentation

7 Storage Management

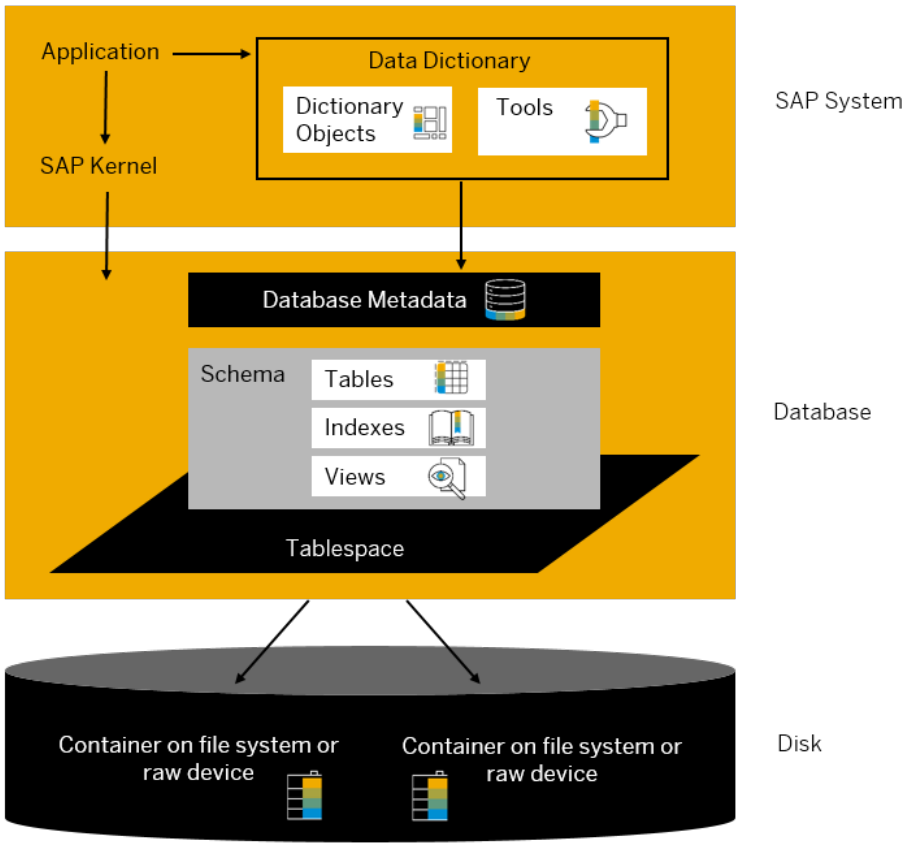
Optimal storage management is the basis of a well-performing database server and allows you to keep your storage costs low. Furthermore, with the appropriate strategy, you are able to run your Db2 database with a low maintenance effort.

The following sections provide information about the database objects in an SAP environment that are used to store the application data, how these objects are persisted in the database, and how you can monitor them.

We also explain which Db2-specific features you can use to optimize your storage management.

7.1 Concepts of the SAP Dictionaries

The following figure shows the relation between SAP application objects and database objects:



To manage database objects in an SAP system, you have to understand the following considerations: The application data of an SAP system is stored in database tables. Other database objects that are used by the SAP system are views and indexes. Db2 maintains a catalog with all existing database objects, the system catalog.

A similar catalog is maintained in the SAP system. This can either be the ABAP Dictionary or the Java Dictionary – or both, in case of an ABAP+Java SAP system.

- ABAP Dictionary
  The ABAP Dictionary describes the logical structure of the application development objects such as tables, views, and data types, as well as their representation in the structures of the underlying relational database. These are just two of the functions that the ABAP Dictionary provides.
  You can access the ABAP Dictionary in the SAP system using transaction SE11. The ABAP Dictionary uses the database utility as an interface to the underlying database. The database utility enables you to physically create ABAP-defined objects such as tables and database views in the database. You can access the database utility using transaction SE14.
  You can also check the overall the consistency of the ABAP dictionary on the *Missing Tables and Indexes* screen in the *Diagnostics* task area of the DBA Cockpit.
- Java Dictionary
  The Java Dictionary is a repository for logical definitions of database objects. It is used for defining database objects (tables and indexes) in the Open SQL for Java framework. You can access the Java Dictionary using the Java Dictionary perspective that is integrated in the SAP NetWeaver Developer Studio.

More Information

See SAP Help Portal, for example, for SAP NetWeaver 7.4:

ABAP Dictionary

Providing Java Dictionary Tables and Data Types

7.2 Schemas and the SAP System Database

7.2.1 Example: CREATE DATABASE Statement

Each SAP system uses one Db2 database to store its data. Database schemas are used to group the database objects of the AS Java and AS ABAP. The relation between the database of the SAP system and the database schema is as follows:

- SAP ABAP-only system
  One database with one schema named by default `sap<sapsid>`
- SAP Java-only system
  One database with one schema named by default `sap<sapsid>db`
- SAP ABAP+Java system
  One database with two schemas, that is, `sap<sapsid>` for the ABAP objects and `sap<sapsid>db` for the Java objects

It's also possible to store multiple SAP systems in the same Db2 database. This setup is known as “multiple components in one database” (MCOD). For more information about how to install an MCOD system, see your relevant installation guide that you can find on SAP Help Portal.

The SAP installation tool creates the database during the installation phase. The exact settings for the database depend on the user input during the dialog phase.

The following figure shows an example of a typical **CREATE DATABASE** statement that is based on the default values provided by the current SAP installation tool:

```
create database <DBSID>
(1)  automatic storage yes
(2)  on /db2/<DBSID>/sapdata1,
       /db2/<DBSID>/sapdata2,
       /db2/<DBSID>/sapdata3,
       /db2/<DBSID>/sapdata4
(3)  dbpath on /db2/<DBSID>
(4)  using codeset <UTF-8|ISO8859-1>
(5)  territory en_US
(6)  collate using <IDENTITY_16BIT|IDENTITY>
(7)  pagesize 16 k
(8)  dft_extent_sz 2
(9)  catalog tablespace managed by automatic storage
(10) with 'SAP database <DBSID>';
```

Example of a Typical CREATE DATABASE Statement

The database is created in the path `/db2/<DBSID>` (3) with the automatic storage management option enabled (1) and four automatic storage paths provided (2). As code set (4), `UTF-8` is used for Unicode systems and `ISO8859-1` is used for non-Unicode systems. The territory element (5) determines how Db2 treats locale-sensitive data, for example, time and money. This is also reflected in the database configuration parameter of the same name.

The collating sequence (6) defines the sort order for the characters of the code set (`IDENTITY_16BIT` for Unicode systems, `IDENTITY` for non-Unicode systems).

> **i** Note
>
> If an incorrect collating sequence is specified during the database creation, the SAP system will not function properly.

The default page size (7) for all tablespaces and buffer pools is set to 16 KB and the default extent size (8) to 32 KB (2 pages). The catalog tablespace (SYSCATSPACE) is managed by automatic storage (9) and a comment (10) for the description of the database is added in the database catalog.

7.2.2 Determining the Size of the Database

Use

To determine the allocated size of all tablespaces and the remaining database capacity, that is, the amount of free space still available, you can either use the DBA Cockpit or the Db2 stored procedure `GET_DBSIZE_INFO`.

Procedure

Using the DBA Cockpit

You can view the database and tablespace sizes in the *Space* task area of the DBA Cockpit.

For more information, see the DBA Cockpit documentation [page 228].

Db2 Stored Procedure GET_DBSIZE_INFO

1. Log on to the database server as user `db2<dbsid>`.
2. Enter the following command:
   ```
   db2 "CALL GET_DBSIZE_INFO(?, ?, ?, -1)"
   ```
   The result is displayed in bytes.

For more information, see GET_DBSIZE_INFO procedure in the IBM documentation.

7.3 Tables and Indexes

In an SAP system, tables and indexes are stored in different tablespaces. The data class, which is an attribute of every table in the *Technical Settings* area of the ABAP Dictionary, defines in which tablespaces the table and their indexes are created.

If you want to see the mapping between data classes and tablespaces, call transaction `DBACOCKPIT`. On the *Database* tab page of the DBA Cockpit, choose *Configuration* > *Data Classes*.

Table Features Supported by the ABAP Dictionary

All tables are created with tablespace options, for example, `CREATE TABLE ...IN <tablespace1> INDEX IN <tablespace2> LONG IN <tablespace3>`. The ABAP Dictionary supports the following storage options for tables:

- Tablespaces for the data, index, and long part of the table
- Preferred lock granularity (`LOCKSIZE`) for the table (the default value is `ROW`)
- Compression mode: no compression, static (classic) compression, and adaptive compression

> **i Note**
>
> Value compression is not relevant for the ABAP Dictionary because the database shared library (DBSL) automatically switches on value compression if this leads to storage savings.

- `ORGANIZE BY DIMENSIONS (...)` option to support multidimensional clustering (MDC) tables
- `ORGANIZE BY INSERT TIME (...)` option to support insert time clustering (ITC) tables (see SAP Note 1701181)
- `PARTITIONING KEY (...)` option to support tables on multiple partitions (database partitioning feature (DPF)). With the corrections of SAP Note 1701181, `DISTRIBUTE BY HASH ...` is also supported.
- `PARTITION BY (...)` option to support range-partitioned tables (see SAP Note 1379362)
- `ORGANIZE BY COLUMN (...)` option (see SAP Note 1819734)
- As of Db2 10.1, hidden columns are also supported (see SAP Note 1701181).

Index Features

Indexes are used by the database system to provide fast access to table data and to ensure unique constraints. In an SAP system, the following indexes are used in the database:

- Primary indexes to ensure a primary key constraint
- Unique indexes to ensure a unique constraint
- Secondary indexes

> **i Note**
>
> Foreign keys are not used in an SAP system database. You can define foreign keys in the ABAP Dictionary. The SAP system then ensures referential integrity.

All indexes (AS ABAP and AS Java) are created with the `ALLOW REVERSE SCAN` option so that they support both forward and backward scans.

The following Db2-specific index features are also supported by the ABAP Dictionary:

- Clustered index
  A clustered index groups data in the referenced table according to the index order.
- Index with include columns
  Include columns are additional columns in a compound unique or primary index that are used to enforce uniqueness. Their only purpose is to provide faster access paths to table data.
- Index compression
  The index compression attribute is preserved during a conversion. By default, indexes are created with compression attribute if the underlying table is compressed.
  For more information, see **SAP Note** 1379984.
- Partitioned and non-partitioned indexes
  Indexes of partitioned tables can be partitioned or non-partitioned.

You can enable these options either on the database level or using the database utility (transaction SE14) in the ABAP Dictionary.

More Information

The Database Utility on SAP Help Portal (for example, for SAP NetWeaver 7.4)

CREATE INDEX statement in the IBM documentation

7.3.1 Checking the Size of Tables and Indexes

You can check the size of tables and indexes using one of the following options:

- DBA Cockpit
  In your SAP system, call transaction DBACOCKPIT and choose *Space* > *Tables and Indexes*.
  For more information, see the DBA Cockpit documentation [page 228].
- Table function `ADMIN_GET_TAB_INFO`
  For more information, see ADMINTABINFO administrative view and ADMIN_GET_TAB_INFO table function - retrieve table size and state information in the IBM documentation.

7.4 Tablespaces, Containers, and the File System

Db2 uses containers to physically store data. A container can be a file, a directory, or a raw device. Since containers are objects that actually exist on a disk or a raw device, they are considered as physical database objects.

In contrast to containers, tablespaces are logical database objects. They are primarily used to map tables and indexes (which are also logical database objects) to containers and buffer pools (physical database objects).

By default, an SAP system has several database partition groups. You use them to assign tablespaces and buffer pools to database partitions if you are using the database partitioning feature (DPF). All SAP basis tables must be assigned to `SAPNODEGRP_<SAPSID>`. `SAPNODEGRP_<SAPSID>` must be on partition 0 only and not spread across other partitions. If you create, for example, a new tablespace in the DBA Cockpit for non-BW tables, you must use `SAPNODEGRP_<SAPSID>`.

For SAP BW tables, you can use different database partition groups and they may be distributed across several partitions. For more information about BW tables, see the guide *SAP Business Warehouse on IBM Db2 for Linux ,UNIX, and Windows: Administration Tasks*, which you can find in the appendix under References [page 228].

Db2 knows various tablespace types. Based on how a tablespace is managed, Db2 distinguishes between the following types:

Tablespace Type	Description
SMS	Managed by the operating system Usually, they are mapped to a directory that contains files for every database object that is created within the tablespace. For SAP systems, SMS tablespaces are only permitted for temporary tablespaces. SMS tablespaces must not be used for table or index data.
DMS	Managed by the database management system There are three types of DMS tablespaces: • DMS file tablespaces are mapped to OS files. • DMS raw tablespaces are mapped to raw devices. • DMS automatic resize tablespaces are mapped on files and can grow automatically. As of Db2 10.1 Fix Pack 1, the DMS tablespace type is deprecated. Use Db2 automatic storage tablespaces instead.
Managed by Db2's automatic storage management	Db2's automatic storage management must be enabled when the database is created. You can convert existing nonautomatic storage databases to use Db2's automatic storage management. When the database is created, storage paths are specified. Automatic storage tablespaces get storage from these storage paths. You do not have to specify containers during tablespace creation. Automatic storage tablespaces still rely on the infrastructure that is provided by SMS and DMS tablespaces. If they are temporary, they are mapped to SMS tablespace. Otherwise, they are mapped to DMS tablespaces.

Automatic storage is the default storage setup for SAP systems. While you can still use DMS tablespaces in existing systems, we strongly recommend that you use automatic storage for new deployments and also consider the migration of existing systems to automatic storage as soon as possible. For more information about automatic storage, see 1895425.

As of Db2 Version 10.1 Fix Pack 1, the DMS tablespace type has been deprecated.

Based on which type of data a tablespace contains, Db2 distinguishes between the following tablespace types:

- Regular
  Contains any kind of data except temporary data. In an SAP database, only SYSCATSPACE should be a regular tablespace.
- Large
  Contains any kind of data except temporary data, but allows for larger row identifiers (RIDs). All SAP data and index tablespaces should be large tablespaces.
- Temporary
  Contains only temporary data

Multi-Temperature Data Management and Storage Groups

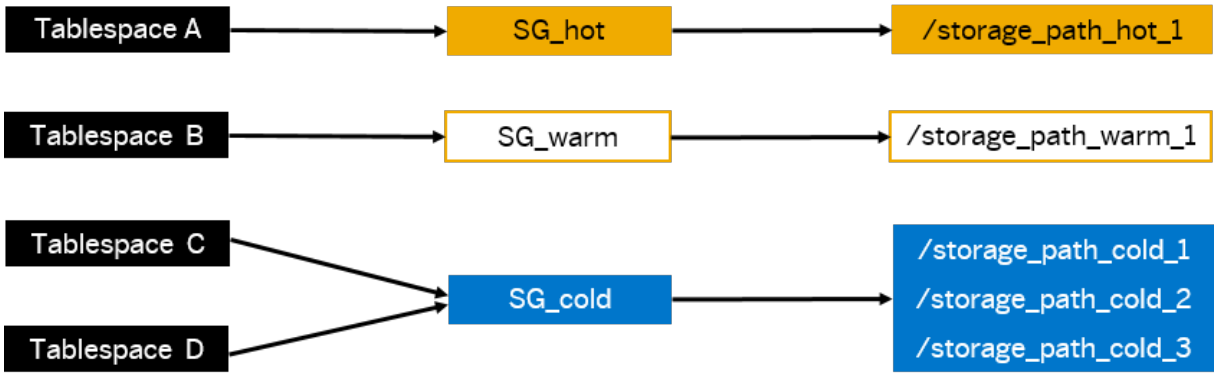
If you are using Db2 10.1 or higher and your database is enabled for automatic storage, you can benefit from multi-temperature data management. This means that you distribute your data in different tablespaces according to its temperature, that is, based on the frequency of usage of the data, for example, as follows:

- Hot data (accessed very often)
- Warm data (accessed frequently)
- Cold data (accessed infrequently)

Instead of this three-tier approach, you can also use a two-tier distinction and divide your data into two different temperatures only, that is, hot and cold data.

With the separation of data according to its temperature, you can subsequently distribute the data on different storage media using storage groups. Storage groups contain multiple storage paths that you can define and to which you can assign tablespaces. Using storage groups, you can improve system performance and reduce costs because you can place frequently accessed data on fast and expensive storage devices and put data that is not so often accessed on slower and cheaper storage devices. Placing less frequently changed data on separate media also allows you to take backups of this data less frequently, which reduces the overall backup volume and costs.

You can think of storage groups as an additional layer between the tablespaces and the automatic storage paths in your database. They allow you to place each tablespace on a defined storage type as shown in the following figure:



Storage Groups

The use of storage groups is also supported by the DB6 Partitioning Administrator. For more information about the DB6 Partitioning Administrator, see SAP Note 1686102.

To use storage groups, you must perform the following steps:

1. Make your different storage systems available in the file system of your database server.
2. Define a storage group for each storage type.
3. Divide your data in different tablespaces according to its temperature.
4. Assign each tablespace to a storage group based on the temperature of the data it contains.

Tablespaces with Reclaimable Storage

With Db2 V9.7, the tablespace attribute *reclaimable storage* was introduced for DMS tablespaces. Tablespaces with this attribute allow for an easier reduction of the high-water mark (HWM) and thereby a better reuse of free space. As of Db2 V9.7, all DMS tablespaces that are created use reclaimable storage by default.

> ### i Note
>
> You **cannot** convert tablespaces that were created with a Db2 version lower than V9.7 to reclaimable storage tablespaces. Instead, you can use the `DB6CONV` tool to move all tables to a new tablespace with reclaimable storage. For more information about `DB6CONV`, see SAP Note 1513862.

Reclaiming Space Using the DBA Cockpit

For a single tablespace with reclaimable storage, you can reclaim space in the DBA Cockpit using the *Reduce HWM* button under *Space* > *Tablespaces*. If you want to reclaim space from all tablespaces, you can use the DBA Planning Calendar and schedule the job *Reduce High Water Mark for Tablespaces*.

Tablespaces in an SAP System

The following applies for tablespaces in an SAP system:

- Table data and indexes reside in separate tablespaces.
- Tablespaces are named using the following naming convention: `<SAPSID>#<TBSPACE_NAME><Extension>` where `<Extension>` can either be *D* (for table data) or *I* (for indexes).

  > ### Example
  >
  > `DEF#SOURCEI` is the name of a tablespace in SAP system DEF. This tablespace contains index data. The related table data is located in tablespace `DEF#SOURCED`.

- Temporary tablespaces are created as automatic storage tablespaces if automatic storage is enabled in the database. Otherwise, temporary tablespaces are created as SMS tablespaces.
- As of SAP NetWeaver 7.0 SP13, the database is always created as automatic storage-enabled database. During the installation, you can still specify if user tablespaces are created as DMS or as automatic storage tablespaces, but the `SYSCATSPACE` tablespace is always created as automatic storage tablespace.
- As of SAP NetWeaver 7.0 SP13, all tablespaces (including `SYSCATSPACE`) are created with a uniform page size of 16 KB, an extent size of two pages and with the prefetch size set to *automatic*.
- Prior to Db2 V9.7, you could enable a Db2 database for automatic storage only during its creation. As of Db2 V9.7, existing tablespaces can be converted from DMS to automatic storage.
- For existing SAP systems, we recommend that you enable the automatic resize function for all DMS tablespaces. To do so, run the script `db6_update_db` that is part of every Fix Pack shipment.
- We do **not** recommend the use of raw devices. Since the general trend is more oriented toward file systems, the support of raw devices is likely to decrease in the future.
- You can use tablespace pools, which allows for an even distribution of tables and better balanced tablespace sizes in your SAP system. For more information about tablespace pools, see SAP Note 2267446.

7.4.1 Determining the Tablespace Type

You can determine the type of a tablespace using the DBA Cockpit or the Db2 command line.

Using the DBA Cockpit

In your SAP system, call transaction `DBACOCKPIT` and choose *Space* > *Tablespaces*.

You need to use SAP NetWeaver 7.0 SP13 or higher to be able to differentiate between automatic storage tablespaces and DMS tablespaces in the DBA Cockpit.

Using the Command Line

On the command line, enter the following command as user `db2<dbsid>`:

```
db2 get snapshot for tablespaces on <DBSID>
```

In the output, search for the relevant lines that look as follows for each tablespace:

```
Tablespace name                 = DEF#CLUD

Tablespace ID                   = 10
Tablespace Type                 = Database managed space

Tablespace Content Type         = All permanent data. Large table space.

…

Using automatic storage         = No

Auto-resize enabled             = Yes

…
```

Tablespace Types

In the example above, the tablespace `DEF#CLUD` is a DMS tablespace that is enabled for automatic resize and contains large data.

To summarize information about the automatic resize and automatic storage status of all tablespaces, you can also use the following SQL statement:

```
db2 "SELECT SUBSTR(TBSP_NAME,1,10) AS TBSP_NAME, TBSP_TYPE, TBSP_CONTENT_TYPE,
TBSP_USING_AUTO_STORAGE, TBSP_AUTO_RESIZE_ENABLED FROM TABLE(SNAP_GET_TBSP_V91(''))
AS T"
```

A value of `1` in the columns `TBSP_USING_AUTO_STORAGE` and `TBSP_AUTO_RESIZE_ENABLED` indicates that the respective feature is enabled, `0` would indicate that it is not enabled.

To check if the database is enabled for automatic storage, use the following command:

```
db2 “get snapshot for database on <SAPSID>”
```

> **i Note**
>
> If automatic storage is enabled for the database, the value reported after *Number of automatic storage paths* is higher than `0` and all defined automatic storage paths are listed afterwards.

As of DB2 V9.7, tablespaces can have the reclaimable storage attribute. To check if tablespaces have the reclaimable storage attribute, use the following SQL statement:

```
db2 "SELECT varchar(tbsp_name, 30) as tbsp_name, tbsp_type,
reclaimable_space_enabled FROM TABLE(MON_GET_TABLESPACE('',-2)) AS t"
```

A value of `1` in the `reclaimable_space_enabled` column indicates that this tablespace has the reclaimable storage attribute.

More Information

SNAPTBSP administrative view and SNAP_GET_TBSP table function - Retrieve table space logical data group snapshot information in the IBM documentation

7.4.2 Checking the Size of a Tablespace

You can check the size of a tablespace using one of the following tools:

- DBA Cockpit
  In your SAP system, call transaction DBACOCKPIT and choose *Space* > *Tablespaces*.
  For more information, see the DBA Cockpit documentation [page 228].
- Db2 CLP
  To display a list of all tablespaces, enter the following command:
  ```
  list tablespaces show detail
  ```
  The lines *Total pages*, *Used pages*, and *Free pages* under each tablespace indicate the size in pages.

7.4.3 Maintaining the Size of a Tablespace Manually

You can maintain the tablespace size using either the DBA Cockpit or Db2 CLP.

> ⚠ Caution
>
> The following information does **not** apply if you are using **automatic storage tablespaces** or tablespaces enabled for Db2's automatic resize function.

Increasing the Size of a Tablespace

The capacity for a DMS tablespace corresponds to the total size of all containers that are allocated to this tablespace. If a DMS tablespace reaches the threshold capacity (depending on the usage of the tablespace - 90% is a possible threshold), you should increase its size using one of the following tools:

- DBA Cockpit

  In your SAP system, call transaction `DBACOCKPIT` and choose *Space* > *Tablespaces*. On the *Tablespaces* screen, choose the *Change* pushbutton and specify the required values on the *Change Tablespaces* dialog box.
  For more information, see the DBA Cockpit documentation under References [page 228].
- Db2 CLP
  Using SQL statements, you can perform the following actions:
  - Add a new container to a DMS tablespace as follows:
    ```
    alter tablespace <tablespace name> add ( file '<container name>' <number of pages> )
    ```
    The Db2 database manager automatically rebalances the tables in the DMS tablespace across all available containers. During rebalancing, data in the tablespace remains accessible.
  - Increase the size of one or more containers in the DMS tablespace by entering the following SQL statement:
    - Extend one or more existing tablespace containers **by a specific size** by entering the following SQL statement:
      ```
      alter tablespace <tablespace name> extend (all containers <number additional pages> )
      ```
    - Resize one or more existing tablespace containers **to a specific size** by entering the following SQL statement:
      ```
      alter tablespace <tablespace name> resize (all containers <new container size in pages> )
      ```

    Under certain circumstances the Db2 database manager automatically rebalances the extents in the DMS tablespace across all available containers. During rebalancing, data in the tablespace remains accessible.
  - Add a new stripe set to increase the size of a DMS tablespace by entering the following SQL statement:
    ```
    alter tablespace <tablespace name> begin new stripe set (file '<container name>' <number of pages>)
    ```
    Adding a new stripe set does not result in rebalancing. The additional space is immediately available.

    > → Recommendation
    >
    > We recommend that you avoid rebalancing because it has a significant impact on the database performance. In addition, the added space is not available until rebalancing has finished.
    >
    > Always resize all containers of the last stripe set, because no rebalancing occurs there. If this is not possible any more, for example, because the file system is full, begin a new stripe set with an appropriate number of containers.
    >
    > If you are using automatic resize (DMS tablespaces) or automatic storage, database containers do **not** have to be resized because this is automatically done by Db2.

Reducing the Size of a Tablespace

Reduce the size of a tablespace using one of the following tools:

- DBA Cockpit
  In your SAP system, call transaction `DBACOCKPIT` and choose *Space* > *Tablespaces*. On the *Tablespaces* screen, choose the *Change* or *Reduce* pushbutton.
  For more information, see the DBA Cockpit documentation under References [page 228].
- Db2 CLP
  Enter the following SQL statement:
  ```
  alter tablespace <tablespace name> resize (all containers <new container size in pages>)
  ```

> ⚠ Caution
>
> You can explicitly lower the high-water mark as follows:
>
> - For **non-automatic** storage tablespaces created with Db2 V9.7 or higher, you use the **`ALTER TABLESPACE ... LOWER HIGH WATER MARK`** command and then the **`ALTER TABLESPACE ... REDUCE`** command.
> - For **automatic** storage tablespaces created with Db2 V9.7 or higher, you can use the **`ALTER TABLESPACE ... REDUCE`** command.

7.4.4 Checking the Available Space in a File System

You can check the available space in one of the following ways:

- Using the DBA Cockpit
  In your SAP system, call transaction `DBACOCKPIT` and choose *Space* > *File Systems*. The information displayed on this screen helps you to determine how much free space is available in your file systems.
  For more information, see the DBA Cockpit documentation [page 228].
- Operating system means
  Use the command **`df`** on UNIX systems or the Windows explorer on Windows systems.

7.5 Compression

7.5.1 Data Compression

Disk storage systems can often be the most expensive components of a database solution. For large warehouses or databases with huge data volumes, the cost of the storage subsystem can easily exceed the combined cost of the hardware server and the data server software. Therefore, even a small reduction in the storage subsystem can result in substantial cost savings for the entire database solution.

Data compression can reduce storage requirements, improve I/O efficiency, and provide quicker data access from the disk.

IBM Db2 offers the following compression modes:

- **Value compression**, which introduced a new row format that saves space for certain data types.
- **Classic or static row compression**, which uses standard data compression algorithms to compress data in a row. It replaces recurring patterns in table rows with shorter symbol strings. The patterns and symbols are stored in the table-level compression dictionary.
- **Adaptive compression** as of DB2 10.1: This compression technique comprises the classic row compression and a new compression algorithm that works on page level.
- **Index compression**: To compress indexes, the Db2 database manager uses various index compression techniques, for example, variable slot directory, RID list compression, and prefix compression.

You can determine the default compression option for newly created tables using the Db2 global variable `SAP<SID>.GLOBAL_COMPRESSION_OPTION`.

All compression modes and the global compression option are described in the following sections in more detail.

7.5.1.1 Value Compression

Db2 can store NULL, 0 length values, and system default values efficiently using value compression. By specifying the `VALUE COMPRESSION` clause when creating a table, a new data row format is used to store NULL and 0 length values. These values, which have been assigned to specific variable-length data types (for example, VARCHAR, VARGRAPHIC, LONG VARCHAR, LONG VARGRAPHIC, BLOB, CLOB, and DBCLOB), are then not stored on disk. Only overhead values associated with these data types consume disk space.

As of SAP release 7.00, the ABAP kernel creates tables automatically with value compression if this saves space. For more information, see SAP Note 886231. Also, all tables created by the Java dictionary in the Java schema use value compression.

7.5.1.2 Classic or Static Row Compression

You can use classic or static row compression (also known as deep compression) to compress data rows in tables. Only data rows are compressed, **not** indexes, LONG, or LOB data.Classic row compression is software-based and therefore requires additional CPU cycles.

In general, the I/O data transfer is reduced due to the smaller record length of compressed data records. Row compression replaces recurring patterns in table rows with shorter symbol strings. The patterns and symbols are stored in the compression dictionary. The dictionary is static — that is, after it has been created, you can no longer change it. If you want to create a new compression dictionary, you have to either move the table using the `DB6CONV` tool (see SAP Note 1513862) or issue an offline `REORG` operation against the table. Compression dictionaries are stored in the table object.

In multi-partition databases, tables have a compression dictionary on each database partition where the table is located. The maximum size of a compression dictionary is 150 KB, whereas the average size is 75 KB. Data is compressed both on the disk and in the buffer pool. The log records for compressed data contain the data in

compressed format. Data rows are only compressed when space can be saved. If and how much space can be saved depends on the data as well as on the minimum data record length.

Advantages of Row Compression

Row compression offers the following advantages:

- The size of table data can be reduced significantly. As a result, you can reduce the amount of disk space and therefore, the total cost of ownership (TCO).
- Since the data is also compressed in the buffer pool, the buffer pool hit ratio is increased.
- Log records are smaller (except for update operations where they might even increase in size).
- Less I/O data transfer is needed due to smaller data records.

After the compression dictionary has been created, newly inserted data is compressed automatically. The compression dictionary remains in the table even if all the data is deleted. As soon as new data is inserted, it is compressed with the existing dictionary. This works well unless the new data follows completely different patterns than the one from which the dictionary was originally created.

Constraints

- Row compression requires approximately 10% of additional CPU resources for compressing and decompressing the data. However, in many cases, the additional need for user CPU time is offset by savings in system CPU time that is caused by the reduction in I/O.
- Database heap:
  Memory for compression dictionaries is allocated from the database heap. On average about 75 KB is needed for the creation of one dictionary.
- Utility heap:
  Building compression dictionaries requires a temporary in-memory buffer of 10 MB that is allocated from the utility heap.
- Longer log records might be required for update operations.
- Update operations can result in increased fragmentation because a compressed and updated row might be longer than the compressed original row. Therefore, the updated row might no longer fit into the slot occupied by the original row.
- You cannot compress the system catalog.
- Db2 automatically compresses temporary tables if they are eligible for compression.
- Row compression is compatible with the Db2 replication feature.

Automatic Dictionary Creation (ADC)

A compression dictionary is automatically created if the following conditions are met:

- The compression flag of the table to be compressed is set to *YES*.

- The table does not yet have a compression dictionary.
- The table contains approximately 1 MB of data or more.

The compression dictionary is created as part of an `INSERT`, `LOAD`, or `IMPORT` operation as a synchronous action. After the dictionary has been created, newly incoming data is compressed. Data that was inserted into the table before the dictionary was created remains uncompressed.

7.5.1.2.1 Checking for Row Compression

You can check for row compression using the DBA Cockpit, an SQL statement, or the table functions `ADMIN_GET_TAB_COMPRESS_INFO` (Db2 V9.7) and `ADMIN_GET_TAB_DICTIONARY_INFO` (as of Db2 10.1):

DBA Cockpit

As of Enhancement Package 2 for SAP NetWeaver 7.0 SP6, you can display an overview list of all tables that are already compressed and that are candidates for compression.

To do so, call transaction `DBACOCKPIT` and choose *Space* > *Tables and Indexes* > *Compression Status* and *Compression Candidates*.

For more information, see the DBA Cockpit documentation [page 228].

SQL Statement

To check if row or value compression is activated for a specific table, use the following SQL statement:

```
db2 "select compression from syscat.tables where tabschema = 'SAP<SAPSID>' and
tabname = '<tablename>'"
```

The following values are possible:

Value	Description
N	No compression is activated.
R	Row compression is activated.
V	Value compression is activated and a row format that supports compression is used.
B	Value and row compression are activated.

Table Functions ADMIN_GET_TAB_COMPRESS_INFO and ADMIN_GET_TAB_DICTIONARY_INFO

With Db2 V9.7, you can retrieve detailed information about the compression status of a table and the compression dictionary (including the creation time of an automatically created dictionary) using the table function `ADMIN_GET_TAB_COMPRESS_INFO`. For more information, see ADMINTABCOMPRESSINFO administrative view and ADMIN_GET_TAB_COMPRESS_INFO_V97 table function - returns compressed information in the IBM documentation.

As of Db2 10.1, you use the table function `ADMIN_GET_TAB_DICTIONARY_INFO` for this purpose. For more information, see ADMINTABCOMPRESSINFO administrative view and ADMIN_GET_TAB_COMPRESS_INFO table function (deprecated) - returns compressed information in the IBM documentation.

7.5.1.2.2 Enabling Tables for Row Compression

You can enable tables for row compression using either the DBA Cockpit or the Db2 command line.

In the DBA Cockpit, up to SAP Netweaver 7.01, you can enable compression for a single table by using the *Compress* function on the *Space* > *Single Table Analysis* screen.

As of SAP Netweaver 7.02 SP6, you can enable compression for a set of tables. To do so, go to the *Space* > *Compression Candidates* screen in the DBA Cockpit.

For more information, see the DBA Cockpit documentation [page 228].

7.5.1.3 Adaptive Compression

As of Db2 10.1, table data can be compressed using adaptive compression. This compression technique comprises the classic row compression and a new compression algorithm that works on page level. The new algorithm searches for repeated byte patterns within one page and replaces them with shorter symbols. Similar to the compression dictionary on table level used by the classic row compression approach, a page-level dictionary is used by new adaptive compression algorithms to translate between the symbols and the byte pattern. In contrast to the static table-level dictionary for row compression, the page-level dictionary is adapted automatically when new data is inserted or deleted.

In Db2 10.1, adaptive compression and inplace table reorganization are not compatible. Tables that use adaptive compression cannot be reorganized with a `REORG <table> INPLACE...` command. As of Db2 10.5, this restriction no longer exists.

Prerequisites

Before you use adaptive compression, make sure that the ABAP Dictionary contains the corrections from SAP Note 1701181.

More Information

Adaptive Compression in the IBM Db2 documentation.

7.5.1.3.1 Checking for Adaptive Compression

Use

To check for adaptive compression, you can use one of the following:

- DBA Cockpit
- SQL statement

Procedure

Using the DBA Cockpit

As of Enhancement Package 2 for SAP NetWeaver 7.0 SP6, you can display an overview list of all tables that are already compressed and that are candidates for compression.

To do so, call transaction `DBACOCKPIT` and choose *Space* > *Compression Status* or *Compression Candidates*.

For more information, see the DBA Cockpit documentation [page 228].

Using an SQL Statement

To find out which tables in your database are using adaptive compression, enter the following SQL statement:

```
SELECT tabname FROM syscat.tables WHERE tabschema = <schema> AND rowcompmode = 'A'
```

In this statement, replace `<schema>` with the database connect user (`SAPR3` or `SAP<SAPSID>`) in upper case.

> **i Note**
>
> If you enable table compression with `COMPRESS YES` as of Db2 10.5, adaptive compression is used by default.

7.5.1.3.2 Enabling Tables for Adaptive Compression

You can enable tables for adaptive compression using either the DBA Cockpit or the DB6CONV report.

To search for candidates for adaptive compression in the DBA Cockpit, call transaction `DBACOCKPIT`. On the *Database* tab page or in the SAP GUI navigation frame, choose *Space* > *Compression Candidates* (as of

SAP NetWeaver 7.02 SP12, 7.03 SP 5, 7.3 SP 8, 7.31 SP5, and 7.4 SP 2). For more information, see the DBA Cockpit documentation in References [page 228].

If you want to use the DB6CONV report to convert an existing table to a table that uses adaptive compression, see SAP Note 1513862 and the DB6CONV documentation attached to that SAP Note.

7.5.1.4 Global Compression Option

To specify whether all newly created tables are compressed, you can set the global compression option (Db2 global variable `SAP<SID>.GLOBAL_COMPRESSION_OPTION`).

The system checks whether the Db2 global variable `SAP<SID>.GLOBAL_COMPRESSION_OPTION` exists. If it does, new tables are compressed (or not) depending on the value of the variable (*YES*, *YES ADAPTIVE*, *YES STATIC*, or *NO*). If the variable is set to *YES*, the compression type (static or adaptive) depends on the Db2 registry variable `DB2_ROWCOMPMODE_DEFAULT`. The default of the registry variable `DB2_ROWCOMPMODE_DEFAULT` up to and including Db2 10.1 is static row compression. As of Db2 10.5, the default is adaptive compression. If the global compression option is set to *YES STATIC* or *YES ADAPTIVE*, the compression type is as defined (static or adaptive) and the setting of the Db2 registry variable is ignored.

As of SAP NetWeaver 7.0 SR3, you can enable row compression during the installation of SAP systems by selecting *Use Db2's Row Compression* in the appropriate dialog of the SAP installation tool. This sets the global compression option to `YES`. For more information, see your relevant SAP system installation guide.

If you have not enabled compression during installation, you can do so later using the command line or the DBA Cockpit. In this case, compression is enabled for **newly** created tables **only**.

More Information

- SAP Note 1690077
- SAP NetWeaver installation guides

7.5.2 Index Compression

As of Db2 V9.7, you can use index compression to minimize disk space required by the indexes of a table. To compress indexes, the Db2 database manager uses various index compression techniques, for example, variable slot directory, RID list compression, and prefix compression.

During an upgrade from an older Db2 release to Db2 V9.7, indexes are not compressed. Therefore, after an upgrade, there might be compressed tables with uncompressed indexes in the database. Whenever an index is created with Db2 V9.7, the index inherits the current compression attribute from the table it belongs to, that is:

- If the table is created with the attribute `COMPRESS YES`, all its indexes are also compressed by default.
- If a table is created without the attribute `COMPRESS YES` or with `COMPRESS NO`, all its indexes are **not** compressed, either.

- If an existing table is altered using the SQL statement `ALTER TABLE COMPRESS [YES NO]`, all indexes that are created after the `ALTER` statement was executed inherit the new compression status.

You can also **explicitly** specify the compression status for every index using the SQL statement `ALTER INDEX...COMPRESS [YES NO]`.

For more information, see Enabling Indexes for Compression [page 69].

7.5.2.1 Checking for Index Compression

You can check for index compression using the DBA Cockpit (as of SAP NetWeaver 7.02 SP6), an SQL statement, or the table function `ADMIN_GET_INDEX_COMPRESS_INFO`.

DBA Cockpit

You can check for index compression as you check for row compression, that is, in the DBA Cockpit, choose *Space* > *Compression Status*.

Index and row compression are not handled separately, which means:

- If any of the compression mechanisms are active, a table is assumed to be compressed. The corresponding compression rate is an overall compression rate taking into account compression savings of row and index compression.
- If either row or index compression can improve the overall compression savings, a table is considered to be a candidate for compression.

SQL Statement

To check the value of the index compression flag for a particular index, you can use the following SQL statement:

```
SELECT compression FROM syscat.indexes WHERE tabschema = 'SAP<SAPSID>' AND indname
= ‘<index_name>’
```

The following values are possible:

Value	Description
Y	Index compression is enabled **i Note** The value *Y* does not necessarily mean that the selected index is currently compressed but that it will be compressed after an index reorganization. To check if the selected index is currently compressed, you can use the table function `ADMIN_GET_INDEX_COMPRESS_INFO`.
N	Index compression is not enabled

Table Function ADMIN_GET_INDEX_COMPRESS_INFO

To check if an index is currently compressed on disk, you can use the following `SELECT` statement:

```
SELECT index_compressed FROM TABLE(sysproc.admin_get_index_compress_info('I',
'SAP<SAPSID>', '<index_name>', -2, -2)) AS T
```

The statement assumes that the database is not partitioned and the table to which the index belongs is not range-partitioned.

For more information, see ADMIN_GET_INDEX_COMPRESS_INFO table function - returns compressed index information in the IBM documentation.

7.5.2.2 Enabling Indexes for Compression

You can enable indexes for compression either using the DBA Cockpit (as of SAP NetWeaver 7.02 SP6), implicitly (when compressing a table), or explicitly using the Db2 command line.

DBA Cockpit

In the DBA Cockpit, go to the *Space* > *Compression Candidates* screen where tables without index compression are flagged as compression candidates (you cannot explicitly enable indexes for compression here).

For more information, see the DBA Cockpit documentation [page 228].

Implicitly Compressing Indexes

As of Db2 V9.7, whenever row compression is activated for a table, index compression is also activated for all its indexes (see also Enabling Tables for Row Compression [page 65]).

Explicitly Compressing Indexes (Db2 Command Line)

Even if a table to which an index belongs is not compressed, you can explicitly enable the index of this table for compression. Similar to table compression, this is a two-step approach: First you have to set the compression flag to value *Y*. Then this index has to be reorganized.

To explicitly compress a single index, you can use the following statements:

```
ALTER INDEX <index_name> COMPRESS YES
```

```
REORG INDEXES ALL FOR TABLE <table_name> ALLOW WRITE ACCESS
```

More Information

SAP Note 1379984

7.5.3 Special Considerations About Compression in an SAP Environment

The following applies for compression in an SAP environment:

- Row compression is supported and recommended for SAP systems running on Db2.
- SAP Business Warehouse fully supports row compression as of SAP NetWeaver Business Warehouse 7.0. For more information, see the database administration guide *SAP Business Warehouse on IBM Db2 for Linux, UNIX, and Windows: Administration Tasks* under References [page 228] in the appendix.
- R3load provides you with options to compress tables during installation and system copies. For more information, see SAP Notes 905614 and 1058437.
- As of SAP NetWeaver 7.0 SR3, you can decide to use Db2's row compression by selecting *Use Db2's Row Compression* on the appropriate dialog of the SAP installation tool during the dialog phase of the SAP system installation. Depending on your Db2 version, static row compression, index compression, and adaptive compression are supported.
  For more information, read chapter *Optimization of Database Size on Disk* in the appropriate installation guide for your SAP release (see References [page 228]) .
- The DBA Cockpit supports row compression including the following:
  - Collection of compression check results
  - Determination of compression candidates
  - Enabling of row compression for a set of tables (classic row compression or adaptive compression)

- Switching on the global compression option

The support depends on the SAP NetWeaver release and the support package of your SAP system. For more information, see the DBA Cockpit documentation under References [page 228].

7.5.4 Compression of Archived Log Files

As of Db2 10.1, you can compress archived log files if your database is enabled for rollforward recovery and your log file archiving method is DISK, TSM, or VENDOR. Log file compression leads to additional disk space savings in your Db2 database environment.

Enabling the Compression of Archived Log File

To enable the compression of archived log files, set the database configuration parameters `logarchcompr1` and `logarchcompr2` (if a secondary log archive destination is specified using `LOGARCHMETH2`) to *ON* with the following commands:

```
UPDATE DB CFG USING logarchcompr1 ON
UPDATE DB CFG USING logarchcompr2 ON
```

> **i** Note
>
> A database restart is not necessary.

More Information

Archived Log File Compression in the IBM documentation

7.6 Range-Partitioned Tables

Range partitioning, also known as table partitioning, means that a large table is split into data partitions, which are also known as ranges, based on a partitioning key. Each partition constitutes its own storage object and can be detached from the table (roll-out) or attached to the table (roll-in) with minimal effort. Query performance may improve because Db2 only scans partitions that are relevant to the query (data partition elimination). Range partitioning is completely transparent to database applications. The benefits of range-partitioned tables are, for example:

- Easier maintenance of large database objects, for example, through a partition-level REORG
- Separation between operational (hot) and aged (cold) data
- Performance improvements

Indexes can also be partitioned together with the data partitions. In an SAP ABAP system, table range partitioning is supported for all tables except BW tables. As of Db2 10.5, you can use Db2 storage groups to distribute partitions based on their temperature (that is, hot or cold data) to different storage media.

Prerequisites

Your system must meet the requirements mentioned in **SAP Notes** 1379362 and 1612416.

Enabling Range Partitioning for an Existing Table

To partition an existing table that contains data, add the partitioning clause as storage parameters to the table in the ABAP Dictionary (transaction SE14) and then perform an online table move.

Additionally, the DB6 partitioning administrator tool is available with SAP Note 1686102. This tool creates partitions for some of the largest OLTP tables. For more information, see the following section and the documentation attached to SAP Note 1686102.

Table Partitioning Using the DB6 Partitioning Administrator

The DB6 partitioning administrator is an ABAP-based tool for SAP OLTP systems that run on IBM Db2. It helps you to perform table range partitioning on a specified set of tables that usually grow large. The tool supports table partitioning based on the following criteria:

- Fiscal year
  A certain set of tables can be partitioned based on the fiscal year.
- Numbers from number ranges
  Since time-dependent criteria such as the fiscal year are not always available, other tables can be partitioned based on numbers drawn from number ranges, which are usually also an indicator of the age of the data.

Both partitioning criteria enable you to distinguish between operational (hot) and aged (cold) data and to separate them into different table partitions. Furthermore, as of Db2 version 10.1, the tool allows you to assign the calculated partitions to storage groups and to move a partition from one storage group to another. Storage groups are a new concept for automatic storage databases that was introduced with Db2 10.1. For more information about storage groups, see section *Multi-Temperature Data Management and Storage Groups* in Tablespaces, Containers, and the File System [page 53].

Range-partitioned tables offer the following benefits:

- Easier maintenance of large database objects
- Separation between hot and cold data
- Improvements of query performance

The DB6 partitioning administrator provides the following functions:

- Calculation of suitable partitions for 30 of the largest SAP OLTP tables (report RSDB6PARTGEN) and the assignment of the calculated partitions to storage groups.
- Range partitioning for these tables with the DB6CONV report

- Detailed information about the partitions of a partitioned table (report `RSDB6PARTMON`)
- Movement of all tablespaces of a partition to another storage group (report `RSDB6PARTMOVE`)

For more information - also about prerequisites and limitations - of the DB6 partitioning administrator, see the detailed documentation that is attached to SAP Note 1686102.

You can install the DB6 partitioning administrator by importing the relevant transport that is attached to SAP Note 1686102 into your SAP system. You also have to update the DB6CONV report to the latest available version (see SAP Note 1513862). At least version 5.10 of DB6CONV is required.

Identifying Range-Partitioned Tables in the Database

To find out which tables in your database are range-partitioned, enter the following SQL statement:

```
SELECT UNIQUE(tabname) FROM syscat.datapartitionexpression WHERE tabschema =
<schema>
```

In this statement, replace `<schema>` with the database connect user (`SAPR3` or `SAP<SAPSID>`) in upper case.

7.7 Insert Time Clustering (ITC) Tables

As of Db2 10.1, a new table type allows you to cluster the records of a table based on their insert time. Insert time clustering (ITC) tables are based on the infrastructure for multidimensional clustering (MDC) tables. The main advantage of ITC tables is the lightweight storage reclamation within the tablespace. With the `REORG TABLE...RECLAIM EXTENTS` statement, all unused extents are given back to the tablespace.

As of Db2 10.5, space reclamation from ITC tables using `REORG TABLE...RECLAIM EXTENTS` has been extended to reclaim space also from extents that are not completely empty. Db2 consolidates all extents in the table that are below a certain level of use into a smaller number of extents and then frees all completely empty extents. After an upgrade, you can convert existing tables to ITC tables using the `DB6CONV` report. As of Version 5.20, the `DB6CONV` report offers the option to find suitable table candidates and convert them to ITC tables.

For more information, see the `DB6CONV` documentation that is attached to SAP Note 1513862.

> **i** Note
>
> If you are using the Db2 pureScale Feature, ITC tables are **not** supported.

Prerequisites

Before you use ITC tables, make sure that the following prerequisites are met:

- The ABAP Dictionary contains the corrections from SAP Note 1701181.
- You are using `DB6CONV` Version 5.20 or higher (see SAP Note 1513862).

Converting Existing Tables to ITC Tables

For information about converting existing tables to ITC tables, see the `DB6CONV` documentation attached to SAP Note 1513862.

Identifying ITC Tables in Your Database

To find out which tables in your database are ITC tables, enter the following SQL statement:

```
SELECT tabname FROM syscat.tables WHERE tabschema = <schema> AND clustered = 'T'
```

In this statement, replace `<schema>` with the database connect user (`SAPR3` or `SAP<SAPSID>`) in upper case.

More Information

SAP Note 1700631

7.8 Column-Organized Tables (IBM Db2 With BLU Acceleration)

As of Db2 version 10.5, you can use Db2 BLU Acceleration, which comprises the following technologies:

- Column-organized table storage and columnar processing
- Column data compression
- Parallel processing
- Data skipping

Traditional tables are stored and processed row by row. By storing and processing the values of the columns instead of the values of the rows of a table together, analytical queries can be processed much faster.

Column-organized tables are created by adding the `ORGANIZE BY COLUMN` clause to the `CREATE TABLE` statement.

Note the following if you want to use column-organized tables in SAP systems:

- It's only supported for certain types of SAP BW tables and for a few other usage scenarios (see SAP Note 1819734). Converting standard non-BW tables and converting tables outside the defined usage scenarios to column-organized tables is **not** supported.
- It requires changes in the Db2 configuration.

For more information about the restrictions and requirements for the use of column-organized tables, see the following documentation:

- Restrictions, limitations, and unsupported database configurations for column-organized tables in the IBM documentation
- SAP Note 1819734 *DB6: Use of BLU Acceleration*
- Database administration guide SAP Business Warehouse on IBM Db2 for Linux, UNIX, and Windows: Administration Tasks on SAP Help Portal

7.9 Converting Tables Using Report DB6CONV

Using report DB6CONV, you can convert a single table or multiple tables. Table conversion in this document means that you move one or more tables within the same tablespace or to another tablespace as fast and efficient as possible using Db2 means without explicitly changing the table structure. During the move, you can change certain characteristics of the table, that is, you can do, for example, the following:

- Compress uncompressed tables
- Convert tables to ITC or MDC tables (and re-convert them)
- Range-partition tables
- Convert row-organized tables to column-organized tables and vice versa (for SAP Business Warehouse only)

Moving tables with DB6CONV is also an alternative to table reorganization and index reorganization.

Depending on your database release and SAP system release level, you use one of the following reports:

DB6CONV Version	Required Software Level	SAP Note
6.0 and higher	SAP Basis release 7.0 or higher	1513862
5	SAP Basis release 6.20 or higher	2091519

> **i Note**
>
> The DB6CONV report is **not** part of the standard SAP software delivery. It's only available as attachment to the SAP Notes mentioned in the above table. These SAP Notes also provide the required documentation about how to use the DB6CONV report.

7.10 Dealing with Growing Diagnostic Data

Depending on the settings of the Db2 database manager parameters `diaglevel` and `notifylevel`, the diagnostics log files `db2diag.log` and the notification log can grow considerably in size.

On UNIX, the notification log (`<instance name>.nfy`) is located in the `DIAGPATH` directory. On Windows, the notification records are written to the *Event Viewer* in the *Application* section.

To split the log files, you can simply move them to another directory. Db2 re-creates them and continues to fill them with log data. You can rename the `db2diag.log` and the notification log. You should do this in regular intervals to avoid that these files become too large.

> **i** Note
>
> There are no fixed rules for their maximum size, but investigating diagnostic data of 1 GB or even 100 MB extends the problem investigation.

For the `db2diag.log`, you can also use the `db2diag` command to split the log file at a specific point in time. For example, save it to a backup device and then delete the split part of the log file. To archive the `db2diag.log`, you can use the command `db2diag -A`.

You can schedule regular switching of the diagnostics log files using the DBA Cockpit. To do so, call transaction DBACOCKPIT in your SAP system and choose *Configuration* > *Monitoring Settings*. For more information, see the DBA Cockpit documentation [page 228].

You can also use the database manager configuration parameter `DIAGSIZE` to enable rotating diagnostic and administration log files and thereby limiting their size on disk.

More Information

Diagnostic Tool db2diag [page 164]

8 Backup and Recovery

A database can become unusable because of hardware or software failure, or both. You might encounter storage problems, power interruptions, or application failures, and each failure scenario requires a different recovery action. Therefore, it is absolutely mandatory that you are able to protect your data against the possibility of loss by having a recovery strategy in place.

For more information, see Developing a backup and recovery strategy in the IBM documentation.

Related Information

Enabling the Database for Rollforward Recovery [page 77]
Components of the Db2 Log File Management [page 79]
Database Backup [page 109]
Database Recovery Using the RECOVER Command [page 114]
File System Backups and db2inidb Tool [page 116]
Checking the Database for Consistency [page 123]

8.1 Enabling the Database for Rollforward Recovery

For **production systems**, your database must be in archive logging mode. For test and quality assurance systems, this mode is highly recommended but not mandatory.

Enabling the database for rollforward recovery allows you to make online backups, split- mirror backups, and point-in-time recoveries. In addition, you can use the HADR feature.

In archive logging mode, the active log files, in which all database changes are recorded, are archived by the Db2 log manager as soon as they are filled with log records. The numbering of log files increases with each new log file (up to 9999999).

You enable rollforward recovery by setting the database configuration parameter `LOGARCHMETH1` and optionally `LOGARCHMETH2` according to your needs.

For more information about the configuration of the log file management, see Configuration [page 82].

After setting these parameters, you must perform an offline backup. The backup procedures differ depending on whether you are using a single-partition or multi-partition database system.

More Information

Enabling Rollforward Recovery for a Single-Partition Database [page 78]

Enabling Rollforward Recovery for a Multi-Partition Database [page 78]

8.1.1 Enabling Rollforward Recovery for a Single-Partition Database

To enable your single-partition database for rollforward recovery, follow these steps:

Procedure

1. Log on to the database server as user `db2<dbsid>`.
2. **Windows only:**
   Start the Db2 command window.
3. To update the database configuration parameters, enter the following command:
   ```
   db2 update db cfg for <DBSID> using LOGARCHMETH1 <method>
   ```

8.1.2 Enabling Rollforward Recovery for a Multi-Partition Database

To enable your multi-partition database for rollforward recovery, follow these steps:

Procedure

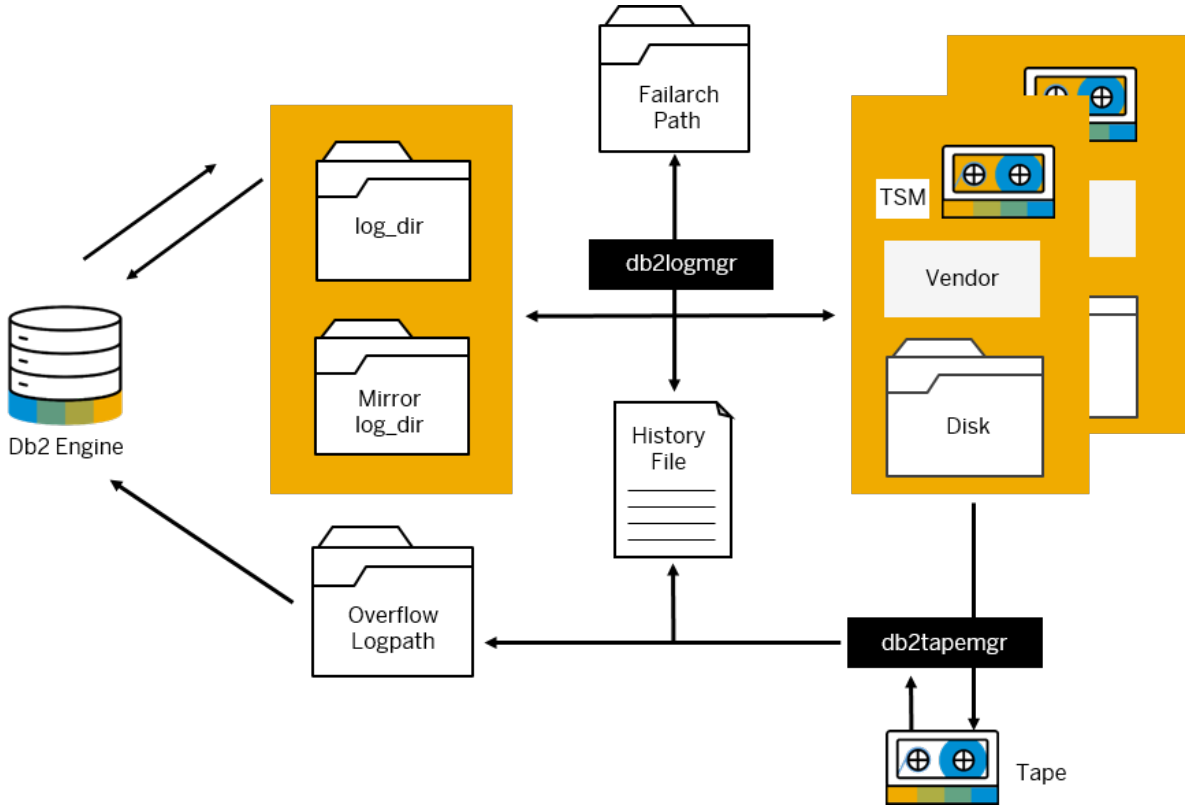
1. Log on to the database server as user `db2<dbsid>`.
2. To update the database configuration parameters, enter the following command:
   ```
   db2 update db cfg for <DBSID> using LOGARCHMETH1 <method>
   ```
3. Restart the database instance.

After the first activation of the database, the value of the database configuration parameter `BACKUP PENDING` is switched to *YES* and you need to perform an offline backup on all partitions. For more information, see Performing the Database Backup [page 111].

8.2 Db2 Log File Management

8.2.1 Components of the Db2 Log File Management

The following figure shows the components of the Db2 log file management, including the Db2 log manager (`db2logmgr`) and the Db2 tape manager (`db2tapemgr`).



Db2 Log File Management

Db2 Log Manager

The Db2 log manager (`db2logmgr`) is the central component for managing log files. It is part of the Db2 engine and responsible for archiving and retrieving log files.

The location of the log files is recorded in the Db2 history file. The Db2 log manager supports the following archiving media:

- TSM (`TSM:<TSM management class>`)
  The Db2 log manager has built-in support for accessing the Tivoli Storage Manager (TSM).
- Disk (`DISK:<path>`)
  The Db2 log manager can use a disk location for archiving log files.
- Vendor library (`VENDOR:<vendor library>`)
  Db2 provides a vendor API for the log file management, which is an extension to the existing backup API. Storage vendors can provide their own library to allow log file management with Db2.

- User exit (`USEREXIT`)
  You can create a user exit program to automate log file archiving. However, the user exit is deprecated and not supported any more by SAP. We recommend that you use one of the other archiving media.

The Db2 log manager supports two archiving locations so that a log file can be stored on two different locations, for example, on TSM and on disk. The Db2 log manager is configured using database configuration parameters.

If log files are retrieved, the Db2 log manager directly retrieves them from the back end and puts them in the Db2 transaction log directory (`log_dir`). From there, the Db2 engine can read them to perform a database recovery or rollforward operation.

If log files cannot be archived to the designated destination, for example, due to a network outage, you can specify a local directory (`FAILARCHPATH`) that is used as intermediate storage for the log files. The Db2 log manager puts log files into the `FAILARCHPATH` if the archiving destination is not available. If the archiving destination becomes available again, the Db2 log manager moves them to the archiving destination. In this way, you can avoid that the transaction log directory becomes full (“log_dir full problem”).

Db2 Advanced Log Space Management (ALSM)

As of Db2 for LUW 11.5 Mod Pack 4 Fix Pack 0, IBM introduced Advanced Log Space Management (ALSM). It helps avoid error SQL0964C, which means that the transaction log for the database is full. SQL0964C can occur if a long-running transaction does not commit and holds up active log space. This might be the case, for example, for long-running ABAP batch programs, which do not or seldomly commit, during the creation of indexes or during a LOAD operation. Using LOAD is an option in the DB6CONV report. For more information, see SAP Note 1513862.

Prior to Db2 11.5 Mod Pack 4 Fix Pack 0, the Db2 log directory contained only active log files with file names according to the following pattern: `S<log nr>.LOG, for example S0000025.LOG.`

ALSM introduces extraction log files that are located in the log directory, too. The following types of extraction files exist:

- `X<log nr>.TMP`
- `X<log nr>.META`
- `X<log nr>_TID<tx nr>.LOG`

Example: `X0000025.META`

If enabled, ALSM copies (= extracts) single log records of uncommitted transactions from active log files into extraction log files as soon as more than 80 percent of the active log space is used up. Extraction log files remain in the log directory until the transaction ends and are deleted automatically. This allows Db2 to remove active log files from the log directory although the transaction is still running.

> **i Note**
>
> When using ALSM, consider spending at least 20 percent of free space of the log directory for the temporary creation of extraction log files.

To turn on ALSM, set the Db2 registry variable `DB2_ADVANCED_LOG_SPACE_MGMT=ON`.

For more information, including possible limitations of ALSM, see https://www.ibm.com/docs/en/db2/11.5?topic=logging-advanced-log-space-management in the IBM Db2 documentation and the SAP blog post about ALSM on SAP Community that also includes a video.

Considerations for Long-Running Transactions

ALSM has primarily been designed to avoid error SQL0964C that occurs due to long-running transactions with low log volume. Transactions with a high log volume that consumes almost all active log space do not benefit from ALSM. Beside ALSM, Db2 provides the following capabilities to handle different characteristics of transactions:

- Database configuration parameter `NUM_LOG_SPAN` allows you to limit the number of log files a transaction can span.
- Database configuration parameter `MAX_LOG` allows you to limit the log volume of a transaction to a percentage of the active log space.
- The threshold configuration parameter `UOWTOTALTIME` of the Db2 workload manager allows you to limit the runtime of transactions to seconds, minutes, hours, or days.

If you are using ALSM, it's very important that you consider `NUM_LOG_SPAN`. The benefit of ALSM is that transactions can span more than the configured active log space. Therefore, you might want to increase `NUM_LOG_SPAN`. A too restrictive setting can lead to an early rollback of the long-running transaction before ALSM extraction takes place on this transaction.

As an alternative to `NUM_LOG_SPAN`, consider setting the `UOWTOTALTIME` workload manager threshold. The `UOWTOTALTIME` threshold allows you to set a time limit on the transaction runtime, regardless of the number of log files spanned by the transaction. If you have already set a `UOWTOTALTIME` threshold, you may want to reconsider and possibly increase it when you enable ALSM.

For more information, see also SAP Notes 1497040 and 1493587.

Db2 Tape Manager

This is an executable that you can call from the command line. The Db2 tape manager can be used to archive Db2 log files to tape. The Db2 log manager cannot directly handle the log files that are stored on tape. You have to call the Db2 tape manager (`db2tapemgr`) **explicitly** from the command line.

With SAP NetWeaver 7.0 SP12, the job *Archive Log Files to Tape* has been integrated in the DBA Planning Calendar. This job allows you to schedule the archiving of log files to tape by the Db2 tape manager in the DBA Planning Calendar.

The log files that are retrieved from the Db2 tape manager are put in the Db2 overflow log path (`OVERFLOWLOGPATH`) and not directly in the transaction log directory (`log_dir`). From there, the Db2 engine can read them to perform a database recovery or rollforward operation. If log files are archived to tape, the Db2 tape manager updates the history file. This helps you identify the tapes that are needed for a database recovery.

More Information

Configuration of the Db2 Log File Management [page 82]

Log File Chains [page 84]

8.2.2 Configuration of the Db2 Log File Management

The following table lists the database configuration parameters that control the Db2 log file management configuration.

Parameter	Description
LOGARCHMETH1	Specifies the media type of the primary destination for archived log files Possible values are: DISK:<path> TSM:<TSM management class> VENDOR:<vendor library> USEREXIT
LOGARCHMETH2	Specifies the media type of the secondary destination for archived log files If this variable is specified, log files are archived to both this destination and the destination that is specified by the database configuration parameter LOGARCHMETH1. **i Note** Only the destinations DISK, TSM, and VENDOR are allowed for this parameter.
LOGARCHOPT1	Specifies the options for the primary destination specified in LOGARCHMETH1 for archived log files (if required). You can use this parameter, for example, to specify an additional TSM parameter, such as **-fromnode <node> -fromowner <owner>**. As of Db2 11.1 Mod 3 FP3 iFix001SAP, to avoid a potential hang, you can specify a vendor archive timeout by using the following command: **db2 update db cfg for sample using LOGARCHOPT1 "--vendor_archive_timeout=<NumberOfSeconds>"**

Parameter	Description
LOGARCHOPT2	Specifies the options for the secondary destination specified in LOGARCHMETH2 for archived log files (if required). As of Db2 11.1 Mod 3 FP3 iFix001SAP, to avoid a potential hang, you can specify a vendor archive timeout by using the following command: `db2 update db cfg for sample using LOGARCHOPT1 "--vendor_archive_timeout=<NumberOfSeconds>"`
LOGARCHCOMPR1	As of Db2 10.1: Allows you to turn on log compression during archiving to the destination specified in LOGARCHMETH1
LOGARCHCOMPR2	As of Db2 10.1: Allows you to turn on log compression during archiving to the destination specified in LOGARCHMETH2
FAILARCHPATH	Intermediate location for log files that cannot be archived to either the primary or (if set) the secondary archiving destinations (because of a media problem affecting these destinations) **i Note** The specified path must reference a disk location.
NUMARCHRETRY	Specifies the number of times that Db2 tries to archive a log file to the primary or the secondary archiving destination before trying to archive the log file to the failover directory This parameter is only used if the FAILARCHPATH database configuration parameter is set. If NUMARCHRETRY is **not** set, Db2 continues to try archiving to the primary or the secondary log archiving destination.
ARCHRETRYDELAY	Specifies the number of seconds Db2 has to wait after a failed archiving attempt before trying to archive the log file again Subsequent retries only take effect if the value of the NUMARCHRETRY database configuration parameter is at least 1.

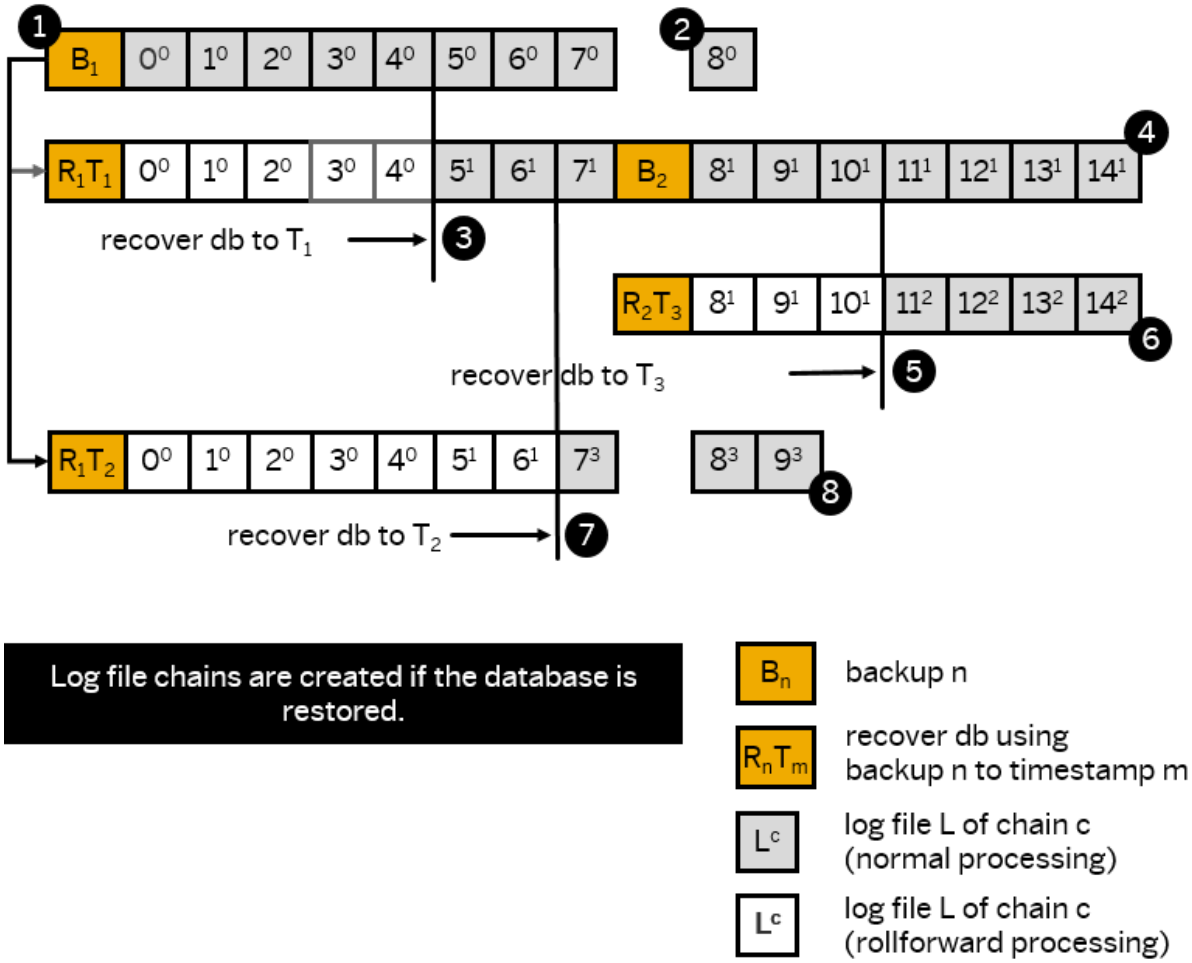
Parameter	Description
`OVERFLOWLOGPATH`	Points to the directory into which the Db2 tape manager stores log files and specifies an additional location for Db2 to find log files that are needed for a rollforward operation

> **i** Note
>
> There is an additional Db2 registry variable called `DB2_TAPEMGR_TAPE_EXPIRATION`. This variable specifies when it is allowed to overwrite log file tapes. The value DB2_TAPEMGR_TAPE_EXPIRATION defines the number of days before the tape can be overwritten. By setting this variable, you can avoid that you overwrite log files on tape that are still needed for a database recovery.

8.2.3 Log File Chains

The Db2 transaction log files have consecutive names from `S0000000.LOG` to `S99999999.LOG`. If a log file is full, Db2 creates a new log file with the next number. In some special cases (for example, after a point-in-time recovery) Db2 can create log files with the same name but different contents. These log files will automatically be archived in a new directory under the name of the new log file chain.

The following figure describes a possible scenario with different log file chains.



Log File Chains

All the information of log file locations is stored in the history file. The figure above shows the content of the history file if the following steps have been performed:

1. Backup B1 is created.
2. Transactional work on the database creates log files 0 – 8, which belong to log file chain 0.
3. A database recovery to point in time T1 is performed. This is done by using backup image 1 and applying log files 0 - 4.
4. Transactional work on the database creates log files 5 -14, which belong to log file chain 1.
5. A database recovery to point in time T3 is performed. This is done by using backup image 2 and applying log files 8 - 10 of log file chain 1.
6. Transactional work on the database creates log files 11 - 14, which belong to log file chain 2.
7. A database recovery to point in time T2 is performed. This is done by using backup image 1 and applying log files 0 - 4 of log file chain 0 and log files 5 - 6 of log file chain 1.
8. Transactional work on the database creates log files 7 - 9, which belong to log file chain 3.

The example shows that the log file chaining ensures that you can recover the database to any point in time with the right set of log files.

8.2.4 Db2 Log Manager Back-End Support

8.2.4.1 Disk

You activate log file archiving to disk by using the prefix `DISK` for the database configuration variables `LOGARCHMETH1` and `LOGARCHMETH2`. The general format of the value is:

`DISK:<log_archive>`

> ⚠ Caution
>
> The directory `<log_archive>` must exist before you enter this command.

> ✤ Example
>
> The following command sets the log archiving method 1 for the database PRD to directory `/db2/PRD/log_archive`:
>
> ```
> db2 update db cfg for PRD using logarchmeth1 DISK:/db2/PRD/log_archive
> ```

The log files will be stored in a hierarchy of subdirectories under the path specified for the log archiving method `<log_archive>`. The hierarchy looks as follows:

`<log_archive>/<instance>/<database>/NODEwwww/LOGSTREAMxxxx/Cyyyyyyy/Szzzzzzz.LOG`

This hierarchy avoids that log files are overwritten by other database instances.

Subdirectory	Description
`<log_archive>`	Path specified by the database configuration parameter `LOGARCHMETH1` or `LOGARCHMETH2`
`<instance>`	Name of the database instance
`<database>`	Name of the database identifier
`NODEwwww`	Partition for which the log file was created `wwww` are digits from 0 - 9 and specify the partition number.

Subdirectory	Description
`LOGSTREAMxxxx`	Member of a Db2 pureScale cluster for which the log file was created xxxx are digits from 0 - 9 and specify the member number. **Note** The `LOGSTREAM` characteristic of a log file has been introduced with Db2 V9.8 and Db2 10.1. If you use former versions of the Db2 software, the `LOGSTREAMxxxx` information does not exist in the respective directory structures.
`Cyyyyyyy`	Describes to which log file chain the log files belong yyyyyyy are digits from 0-9 and specify the log file chain number.
`Szzzzzzz.LOG`	Name of log file zzzzzzz are digits from 0-9 and specify the log file number.

Example

```
/db2/PRD/log_archive/db2prd/PRD/NODE0000/LOGSTREAM0000/C0000000/S0000000.LOG
```

An entry in the history file for the log file that was archived to disk looks as follows:

```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID
-- --- ------------------ ---- --- ------------ ------------ --------------
X  D   20130521170957     1    D   S0000000.LOG C0000000
------------------------------------------------------------------------
------------------------------------------------------------------------
Comment:
Start Time: 20130521170957
End Time: 20130521192341
Status: A
------------------------------------------------------------------------
EID: 2 Location: /db2/PRD/log_archive/db2prd/PRD/NODE0000/LOGSTREAM0000/C0000000/
S0000000.LOG
```

In this example, field `Type` contains `1`, which stands for `LOGARCHMETH1`. The device field `Dev` contains `D`, which stands for disk, and the `Location` field contains the fully qualified path of the log file.

8.2.4.2 Tivoli Storage Manager (TSM)

Db2 has integrated back end support for archiving backups to TSM and also provides this support for the Db2 log file management. You activate log archiving to TSM by using the prefix `TSM` for the database configuration variables `LOGARCHMETH1` and `LOGARCHMETH2`.

In addition, you can specify a TSM management class (the default is the default TSM management class) as well as additional TSM parameters by using the `LOGARCHOPTS1` and `LOGARCHOPTS2` database configuration variables.

> ### Example
>
> The log archiving method for TSM is set to the TSM management class `LOGMGMTCLS`. With the second command, the TSM option **`-fromnode`** will be set.
>
> ```
> db2 update db cfg for PRD using logarchmeth1 TSM:LOGMGMTCLS
> ```
>
> ```
> db2 update db cfg for PRD using logarchopts1 "-fromnode prdsystem"
> ```
>
> The specified TSM management class **must** have a TSM *archive copy group* defined. Db2 backups are stored to TSM using the TSM backup copy group. The Db2 log files are stored to TSM using the TSM archive copy group.

In Version 9.8 Fix Pack 3 and later fix packs, the log files on the TSM server are written to the `./NODE****/LOGSTREAM****/C0******/` directory.

An entry in the history file for the log file stored in TSM looks as follows:

```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID

-- --- ------------------ ---- --- ------------ ------------ --------------

X D 20130530170957 1 A S0000006.LOG C0000003

----------------------------------------------------------------------------

----------------------------------------------------------------------------

Comment: –fromnode prdsystem

Start Time: 20130530170957

End Time: 20130531082231

Status: A

----------------------------------------------------------------------------

EID: 31 Location: LOGMGMTCLS
```

Field `Type` contains `1`, which stands for `LOGARCHMETH1`. The device field `Dev` contains `A`, which stands for Adstar Storage Manager (the former name of TSM). The `Location` field contains the used TSM management class. In the `Comment` field, you can see the first 30 characters of the `LOGARCHOPTS1` or `LOGARCHOPTS2` database configuration parameters.

To query, extract, or delete log files from TSM, you can use the Db2 tool `db2adutl`. For information about the *db2adutl* tool, go to the IBM Db2 documentation at https://www.ibm.com/docs/en/db2 .

> ### ⚠ Caution
>
> However, if you want to delete log files in TSM, we strongly recommend that you use the Db2 **`prune`** command. In this way, the entries in the history file are also updated. For more information, see *Deleting Entries in the History File* in section History File [page 91].

> **i Note**
>
> The log files are stored with the **database instance owner user**. Therefore, you always have to use the `db2adutl` tool as **database instance owner**. Otherwise, you will not see the log files in TSM.

8.2.4.3 Other Storage Vendors

Log file management support for other vendors is provided through a vendor-specific library from the respective vendor.

You activate log archiving with a vendor library by using the prefix VENDOR and the path to the vendor library for the database configuration variables `LOGARCHMETH1` or `LOGARCHMETH2`. In addition, you can specify vendor-specific options using database configuration variables `LOGARCHOPT1` and `LOGARCHOPT2`. Db2 passes the options, which are set with `LOGARCHOPT1/2`, on to calls to the vendor library.

To set, for example, `LOGARCHMETH1` for archiving with a vendor library, enter the following command:

```
db2 update db cfg for PRD using LOGARCHMETH1 VENDOR: d:\sqllib\bin\db2vendor.dll
```

```
db2 update db cfg for PRD using logarchopt1 "@d:\sqllib\bin\db2vendor.opt"
```

A typical entry in the history file for log files that were archived with a vendor library looks as follows:

```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log  Backup ID
-- --- ------------------ ---- --- ------------ ------------ -- ------
X  D  20130601114218      1    O  S0000007.LOG C0000003
------------------------------------------------------------------------
------------------------------------------------------------------------
   Comment: :@d:\sqllib\bin\db2vendor.opt
Start Time: 20130601114218
 End Time: 20130601115047
      Status: A
 ------------------------------------------------------------------------
EID: 33 Location: d:\sqllib\bin\db2vendor.dll
```

In this example, the field `Type` contains `1`, which stands for `LOGARCHMETH1`. The field `Dev` contains `O`, which stands for another vendor. The `Location` field contains the used vendor library. In the `Comment` field, you can see the first 30 characters of the `LOGARCHOPT1` or `LOGARCHOPT2` database configuration parameters.

As of Db2 11.1 Mod 3 FP3 iFix001SAP, to avoid a potential hang, you can specify a vendor archive timeout by using the following command in the `LOGARCHOPT1` or `LOGARCHOPT2` database configuration parameter:

```
db2 update db cfg for sample using LOGARCHOPT1 "--
vendor_archive_timeout=<NumberOfSeconds>"
```

Example:

```
db2 update db cfg for sample using LOGARCHOPT1 "--vendor_archive_timeout=300"
```

For more information, see also Configuration of the Db2 Log File Management [page 82].

8.2.5 Deleting Archived Log Files

To delete archived log files, Db2 provides various methods, some of which depend on the archiving location. In general, you can use one of the following commands:

- `"PRUNE HISTORY … AND DELETE"`
- `"PRUNE LOGFILE PRIOR TO ..."`

These commands work independently from the archiving location. However, they require an up-to-date history file. For more information about the *PRUNE* commands, see the IBM documentation for your Db2 version.

Deleting Log Files from TSM

If TSM is configured properly, you typically do not need to delete log files from TSM manually. TSM lets you configure the retention time of the log files.

> ⚠ Caution
>
> If you use TSM for the automatic deletion of log files, make sure that you set a retention time that is long enough to ensure that you have all required log files available for database backups. Otherwise, logs that are required in the case of a database recovery might be deleted too early.

To manually delete log files from TSM, you can use the Db2 tool `db2adutl` with the `DELETE LOGS` options.

For more information about the Db2 tool `db2adutl`, see the IBM documentation for your Db2 version.

Automatic Log File and Backup Retention

To configure Db2 to automatically delete unneeded recovery objects after every full database backup, you can use the database configuration parameter `AUTO_DEL_REC_OBJ` and automated pruning of the recovery history file.

After every successful full (that is, non-incremental) database backup, the database manager prunes the recovery history file according to the values of the database configuration parameters `NUM_DB_BACKUPS` and `REC_HIS_RETENTN`.

If there are more database backup entries in the recovery history file than the value of the configuration parameter `NUM_DB_BACKUPS`, the database manager will prune entries that are older than the value of the `REC_HIS_RETENTN` configuration parameter and that do not have the status `DB2HISTORY_STATUS_DO_NOT_DEL` from the recovery history file.

If you set the `AUTO_DEL_REC_OBJ` database configuration parameter to *ON*, the database manager - in addition to pruning entries from the recovery history file - will delete the following:

- Physical log files associated with the pruned entries
- Backup images associated with the pruned entries
- Load copy images associated with the pruned entries

If there are no full database backup images available in the current recovery history (if none were ever taken), backup images that are older than the range of time specified by the configuration parameter `REC_HIS_RETENTN` will be deleted as well (provided they are not in the range of `NUM_DB_BACKUPS`).

If the database manager cannot delete a file because the file is no longer at the location that is listed in the recovery history file, the database manager will prune the entry in the history file accordingly.

If the database manager cannot delete a file because of a communication error between the database manager and the storage manager or storage device, the database manager will **not** prune the entry in the history file. When the error is solved, the file can be deleted during the next automated prune operation.

To configure the database manager to automatically delete unneeded recovery objects, proceed as follows:

1. Set the database configuration parameter `AUTO_DEL_REC_OBJ` to *ON*.
2. To enable automated pruning of the recovery history file, set the configuration parameters `REC_HIS_RETENTN` and `NUM_DB_BACKUPS`.

More Information

For more information about deleting archived log files, go to the IBM documentation at https://www.ibm.com/docs/en/db2 .

8.2.6 History File

The history file contains information about the location of archived log files. Log file entries are created when a new log file is used by the database during normal operation or when a log file is applied during a database rollforward.

To list the log file information on the command line, you can use the Db2 command `list history`:

```
db2 list history archive log all for <dbsid>
```

A sample output of this command looks like this:

```
D:\>db2 list history archive log all for sample
                    List History File for sample
Number of matching file entries = 103...
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log  Backup ID
-- --- ------------------ ---- --- ------------ -
X  D  20130327125117      1    D  S0000000.LOG C0000000
----------------------------------------------------
----------------------------------------------------
   Comment:
Start Time: 20130327125117
 End Time: 20130327125932       Status: A
 ----------------------------------------------------
EID: 2 Location:
e:\log_archive\DB2\SAMPLE\NODE0000\LOGSTREAM0000\C0000000\S0000000.LOG...
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log  Backup ID
-- --- ------------------ ---- --- ------------ -
X  D  20130327142209      N  S0000024.LOG
----------------------------------------------------
----------------------------------------------------
   Comment: ARCHIVE LOG
```

```
Start Time: 20130327142209
 End Time: 20130327142209
      Status: A
 ------------------------------------------------
EID: 31
```

The sample output of this command has two entries:

- The first entry refers to an automatically archived log file.
- The second entry displays the result of the **`db2 archive log for db <dbsid>`** command; `ARCHIVE LOG` in the *Comment* field and `N` in the `Type` field in the entry for the Db2 command **`archive log for db`**.

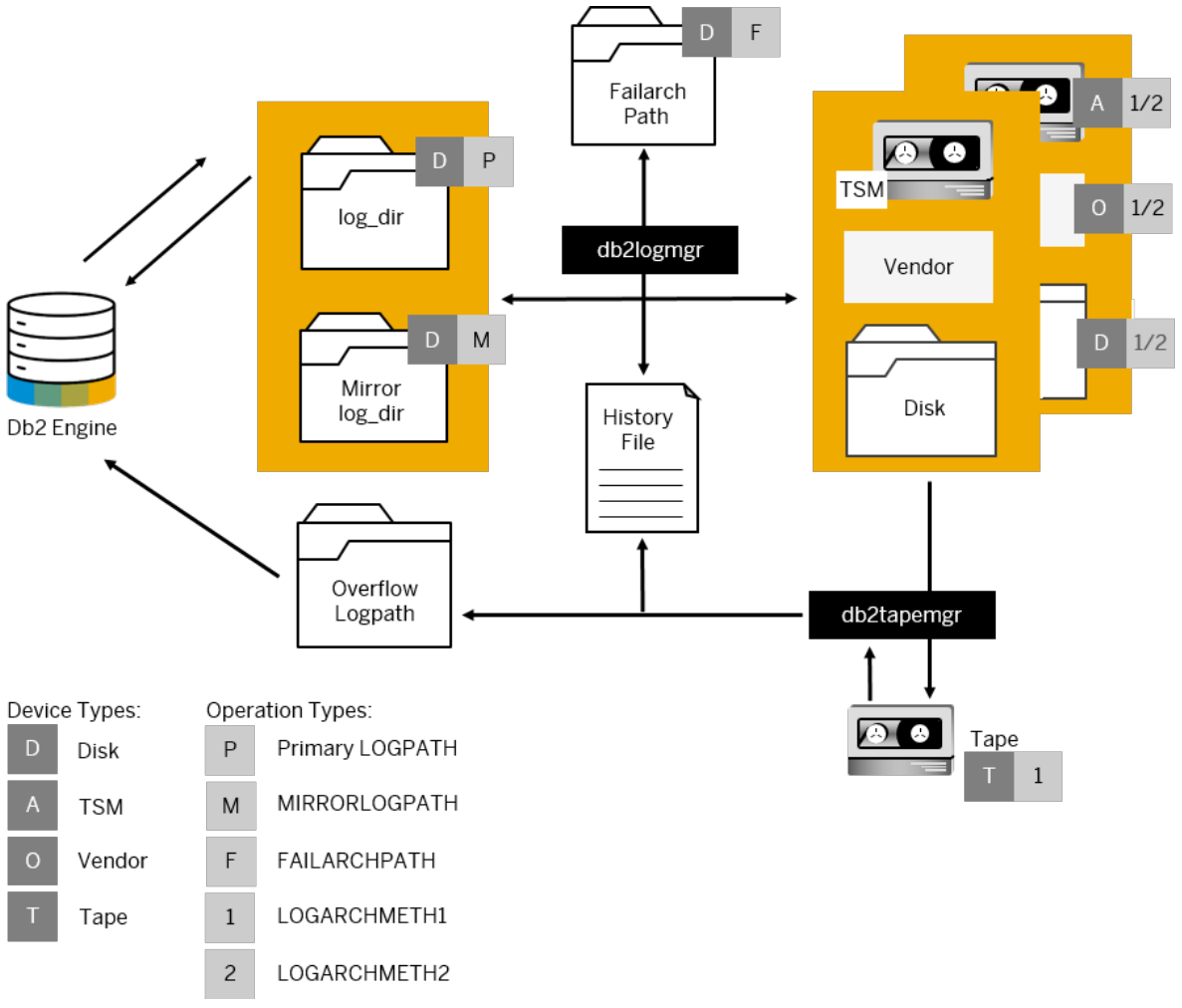
For archived log file entries, the following fields are used:

Field	Contents
`Op` (Operation)	Always `X`
`Obj` (Object)	Always `D`
`Timestamp+Sequence`	A 14-digit timestamp that indicates the log file creation time during normal operation. The sequence number is not used. If the log file entry was created during rollforward, the field contains the time when the log file was applied.
`Type`	The following types are possible: • `P (Primary LOGPATH)` • `M (MIRRORLOGPATH)` • `F (FAILARCHPATH)` • `1 (LOGARCHMETH1)` • `2 (LOGARCHMETH2)` • `N (ARCHIVE LOG` command)
`Dev` (Device)	The following devices are possible: • `D` (Disk) • `A` (TSM) • `O` (Vendor) • `U` (User exit) • `T` (Tape)
`Earliest Log`	Log file name, for example, `S0000000.LOG`
`Current Log`	Chain number the log file belongs to. An eight-character string starting with C and the chain number, for example, `C0000000`
`Comment`	Additional information about log file location, archiving options or tape location

Field	Contents
`Starttime`	Same as `Timestamp`
`Endtime`	14-digit timestamp that indicates when log file was archived
`Status`	Always A (active)
`Location`	Information about the log file location, depending on the archiving method
`EID`	Unique identifier for the entry in the history file

> **i Note**
>
> If you have specified two archiving methods, the history file will contain two entries per log file.

The following figure shows how the device type and the operation type are mapped to the locations of the log files:



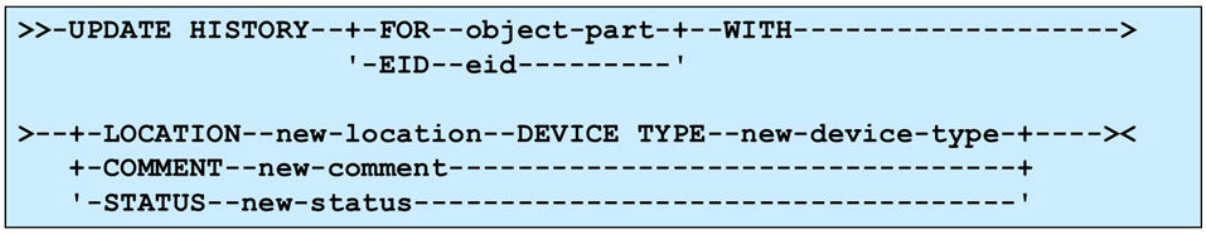
Mapping Locations of Log Files

Updating the History File

You update the history file using the **UPDATE HISTORY** command. This command can be used, for example, in the following situations:

- If you have moved log files or backups, you can update the location and device type.
- If a backup is no longer available, you can update the status to inactive. By doing so, you make sure that the recover command does not try to use the backup for database recovery.
  To update the status to *inactive*, enter the following command:
  db2 update history eid 10 with status "I"

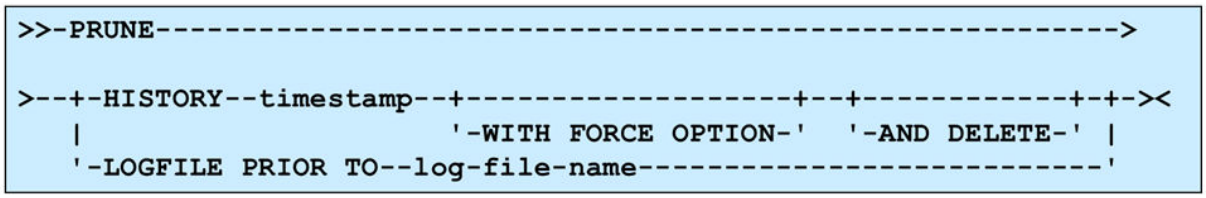
The syntax diagram for the UPDATE HISTORY command looks as follows:



Syntax of UPDATE HISTORY

Deleting Entries From the History File

To delete entries from the history file, use the **PRUNE HISTORY** command as shown in the following syntax diagram:



Syntax of PRUNE HISTORY

> ⚠ Caution
>
> - If you use `AND DELETE` or `LOGFILE PRIOR TO`, the log files are also deleted physically.
> - You **cannot** use this command to delete log files that are stored on tape.
>   If you have configured automatic file deletion in your storage management system, these deletions are not reflected in the history file. We therefore recommend that you do not delete log files in the storage management system directly. Use the `PRUNE HISTORY` command instead.

> ℹ Note
>
> The `PRUNE LOGFILE` command is deprecated as of Db2 10.5. Use the `PRUNE HISTORY` command instead.

8.2.7 Monitoring the Db2 Log Manager

Diagnostic Log

To get an overview of the operations that were performed by the Db2 log manager, you can use the Db2 tool `db2diag` by entering the following command:

```
db2diag -gi "EDUNAME:=db2logmgr"
```

A sample output looks as follows:

```
2013-01-17-04.21.21.509822+060 I55498016E548         LEVEL: Info
PID     : 12474                TID : 140737001809664 PROC : db2sysc 0
INSTANCE: db2dev               NODE : 000
HOSTNAME: db6xen027
EDUID   : 27                   EDUNAME: db2logmgr (DEV) 0
FUNCTION: DB2 UDB, data protection services, sqlpgArchiveLogFile, probe:3180
DATA #1 : <preformatted>
Completed archive for log file S0000349.LOG to /db2/DEV/log_archive/db2dev/DEV/
NODE0000/LOGSTREAM0000/C0000000/ from
/db2/DEV/log_dir/NODE0000/LOGSTREAM0000/.
```

For more information about the *db2diag* tool, see the IBM documentation for your Db2 version at https://www.ibm.com/docs/en/db2 .

Notification Log

The Db2 log manager writes entries to the notification log. On UNIX, the notification log is located in the `DIAGPATH` directory under the file name `<instance name>.nfy`. On Windows, the notification records are written to the Event Viewer in the *Application* section.

A notification entry looks as follows:

```
2013-05-14-09.03.45.533555   Instance:db2dev   Node:000
PID:23193(db2logmgr (DEV) 0)   TID:3808421632   Appid:none
data protection services  sqlpgArchiveLogFile Probe:3150
   ADM1848W  Failed to archive log file "S0000496.LOG" to
"/db2/DEV/log_archive/db2dev/DEV/NODE0000/LOGSTREAM0000/C0000000/" from
"/db2/DEV/log_dir/NODE0000/LOGSTREAM0000/".
```

You can retrieve information from the notification log using the table function `PD_GET_LOG_MSGS`.

8.2.8 Tape Support

The Db2 tape manager (`db2tapemgr`) archives log files to tape and retrieves them from tape.

> **i** Note
>
> For a database recovery, you must retrieve the log files from tape **before** the recovery procedure is started.

The Db2 tape manager updates the entries in the history file. In addition, the Db2 tape manager queries the required log files and their locations including the tapes that are required for a database recovery.

If you want to use the Db2 tape manager, consider the following:

- You have to **explicitly** call the Db2 tape manager to archive or retrieve log files to or from tape.
- Always use `db2tapemgr` as Db2 instance owner `db<dbsid>` because it needs access to the history file on operating system level.
- The Db2 tape manager supports multi-partition environments. However, you have to call the tool for each partition separately and you must execute it on the system where the database partition resides.
- The Db2 tape manager supports the same set of tape devices as the standard Db2 backup.

> **i** Note
>
> This chapter contains information about how log files are stored on tapes up to and including Db2 V9.7. Be aware that, as of Db2 10.1, the `LOGSTREAM` information is additionally stored.
>
> EXAMPLE:
>
> *Db2 V9.7: ...NODExxxx/Cyyyyyyyy/Szzzzzz.LOG*
>
> *Db2 10.1: ...NODEwwww/LOGSTREAMxxxx/Cyyyyyyyy/Szzzzzz.LOG*

8.2.8.1 Configuration of the Db2 Tape Manager

The following list describes configuration steps that you should perform before you call the Db2 tape manager:

- For tape support, you have to set the database configuration parameter `LOGARCHMETH1` to a disk location. To archive log files, the Db2 tape manager checks the history file for entries that are related to the parameter `LOGARCHMETH1`.
  Log files that are stored to disk using the `LOGARCHMETH2` parameter cannot be stored on tape using the Db2 tape manager.
- You should also set the Db2 registry variable `DB2_TAPEMGR_TAPE_EXPIRATION` to a feasible duration in days. This prevents that tapes that contain relatively new log files are accidentally overwritten. For example, if you perform daily backups and you want to keep log files for at least one week, you have to set the variable `DB2_TAPEMGR_TAPE_EXPIRATION` to 7.
- To reduce the amount of parameters that you have to specify for calls to the Db2 tape manager, you can set the database configuration variable `OVERFLOWLOGPATH` and the environment variable or registry variable `DB2DBDFT`.

8.2.8.2 Tape Labeling

To help you manage the tapes, the Db2 tape manager supports a simple tape labeling.

Each tape receives a tape label that was either automatically generated or supplied on the command line during archiving operations.

The automated tape label consists of the database name and the current time as a 14-digit time stamp.

> **Example**
>
> `<dbsid><YYYYMMDDHHMMSS>, sample PRD20040805121303.`

In addition, the tape label is written on the tape as part of the tape header file `DB2TAPEMGR.HEADER`.

The tape label can be up to 22 characters long, has to be alphanumeric (characters A - Z and digits from 0 - 9) and is not case-sensitive.

8.2.8.3 Physical Tape Layout

Each file that the Db2 tape manager writes to tape is encapsulated in a `cpio` archive.

The `cpio` archive file format is a file format well-known to UNIX administrators. `cpio` archive files can be read from and written to with the UNIX command `cpio`. The advantage of writing each file to its own `cpio` archive file is that fast tape positioning is possible.

Between each `cpio` archive, a file marker is created on the tape. This allows fast positioning on the tape. Using the `cpio` format for log files has the advantage that incompletely written files can be recognized and you can use standard UNIX tools to read the tape. The default block size that is used for writing the `cpio` archives to tape is 5120 bytes as with the **`cpio`** option **`-B`**.

Example Layout:

`DB2TAPEMGR.HEADER`
File marker
`NODE0000/C0000000/S0000010.LOG`
File marker
`NODE0000/C0000000/S0000011.LOG`
File marker
`NODE0000/C0000000/S0000012.LOG`
File marker
`NODE0000/db2rhist.asc` (the history file)
File marker
File marker (end of tape marker)

The following is an example content of the tape header file `DB2TAPEMGR.HEADER`:

```
label        :TAPE0
hostname     :PFERD
instance     :DB2PRD
database     :PRD
```

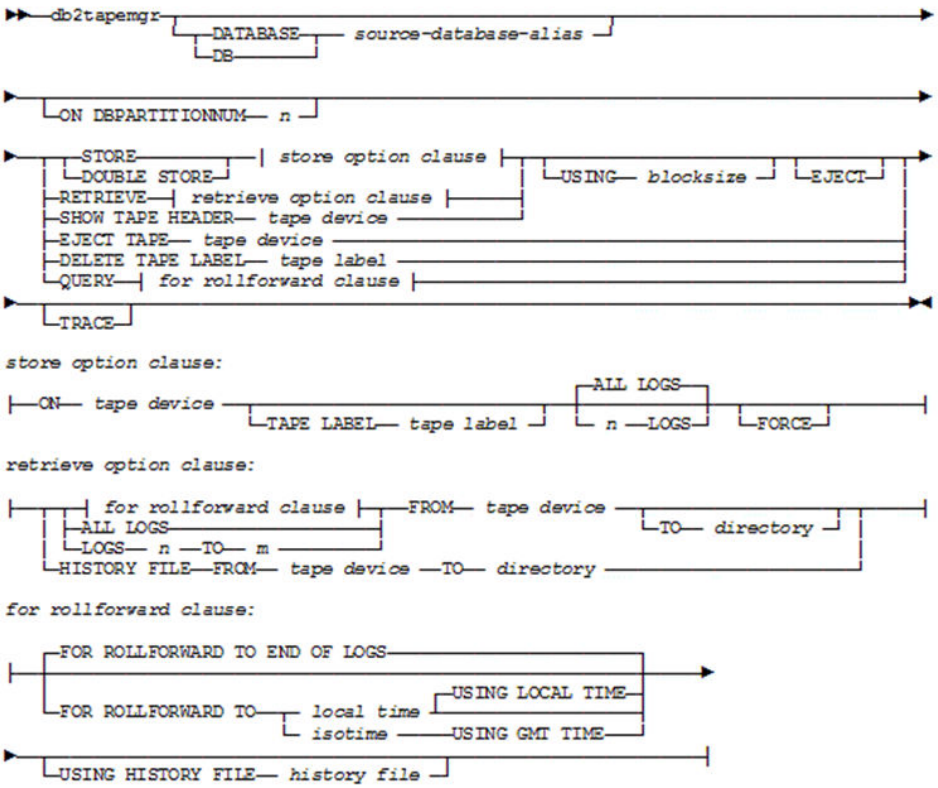
```
partition    :NODE0000
db version   :8.1.7.440
first used   :20040809183742
last modified:20040810173833
usage count  :18
contents     :
0    DB2TAPEMGR.HEADER
1    NODE0000\C0000000\S0000029.LOG
2    NODE0000\C0000000\S0000030.LOG
3    NODE0000\C0000000\S0000031.LOG
4    NODE0000\C0000000\S0000032.LOG
5    NODE0000\db2rhist.asc
```

The following table describes the content of the tape header file in detail:

Field	Value	Description
`label:`	`tape label`	Tape label that was specified at the store operation
`hostname:`	`host name`	Host name of the computer where the tape was created
`instance:`	`instance name`	Instance name
`database:`	`database name`	Database name
`partition:`	`partition number`	Partition where the tape was created
`db version:`	`database version`	Database version
`first used:`	`14 digit timestamp`	Time stamp when the Db2 tape manager wrote to this tape for the first time
`last modified:`	`14 digit timestamp`	Time stamp when the Db2 tape manager wrote to this tape for the last time
`usage count:`	`amount`	Indicates how often the Db2 tape manager has written to this tape
contents:	table containing: `<position on tape> <file name>`	Shows the tape content and allows fast access to special log files even if the history file does not contain information about this tape

8.2.8.4 Db2 Tape Manager Command Line Interface

The Db2 tape manager provides a large set of different options. The following is an overview of the command syntax:



Command syntax

Command Parameter

Parameter	Description
`DATABASE	DB source-database-alias`	Specifies the name of the database. If no value is specified, the value of `DB2DBDFT` will be used. If no value is specified and `DB2DBDFT` is not set, the operation fails.
`ON DBPARTITIONNUM n`	Specifies the database partition number to work on. If no value is specified, `DB2NODE` is used. If `DB2NODE` is not set, the default value is *0*.
`STORE ON tape-device`	Log files are archived to tape and deleted from disk afterwards.

Parameter	Description
`DOUBLE STORE ON tape-device`	All log files that were archived only once and the log files that have never been archived are stored on tape. Only the log files that were stored twice to tape are deleted; the others are kept on disk.
`ON tape-device`	Needs to be the name of a **non-rewind** tape device. The tape device names depend on the operating system platform: • **Windows**: `\\.\TAPE0` • **AIX**: `/dev/rmt0.1` • **HP-UX**: `/dev/rmt/0mn` • **Solaris**: `/dev/rmt/0n` • **Linux**: `/dev/nst0`
`TAPE LABEL tape-label`	Specifies a label to be applied to the tape. If `tape label` is not specified, the label will be generated automatically. For more information, see Tape Labeling [page 96].
`ALL LOGS` `n LOGS`	Specifies that the command applies to all log files or a specified number of log files.
`FORCE`	If the Db2 tape manager rejects writing to tape (because checks to avoid accidental overwrites of tapes failed), you can use the `FORCE` option to overwrite these checks.
`USING blocksize`	Specifies the block size for tape access. The default size is `5120`, and it must be a multiple of `512`. The minimum is `512`.
`EJECT`	Specifies that the tape is to be ejected after the operation completes.
`RETRIEVE [for rollforward clause]`	With this option, the Db2 tape manager determines in the history file which tapes are required for a database recovery. You will be asked to insert the required tapes.
`RETRIEVE (ALL LOGS | LOGS n to m)`	With this option, the Db2 tape manager does not access the information in the history file. You can retrieve all or only some logs with this option even if you do not have an up-to-date history file.
`RETRIEVE HISTORY FILE`	With this option, you can retrieve the history file that was archived during the `STORE` operation from the tape.

Parameter	Description
`QUERY [for rollforward clause]`	Displays the backup image that will be used for a database recovery, the log file locations of the required log files for the database recovery, and the required log files for the database recovery.
`USING HISTORY FILE history file`	Specifies an alternative history file to be used.
`SHOW TAPE HEADER tape-device`	Shows the content of the tape header file `DB2TAPEMGR.HEADER`.
`EJECT TAPE tape-device`	Ejects the tape in the specified tape device.
`DELETE TAPE LABEL tape-label`	Deletes all locations from the history file that refer to the specified tape label.
`TRACE`	We recommend that you add this option for support purposes in case of an error. This will produce detailed output that helps the support team to analyze the problem.

8.2.8.5 Archiving Log Files to Tape Using STORE and DOUBLE STORE

The Db2 tape manager supports the following two archiving operations when archiving log files to tape:

Archiving Operation	Description
`STORE`	The `STORE` operation copies the log file to tape and then deletes the log files from disk. Example `db2tapemgr DB PRD STORE ON \\.\TAPE0`
`DOUBLE STORE`	The `DOUBLE STORE` operation copies log files to tape and then deletes only those log files that were copied to tape a second time. Since tapes are quite unreliable (for example, when used too often they cannot be read anymore), we recommend that you use the `DOUBLE STORE` operation to avoid data loss. Example `db2tapemgr DB PRD DOUBLE STORE ON \\.\TAPE0`

Archiving Process

The Db2 tape manager performs the following steps during the archiving operation:

1. If no log files were found, the tape manager stops the operation with the following message: `“DBT2016I No log files found for processing.”`.
2. Reads the tape header file `DB2TAPEMGR.HEADER`
3. Validates if writing to tape is acceptable
   For more information, see Security Features [page 106].
4. If the tape already contains log files, the history file is updated. Any entries in the history file for log files that are related to the tape you are writing to are marked with a minus sign ‘-‘ in the *Location* field.
5. Writes a new tape header file `DB2TAPEMGR.HEADER` to the tape
6. Copies the log files to tape
   If the log files do not fit on the tape, the Db2 tape manager automatically reduces the number of log files to be stored on tape and starts writing to the tape again from the beginning.

   > → Recommendation
   >
   > To avoid this time-consuming operation, we recommend that you limit the number of files written during an archiving operation by using the `n LOGS` option.

7. Updates the history file
   The entries in the history file for the log files show the location on the tape.
8. Scans the history file for log file entries with `Type equals 1` (`LOGARCHMETH1`) and `Dev equals D` (disk location); you receive a warning message for every file that is supposed to be on disk but not found.
   - For a `STORE` operation, the history file entries are updated as follows:
     The device type is changed to `T` and the `Location` field is updated to contain the position on the tape in the following format:
     `<tape label>:<pos>:<relative path>`
   - For a `DOUBLE STORE` operation the history file entries are updated as follows:
     - If the log file has not been stored to a tape before, the `Comment` field in the history file is updated to contain the position on the tape in the following format:
       `<tape label>:<pos>`
     - If the log file has already been stored to another tape, the device type is updated to `T` and the `Location` field is updated to contain the position on the tape in the following format:
       `<tape label>:<pos>:<relative path>`

   The Db2 tape manager also updates the history field end time. The end time field is set to the time when the tape header file was created.
9. Deletes log files from disk
   - `STORE` operation: Deletes all archived log files
   - `DOUBLE STORE`: Deletes only those log files that were stored twice now
10. Writes the history file to the tape
    If the history file does not fit on the tape, a warning message is displayed. You still have to keep the tape.
11. If you have specified the `EJECT` option, the tape is ejected from the tape drive.

The following table describes the different log file history entries that you might find in the history file when using the Db2 tape manager:

Device Type	Location	Comment	Description
D	<path to log file on disk>		The log file is still located on disk. The Db2 tape manager has not been called for that log file so far.
D	<path to log file on disk>	<tape-label>:<pos>	The log file is still located on disk, but has been stored to tape with a DOUBLE STORE operation.
T	<tape-label>:<pos>:NODExxxx/ Cyyyyyyyy/Szzzzzz.LOG		The log file is only located on tape.
T	<tape-label>:<pos>:NODExxxx/ Cyyyyyyyy/Szzzzzz.LOG	<tape-label>:<pos>	The log file is located on two tapes, after the second DOUBLE STORE operation for this log file.

A minus sign ("-") in the *Location* or *Comment* field indicates that either a DELETE TAPE LABEL operation was performed or a tape that contains log files was overwritten with new log files.

8.2.8.6 Retrieving Log Files from Tape

You can easily retrieve log files from tape by using the RETRIEVE FOR ROLLFORWARD TO option. You are asked to insert any required tape for a database recovery.

The following example output describes the retrieve operation of the Db2 tape manager:

```
db2tapemgr DB PRD retrieve for rollforward to end of logs from \\.\TAPE0
DBT2065I  Using database partition "NODE0000".
Scanning history.
Scanning history.
Required tapes ":".
TESTTAPE
Insert tape "TESTTAPE".
Press '9' to quit or any other key to continue.
Rewinding tape.
Reading tape header.
Retrieving log files from tape "TESTTAPE".
Reading log file "NODE0000\C0000000\S0000000.LOG" from tape.
Reading log file "NODE0000\C0000000\S0000001.LOG" from tape.
Reading log file "NODE0000\C0000000\S0000002.LOG" from tape.
Reading log file "NODE0000\C0000000\S0000003.LOG" from tape.
Reading log file "NODE0000\C0000000\S0000004.LOG" from tape.
Positioning tape.
Reading log file "NODE0000\C0000001\S0000005.LOG" from tape.
Reading log file "NODE0000\C0000001\S0000006.LOG" from tape.
Positioning tape.
```

```
Reading log file "NODE0000\C0000003\S0000007.LOG" from tape.
Reading log file "NODE0000\C0000003\S0000008.LOG" from tape.
Reading log file "NODE0000\C0000003\S0000009.LOG" from tape.
DBT2006I  db2tapemgr completed successfully.
```

If you do not have a current version of the history file and if you know on which tape you find the required log files, you can use the following options to retrieve log files from tape:

- `RETRIEVE ALL LOGS`
- `RETRIEVE LOGS n TO m`

If you need to start a disaster recovery and if you do not know where the log files are, you can use the Db2 tape manager to retrieve the history file from the tape that was stored at the end of the tape.

> **Note**
>
> Always use the latest tape and the following command to retrieve the history file from tape:
>
> **`db2tapemgr RETRIEVE HISTORY FILE FROM \\.\TAPE0 TO c:\temp`**

With this history file, you can use the `QUERY … USING HISTORY FILE <hist-file>` option to **find** the required tapes and the `RETRIEVE … USING HISTORY FILE <hist-file>` to **retrieve** the required log files.

> **Example**
>
> To find the required tapes, enter the following command:
>
> **`db2tapemgr QUERY USING HISTORY FILE c:\temp\NODE0000\db2rhist.asc`**
>
> To retrieve the required log files, enter the following command:
>
> **`db2tapemgr RETRIEVE USING HISTORY FILE c:\temp\NODE0000\db2rhist.asc`**

If you do not specify a destination path using the `TO <directory>` option, the log files are retrieved to the overflow log path directory, which can be set in the database configuration.

The log files are created in a hierarchical manner, for example:

`<dir>/NODExxxx/Cyyyyyyy/Szzzzzzz.LOG`

If the destination directory already contains the node directory `NODExxxx`, the log files are restored as follows:

`<dir>/Cyyyyyyy/Szzzzzzz.LOG`

> **Note**
>
> Since the recover command searches the log files by default in the overflow log path directory, you should set the `OVERFLOWLOGPATH` parameter in the database configuration. This simplifies calls to the Db2 tape manager and the call for the recover command.

To avoid duplicate retrieval of log files from tape, the Db2 tape manager does not retrieve log files that it has already found in the destination path. In this case, a warning message is displayed.

8.2.8.7 Other Operations

You can also perform the following operations using the Db2 tape manager:

- `EJECT TAPE`
- `SHOW TAPE HEADER`
- `DELETE TAPE LABEL`

EJECT TAPE

If you want to eject the tape from the tape drive, enter the following command:

```
db2tapemgr EJECT TAPE /dev/rmt0.1
```

SHOW TAPE HEADER

This option displays the contents of the tape header file. You can use this information to check:

- Which tape has been inserted in the tape drive
- How often the tape has been used to archive log files
- Which log files are on the tape in case you lost the history file

The following is an example output when the `SHOW TAPE HEADER` operation is performed:

```
db2tapemgr SHOW TAPE HEADER \\.\TAPE0
DBT2062I  Working on database "PRD".
DBT2065I  Using database partition "NODE0000".
Rewinding tape.
Reading tape header.
Tape header contents
label        :TAPE0
hostname     :PFERD
instance     :DB2PRD
database     :PRD
partition    :NODE0000
db version   :8.1.7.440
first used   :20040809183742
last modified:20040810173833
usage count  :18
contents     :
0    DB2TAPEMGR.HEADER
1    NODE0000\C0000000\S0000029.LOG
2    NODE0000\C0000000\S0000030.LOG
3    NODE0000\C0000000\S0000031.LOG
4    NODE0000\C0000000\S0000032.LOG
5    NODE0000\db2rhist.asc
DBT2006I  db2tapemgr completed successfully.
```

DELETE TAPE LABEL

This option removes the location information from the log file entries in the history file that are related to the specified tape. If you have lost a tape or a tape is corrupt, you should use this option because this should be reflected in the history file.

To delete tape labels, enter the following command:

```
db2tapemgr DELETE TAPE LABEL TAPE0
```

8.2.8.8 Security Features

To prevent log files from being overwritten on tape, the Db2 tape manager performs the following security checks:

- It checks if the tape has been used for archiving log files before by verifying that a `DB2TAPEMGR.HEADER` is on the tape. This prevents that tapes are overwritten with other content (for example, backup tapes). The other way round is not safe. For example, the Db2 backup command would overwrite tapes containing log files.
- The tape header content is checked to make sure that:
  - The tape content has not expired
    To achieve this, the Db2 tape manager calculates the difference between the last modified field from the tape header file and the current time and compares the difference with the value that was specified by the Db2 registry variable `DB2_TAPEMGR_TAPE_EXPIRATION`.
  - Log files are not overwritten by log files of another instance, database, or partition.
  - The same tape label is not used for different tapes.

> **i Note**
>
> You can use the `FORCE` option to override these checks.

Before log files are retrieved from tape, similar checks are performed. The Db2 tape manager checks if the tape header information about instance, database, and partition are correct. If discrepancies are found, the Db2 tape manager asks you if you want to proceed with the log file retrieval.

In some cases, for example, if you want to retrieve log files for a rollforward after a redirected restore, the check will fail, but you need to continue.

8.2.8.9 Troubleshooting

This section provides information about how to proceed if problems occur with the Db2 tape manager.

> **i Note**
>
> Be aware that the information does not claim to be complete, but it might help you in some cases to resolve problems.

Tracing

To retrieve detailed information about the cause of a problem, you have to create a trace file of a Db2 tape manager run. This trace file is essential for support purposes.

By adding the TRACE option to the **db2tapemgr** command, trace information is written to the standard output and you can redirect it to a file.

> ### Example
>
> The following command is an example of how to create a trace file:
>
> ```
> db2tapemgr DB PRD STORE ON \\.\TAPE0 TRACE > db2tapemgr.trc
> ```

You can also use the standard Db2 tracing function as follows:

1. To activate the Db2 tracing function, enter the following command:
   ```
   db2trc on –f <filename>
   ```
   > ### Note
   >
   > To restrict the impact on system performance, you can limit the traced Db2 components when you activate the trace. The component number of the Db2 tape manager is 143. For example, enter the following command:
   >
   > ```
   > db2trc on –m “*.*.143.*.*” –f <filename>
   > ```
2. Run the **db2tapemgr** command that failed, for example:
   ```
   db2tapemgr DB PRD STORE ON \\.\TAPE0
   ```
3. To deactivate the Db2 tracing function, enter the following command:
   ```
   db2trc off
   ```

Problems with the Tape Drive

Problem Description	Solution
On some Linux distributions, such as SuSE, not all users are allowed to read and write to tape devices.	Make sure that the user who calls the Db2 tape manager has sufficient authorizations to perform tape operations.
The tape drive that you are using does not support the default block size of 5120 bytes.	Specify the block size using the `BLOCKSIZE` parameter.
You have problems accessing the tape with the Db2 tape manager.	To check if the tape drive is working correctly, use the UNIX utilities `dd`, `cpio`, and `mt`.

Tape Loss

If you lost a tape or a tape became unreadable, use the `DELETE TAPE LABEL` option of the Db2 tape manager to remove the history entries of the log files from the history file to reflect that these log files do not exist on

the tape anymore. In addition, you should check if your database is still recoverable. If not, perform a database backup immediately.

Inconsistencies of the History File

If the history file is corrupt or not up-to-date and you want to use the interactive retrieval of log files using the Db2 tape manager, proceed as follows:

1. Retrieve the history file from the latest tape. For example, enter the following command:
   ```
   db2tapemgr RETRIEVE HISTORY FILE FROM /dev/rmt0.1 TO /tmp
   ```
2. Query the required tapes for a database rollforward, using, for example, the following command:
   ```
   db2tapegmr QUERY USING HISTORY FILE /tmp/NODE0000/db2rhist.asc
   ```
3. Retrieve the required log files from tapes. For example, enter the following command:
   ```
   db2tapemgr RETRIEVE USING HISTORY FILE
   /tmp/NODE0000/db2rhist.asc FROM /dev/rmt0.1
   ```

If you know on which tape the required log files are located, you can also use the Db2 tape manager options `RETRIEVE LOGS n TO m` or `RETRIEVE ALL LOGS` to retrieve the log files from tape.

Log Files to Be Archived Not Found

If you loose the history file, for example, due to a disk failure, but there are still log files on the disk that need to be archived to tape, you **cannot** use the Db2 tape manager to archive these log files.

The Db2 tape manager **cannot find** these log files because the history files do not contain any information about them. The only way to archive these log files to tape is to use the UNIX utilities `cpio` or `tar`. However, you have to remember later on that you archived these log files using UNIX tools and not using the Db2 tape manager.

Tape Header File Unreadable

If the tape header file cannot be read, the Db2 tape manager is not able to retrieve log files from tape. Therefore, use the UNIX utility `cpio` to retrieve log files from tape, for example, using the following command:

```
dd if=/dev/rmt0.1 bs=5120 | cpio –iduvB
```

> **i** Note
>
> As the `cpio`, `dd` utilities and `cpio` commands are **not** available on Windows, this was tested with the Cygwin toolset for Windows. You can download the toolset at http://www.cygwin.com .

Error Messages

The following table describes error messages that are not self-explaining and might be displayed when using the Db2 tape manager:

Error Message	Reason	Solution
`Scanning history failed. Reason:` `"SQLO_NLCK_PATH_ERROR: Unknown".`	The user who has called the Db2 tape manager does not have access to the history file `db2rhist.asc`.	Start the Db2 tape manager as instance owner.
`Reading tape header failed.` `Reason: "1106: When accessing a new tape of a multivolume partition, the current block size is incorrect.“`	The tape header cannot be read from tape because you use a new tape on Windows.	Use the `STORE` or `DOUBLE STORE` operation with the `FORCE` option to write to the tape for the first time.

8.3 Database Backup

You must perform backups on a regular basis to be able to restore the database to a consistent state that is as up-to-date as possible.

You can perform backups in online or offline mode:

- **Online** mode
  Access to the database is not blocked. Users can continue to work normally during the backup.
- **Offline** mode
  The backup process connects to the database in exclusive mode. The database can be restored without log files.

In either case, you can completely restore the database and bring it up-to-date by rolling in the log files generated after the backup was taken.

> **i** Note
>
> Database backups are compatible between Fix Pack levels of the same Db2 version. However, you cannot restore a database backup from a higher Db2 version to a lower Db2 version.
>
> With Db2 11.1, the concept of Modification Packs was introduced. A Modification Pack (also referred to as Mod Pack, or simply Mod or MP) introduces new functions to the Db2 product. Since new functions from a higher Mod Pack are not available on lower Mod Pack levels, you cannot restore a database backup on a lower Mod Pack level if it was taken on a higher Mod Pack level.

To perform both online and offline backups of the database, you can use the **BACKUP DATABASE** command. For online backups, you can also use the DBA Planning Calendar in the DBA Cockpit.

> **i** Note
>
> Offline backups are not supported by the DBA Cockpit because the backup commands are issued using an SQL interface (`admin_cmd`) that only allows online backups.

Related Information

Backup Requirements [page 110]
Performing a Database Backup [page 111]
Integrity of Backups [page 111]
Frequency of Backups and Required Time [page 112]
Deletion of Backup Images [page 112]
Advanced Backup Techniques [page 113]

8.3.1 Backup Requirements

> **⚠** Caution
>
> You can only back up a database that is in a usable state or in backup pending mode.

The following is necessary to perform a backup:

- You must have `SYSADM`, `SYSCTRL`, or `SYSMAINT` authorization to use the **BACKUP DATABASE** command.
- You must start the database manager (`db2start`) before taking a database backup.
- In a partitioned database system, keep a copy of the `db2nodes.cfg` file with any backup copies you take. This copy is as a protection against possible damage to this file.
- Use the database tool `db2cfexp` to back up the database manager configuration and the Db2 registry. To do so, enter the following on the command line:
  ```
  db2cfexp <file_name> BACKUP
  ```

The database may be local or remote. The backup image remains on the database server unless a storage management product, such as Tivoli Storage Manager (TSM), is used.

You can back up a database to a specified disk, a tape, or a location managed by TSM or another vendor storage management product.

After an online backup, Db2 forces the currently active log file to be closed and, as a result, it will be archived. This ensures that an online backup has a complete set of archived log files available for recovery.

8.3.2 Performing a Database Backup

You can back up a single-partition or multi-partition database using either the Db2 command line or the DBA Cockpit.

Performing a Backup on the Db2 Command Line

Single-Partition Database

1. Log on to the database server as user `db2<dbsid>`.
2. Enter the following command:
   ```
   db2 backup db <DBSID>...
   ```
   For the complete syntax of this command, check the Db2 documentation.

Multi-Partition Database

1. Log on to the database server as user `db2<dbsid>`.
2. Enter the backup command on the catalog partition:
   ```
   db2 "backup db <DBSID> ON ALL DBPARTITIONNUMS ..."
   ```

Backing Up the Database Using the DBA Cockpit

1. In your SAP system, call transaction `DBACOCKPIT` and choose *Jobs* > *DBA Planning Calendar*.
2. Click a calendar cell. Alternatively, choose the *Schedule an Action* link in the *Favorites* area of the Web browser-based version of the DBA Cockpit.
   The jobs that you can schedule depend on the storage device that you are using. You can also specify if you want to back up a single-partition database or a multi partition database.

For more information, see the DBA Cockpit documentation that you can find under References [page 228] in the appendix .

8.3.3 Integrity of Backups

The `db2ckbkp` utility allows you to test the integrity of a backup image on disk or tape and to determine whether or not it can be restored. It can also be used to display the metadata stored in the backup header to get information about a particular backup image. One or more parts of an image can be checked. You can use the utility as follows:

If the full backup image consists of multiple objects, the validation will only succeed if `db2ckbkp` is used to validate all of the objects at the same time.

When checking multiple parts of an image, the first backup image object (`.001`) must be specified first.

> Example
>
> ```
> db2ckbkp PRD.0.db2prd.DBPART000.20121015173538.
> ```

> **Note**
>
> The naming convention for backup image files as shown here has been introduced with Db2 10.1. If you use Db2 versions older than 10.1, you will see image names with a different structure.

If the backup resides on TSM, you can use the Db2 tool `db2adutl` to check the integrity of backups.

For more information about the use of `db2ckbkp` and `db2adutl`, see the IBM documentation for your Db2 version at https://www.ibm.com/docs/en/db2 .

8.3.4 Frequency of Backups and Required Time

You should perform full database backups regularly, regardless of how often log files are archived. A recent full backup means that there are fewer archived log files to apply to the database in case of a recovery, which reduces the amount of time required by the `ROLLFORWARD` utility to recover the database. It also reduces the chance of a log file not being available (corruption or loss).

To reduce the amount of time that the database is not available, consider performing online backups.

> **Note**
>
> To recover a database from an online backup, you need to have at least the following log files available:
>
> All log files that were still active when the backup was started and all log files that were created during the online backup.

8.3.5 Deletion of Backup Images

From time to time, you may want to delete old backup images to save space on your storage media. How you delete backup images usually depends on the backup location.

- Deleting backup images in TSM
  Since the backup images are stored in a backup copy group in TSM, they are not automatically deleted. To manually delete the backup images in TSM, you can use the Db2 tool `db2adutl` with the `DELETE FULL` option.
  For more information about how to use `db2adutl` in a TSM environment, see the IBM Db2 documentation.
- Managing Db2 snapshot backup objects
  Db2 has an integrated snapshot backup function, that is, a snapshot backup operation uses the fast copying technology of a storage device to perform the data copying during the backup.
  This integrated snapshot backup function includes the `db2acsutil` tool that lets you manage the snapshot backup objects.
  For more information about the `db2acsutil` tool, see the IBM Db2 documentation.
- Using automatic log file and backup retention
  For more information, see Deleting Archived Log Files [page 90].

More Information

IBM Db2 documentation at https://www.ibm.com/docs/en/db2

8.3.6 Advanced Backup Techniques

Incremental or delta backups

To reduce the backup and restore time, you can use incremental or delta backups. For more information, see High availability and Data recovery in the IBM documentation.

File system-based backups

The database needs to be either offline or in write suspend mode when taking backups on file system level. Typically, such file system backups are most useful if you have a storage device that allows for taking very fast file system copies or snapshots. After you have taken a file system backup, you can either use it for a later recovery or you can initialize the file system copy and take a Db2 backup from it.

For more information, see The db2inidb Tool [page 116].

8.3.7 Monitoring Database Backups

You can get an overview of the database backups using the DBA Cockpit.

In the DBA Cockpit, you can use the following functions to monitor database backups:

- Get a list of backups that ran during a specified time range, including information about the backup type, backup size, backup runtime, and whether the backup finished successfully or failed
- Display information about log files that have been moved from the log directory to the log archive or to a storage product

For more information about the DBA Cockpit, see the References [page 228] in the appendix.

In addition, you can also display detailed information about backup resource usage: Investigate the diagnostics logs, for example, using the DBA Cockpit or on operating system level, and look for entries of the component database utilities. You can investigate a diagnostic log entry containing the message `Performance statistics`. For more information about how to interpret the backup and restore statistical information, see the IBM Db2 documentation Example output for backup performance monitoring .

8.4 Database Recovery

8.4.1 Database Recovery Using the RECOVER Command

> **i** Note
>
> Database backups are compatible between Fix Pack levels of the same Db2 version. However, you cannot restore a database backup from a higher Db2 version to a lower Db2 version.
>
> With Db2 11.1, the concept of Modification Packs was introduced. A Modification Pack (also referred to as Mod Pack, or simply Mod or MP) introduces new functions to the Db2 product. Since new functions from a higher Mod Pack are not available on lower Mod Pack levels, you cannot restore a database backup on a lower Mod Pack level if it was taken on a higher Mod Pack level.

The `RECOVER` command automatically selects a suitable backup image and the log files required to recover the database to a specific point in time or to the end of logs. The `RECOVER` command performs the complete restore and the rollforward operation. This also applies in a multi-partition environment with one command call.

The `RECOVER` command is dependent on access to an up-to-date history file because all information about backup images and log files is retrieved from there. If the history file is lost, you can restore it from the latest backup using the `RESTORE ... HISTORY FILE` command. A history file that was restored from a database backup, however, only contains entries up to the backup that it was part of. For access to log files after this backup, the `RECOVER` command uses the search order as listed below.

> **i** Note
>
> Db2 includes a backup copy of the history file in every database backup. But there is no direct Db2 support available if you want to take more frequent backups of your history file. As a workaround, you can create an empty tablespace and back up this tablespace on a regular basis to have regular backups of the history file with a small overhead.
>
> If you archive log files to tape using the Db2 tape manager, you will find a relatively new version of the history file on the latest tape.

The database recovery with the `RECOVER` command is performed in two phases.

In the **first phase**, the `RECOVER` command selects a suitable backup from the history file and restores the backup image. This is done for all partitions.

In the **second phase**, the `RECOVER` command performs the database rollforward to the specified point-in-time or to the end of logs. The `RECOVER` command searches for the required log files in the following search order:

1. Log directory or mirror log directory
2. `<OVERFLOWLOGPATH>/Szzzzzzzz.LOG`
3. `<OVERFLOWLOGPATH>/NODEwwww/LOGSTREAMxxxx/Cyyyyyyy/Szzzzzzz.LOG`
4. Location that is stored in the history file
5. Location specified by `LOGARCHMETH1` or `LOGARCHMETH2`

> **i Note**
>
> The search order above is valid for the `RECOVER` command only. If you use the `ROLLFORWARD` command, the log file from the latest chain (the newest log file with the same number) is always used to roll forward the database.

The database administrator `db2<dbsid>` or the SAP system administrator `<sapsid>adm` can recover the database as follows:

- By performing a recovery to the end of logs:
  1. Log on to the db server as `db2<dbsid>` or `<sapsid>adm`.
  2. Enter the following command:
     ```
     db2 RECOVER DB <DBSID>
     ```
- By performing a recovery to a point in time where the time must be the local time:
  1. Log on to the database server as `db2<dbsid>` or `<sapsid>adm`.
  2. Enter the following command:
     ```
     db2 RECOVER DB <DBSID> TO <local time>
     ```

More Information

- Database Recovery Using the RESTORE and ROLLFORWARD Command [page 115]
- High availability in the IBM documentation
- Data recovery in the IBM documentation
- Database reference in the IBM documentation (use the table of contents on the webpage to browse the various topics)

8.4.2 Database Recovery Using the RESTORE and ROLLFORWARD Command

If the history file is lost and you cannot restore an up-to-date version of it, you must use the `RESTORE` and `ROLLFORWARD` commands instead of the Db2 `RECOVER` command to recover your database step by step.

> **⚠ Caution**
>
> Be aware that this procedure requires Db2 expert knowledge.
>
> Make sure that you specify the point-in-time value in the correct time zone format: The `RECOVER` command uses the local time whereas the `ROLLFORWARD` command by default uses the coordinated universal time (UTC) standard.

> **i Note**
>
> Database backups are compatible between Fix Pack levels of the same Db2 version. However, you cannot restore a database backup from a higher Db2 version to a lower Db2 version.
>
> With Db2 11.1, the concept of Modification Packs was introduced. A Modification Pack (also referred to as Mod Pack, or simply Mod or MP) introduces new functions to the Db2 product. Since new functions from

a higher Mod Pack are not available on lower Mod Pack levels, you cannot restore a database backup on a lower Mod Pack level if it was taken on a higher Mod Pack level.

More Information

- High availability in the IBM Db2 documentation
- Data recovery in the IBM Db2 documentation
- Section *Commands* under Database reference in the IBM Db2 documentation

8.5 File System Backups and db2inidb Tool

File system backups, the Db2 tool `db2inidb`, and the `write suspend` feature in combination with hardware technologies like `EMC Timefinder®` or `IBM ESS®` provide the basis for fast backups and database clone creation.

With these technologies, you can split up an entire file system very fast, that is, create a split image of a file system. This split image can be mounted on a different machine, in a different directory on the same machine where the original file system is located, or replace the current content of the original file system in case of a failure. `EMC Timefinder®` and `IBM ESS®` also provide functions to archive a split image to storage devices, for example, to tape.

To take a file system backup, you have to either shut down the database or suspend write operations using the `write suspend` command before you create a split image. If you create a split image while the database is still writing to disk, you obtain a corrupt image that cannot be restored. For more information, see Performing a File System Backup [page 117].

After you have taken a file system backup, you can use the `db2inidb` tool to initialize the backup image for the following use cases:

- **Fast database backups and restores**
  The `db2inidb` option `as mirror` is used to create fast database backups and restores with nearly no system outage, for example, when a database has become corrupt and needs to be restored as fast as possible. To restore your database, you have to mount a split image that you created earlier and roll forward the database with the existing log files. Mounting the split image is faster than a regular database restore and saves time compared to performing a regular database restore from a regular Db2 backup image.
  For more information, see Using a File System Backup for Recovery [page 119].
- **Creating database clones for quality assurance**
  To create database clones for quality assurance or test systems, you can use the `db2inidb` option `as snapshot`. In the past, you had to copy an SAP system using the homogeneous system copy. The homogeneous system copy requires a full database backup and a redirected restore on the target system. This procedure can be time-consuming if your source system is very large. The `db2inidb` option `as snapshot` allows you to perform a homogeneous system copy for large databases very fast.
  For more information, see Using a File System Backup for Database Cloning [page 120].

- **Creating a hot-standby database**
  To synchronize two databases using log files, that is, to create a hot-standby database, you use the `db2inidb` option `as standby`.
  Hot-standby database systems are used to:
  - Avoid long restore operations after a hardware failure
    The hot-standby database buffers are allocated and filled with the data pages that were changed most recently.
  - Allow a fast recovery from logical failures
    In this case, the hot-standby system is kept in a state with a defined time delay in comparison to the source system. If a logical error is detected on the source system, you can switch to the state of the hot-standby system where the logical error is not yet applied.

  For more information, see Using a File System Backup to Set Up a Hot-Standby Database [page 121].
- **Creating Db2 backups with minimal impact**
  To create normal Db2 backups with nearly no system outage, you can also use the `db2inidb` option `as standby`.
  You can create a Db2 backup image from a file system backup using standard Db2 commands. In this way, you create an online backup from a split image without influencing the performance on the production system. You can use Db2 backups, for example, to restore the database on a different hardware platform, to perform a redirected restore, or to restore the database into a newer Db2 version, for example, to build up a quality assurance or test system on a new database version.
  For more information, see Using a File System Backup to Create a Db2 Backup Image [page 122].

> ⚠ Caution
>
> The procedures provided in the following sections are **only** intended for **experienced** Db2 database administrators.

For more information about the `db2inidb` tool, go to the IBM Db2 documentation for your database version at https://www.ibm.com/docs/en/db2 .

8.5.1 Performing a File System Backup

To take a file system backup, perform the following steps on the source system:

Procedure

1. To determine the file systems to be included in the split image, enter the following statement using the DBPATHS administrative view:
   ```
   db2 "select DBPARTITIONNUM, substr(TYPE,1,30) as TYPE, substr(PATH,1,80) as PATH
   from SYSIBMADM.DBPATHS"
   ```
2. Switch the database to write suspend mode by logging on to the database server as user `db2<dbsid>` and entering the following command for **all** database partitions:
   ```
   db2 set write suspend for database
   ```
   The database suspends all write operations.

Optionally, to reduce the impact on the workload during write suspend mode, you can use this command with the exclude logs option as follows:
`db2 set write suspend for database exclude logs`
In this case, write operations to log files continue instead of quickly filling up the much smaller log buffer. The database can stay longer in write suspend mode and more running transactions can complete.
Do **not** use the exclude logs option if you also want to mirror the log files that are necessary, for example, for SAP system clones to create quality assurance and test systems.

3. Create the split image using the paths obtained in step 1.
   The split image must include:
   - All database containers (`sapdata*`)
   - The database instance directory:
     UNIX: `/db2/<DBSID>/db2<dbsid>`
     Windows: `<drive>:\db2\<DBSID>\db2<dbsid>`

   > ⚠ Caution
   >
   > Make sure that the image does **not** include the log directory, the archive directory, and the retrieve directory. This is important in order to avoid that the current log files, which are contained in these directories, are overwritten with the old log files contained in the split image when the split image is mounted for a database restore.

4. To switch the database mode back to normal operation, enter the following command:
   `db2 set write resume for database`

   > ⚠ Caution
   >
   > Make sure that you issue the **`set write resume`** command from the same connection as the **`set write suspend`** command. Otherwise, the resume operation might fail.

   The database now allows full access again.
5. Archive the split image.

   > ℹ Note
   >
   > After you have created the split image, you can:
   > - Compress the split image before archiving it to tape. This saves you tape space and I/O bandwidth.
   > - Archive to any storage management system. In this way, you are not limited to the destinations provided by the Db2 **`backup`** command.

   > ⚠ Caution
   >
   > With Db2, archiving and restoring the database files and database containers is **only** allowed in connection with the `write suspend` feature and `db2inidb` tool for the regular Db2 backup and restore function.
   >
   > If you do not use these tools, you might cause irrevocable data loss or unexpected system behavior.

More Information

File System Backup and db2inidb Tool [page 116]

Using a File System Backup for Recovery [page 119]

Using a File System Backup for Database Cloning [page 120]

Using a File System Backup to Set Up a Hot-Standby Database [page 121]

Using a File System Backup to Create a Db2 Backup Image [page 122]

8.5.2 Using a File System Backup for Recovery

Prerequisites

You have created the split image as described in Performing a File System Backup [page 117].

Procedure

1. Mount the split image, which you created earlier, on your database server.
2. Log on to the database server as user `db2<dbsid>`.
3. To initialize the split image using the storage management system tools, enter the following command:
   ```
   db2inidb <DBSID> as mirror
   ```
   The database is now in *rollforward pending* mode.
4. Perform a rollforward recovery to the end of logs as follows:
   1. Start the rollforward recovery using the following command:
      ```
      db2 rollforward database <DBSID> to end of logs
      ```
   2. Check if the rollforward recovery is complete using the following command:
      ```
      db2 rollforward database <DBSID> query status
      ```
      If the rollforward recovery is not complete, you have to correct the problem, for example, by providing missing log files and repeating step 1.
   3. Complete the rollforward recovery using the following command:
      ```
      db2 rollforward database <DBSID> stop
      ```
      If the operation was successful, the database should now be in the most current state and can be accessed again.

The function to create backups using snapshot backup technology is fully integrated into the regular backup and restore commands. For more information about the USE SNAPSHOT option of the BACKUP DATABASE and the RESTORE DATABASE commands, see the IBM documentation for your Db2 version at https://www.ibm.com/docs/en/db2 .

8.5.3 Using a File System Backup for Database Cloning

Prerequisites

You have created the split image as described in Performing a File System Backup [page 117].

Procedure

1. On the target system, prepare a standard SAP system environment for SAP databases using homogeneous system copy methods.
2. Mount the split image using your storage management system.
3. To initialize the database, enter the following command:
   `db2inidb <DBSID> as snapshot`
   Alternatively, if you want to change the container layout on the target system to distinguish it from the one on the source system, you can use the `relocate using <relocate_db_file>` option of the `db2inidb` tool in connection with the `as snapshot` option:
   `db2inidb <DBSID> as snapshot relocate using <relocate db file>`
   All open transactions are then rolled back and the target database can be accessed now.
   The following example shows how you easily create a `<relocate db file>` with the `brdb6brt` tool (option `-replace`) when performing a homogenous system copy using the `db2inidb` tool:
   1. Create the `<relocate db file>` with the `brdb6brt` tool by entering the following command:
      `brdb6brt -bm RETRIEVE_RELOCATE -replace PRD=QAS, db2prd=db2qas`
   2. Check the resulting `<DBSID>_NODEXXXX.scr` file.
      The container paths have been replaced on the right side of the comma with the new required paths, for example:
      `CONT_PATH=/db2/PRD/sapdata1/NODE0000/temp16/PSAPTEMP16.directory000,/db2/QAS/sapdata1/NODE0000/temp16/PSAPTEMP16.directory000`
   3. Call the `db2inidb` tool with the `as snapshot` option and the `<relocate db file>` by entering the following command:
      `db2inidb <DBSID> as snapshot relocate using <relocate db file>`
      The database receives the new references to the container paths that are specified in the `<relocate db file>`.

> **i Note**
>
> You must rename the restored files and directories manually by replacing all occurrences of `"PRD"/"prd"` with `"QAS"/"qas"`.
>
> The `db2inidb as snapshot` command with the `relocate using` option only applies those changes to the database container configuration file.

8.5.4 Using a File System Backup to Set Up a Hot-Standby Database

Prerequisites

You have created a database snapshot of the source system as described in Performing a File System Backup [page 117].

Procedure

> ⚠ Caution
>
> The following procedures describe the `end of log` scenario.
>
> The other scenario, that is, `recovering from logical failures`, works according to the `end of log` scenario, but you must use `rollforward to <point of time>` instead of `rollforward to end of logs` to ensure the time delay for the hot-standby system.

Creating an Initial Hot-Standby Database Version

1. To create a database instance using the Db2 instance creation tool `db2icrt`, enter the following command on the hot-standby system:
   ```
   db2icrt <instance_owner>
   ```
2. Adapt the instance and database configuration settings and the profile registry settings according to the settings of the source system.
3. Log on as the instance owner.
4. Create the database by mounting the split image from the source system.
5. Initialize the hot-standby database by entering the following command:
   ```
   db2inidb <DBSID> as standby
   ```

The open transactions are not rolled back and the hot-standby database is now in *rollforward pending* mode.

Preparing the Hot-Standby Database

If you are using TSM, you have to set the TSM configuration parameter `NODENAME` on the hot-standby system as you did on the source database.

Keeping the Hot-Standby Database Synchronized with the Source Database

To keep the hot-standby database synchronized with the source database, you have to perform a rollforward recovery on a regular basis on the hot-standby system (without the `stop` or `complete` option) by entering the following command:

```
db2 rollforward db <DBSID> to end of logs
```

The target system then retrieves the required log files directly from the storage management system where the source database had stored them before.

Switching Over to the Hot-Standby Database System

If you are using **direct** archiving, you perform a rollforward recovery to the end of log files on the hot-standby system as follows:

1. Start the rollforward recovery by entering the following command:
   ```
   db2 rollforward db <DBSID> to end of logs
   ```
2. Check if the rollforward recovery is complete by entering the following command:
   ```
   db2 rollforward db <DBSID> query status
   ```
   If the rollforward recovery is not complete, you have to correct the error, for example, by providing missing log files and by repeating step 1 of this procedure.
3. Stop the rollforward recovery by entering the following command:
   ```
   db2 rollforward database <DBSID> complete
   ```
   If the operation was successful, the hot-standby database is now in the most current state and can be accessed again. Make sure that the former primary database server cannot be accessed in parallel after it has been made available again.

8.5.5 Using a File System Backup to Create a Db2 Backup Image

Prerequisites

You have created the split image as described in Performing a File System Backup [page 117].

Procedure

Perform the following steps on the target system:

1. To create a database instance using the Db2 instance creation tool `db2icrt`, enter the following command:
   ```
   db2icrt <instance_owner>
   ```
2. Adapt the instance and database configuration settings and the profile registry settings according to the settings of the source system.
3. Log on as the instance owner.
4. To start the database instance, enter the following command:
   ```
   db2start
   ```
5. Recreate the database from the split image:
   1. Mount the split image.
   2. Initialize the standby database using the following command:
      ```
      db2inidb <DBSID> as standby
      ```
6. To create a database online backup using the Db2 backup command, enter the following command:
   ```
   db2 backup db <DBSID> online to <destination_directory>
   ```

Performing the Restore

The backup images that were created with the backup command on the standby system can be used as normal backups for the source system. Therefore, the restore procedure is equivalent.

> **Note**
>
> The backup performed on the standby system is not listed in the backup history of the source system.

8.6 Checking the Database for Consistency

Database consistency is essential for the integrity of your data. While typically very rare, inconsistencies can have various causes such as software or hardware failures. Symptoms of inconsistencies can range from page corruption to data structures that no longer reference each other properly.

During regular operations, Db2 only checks data that is accessed. Any failure during data access is reported immediately. However, data that only resides on disk and that is not touched is not checked. Therefore, if corruption occurs to such data, for example, because of a disk corruption, the problem might stay undetected for a long time.

Inconsistencies might cause permanent data loss if they are not detected early. For example, if a data page became corrupt and you do not have any backup available that does not contain the corruption, your data will be lost permanently if you are not able to reconstruct the lost data from other sources.

You should check your database regularly to detect potential inconsistencies early. To check your database for consistency, you can use one of the following options:

- Db2 `INSPECT` command
- Checksum checking during backup

INSPECT Command

The `INSPECT` command checks that the structures of Db2 objects like tables, indexes, and tablespaces are valid. The command includes checks for the integrity of the objects, that is, for example, for correct pointers from indexes to tables, for anchor points of LOBs, and so on. These checks are extensive and can therefore negatively impact performance and cause a lot of I/O. The `INSPECT` command generates an output file that you have to format using the `db2inspf` tool.

To check the entire database, run the command as follows:

```
db2 "inspect check database for error state all results keep inspect.out"

db2inspf inspect.out inspect.txt
```

To analyze a tablespace, use the following:

```
db2 "inspect check tablespace name <tablespace name> for error state all results
keep inspect.out"

db2inspf inspect.out inspect.txt
```

The inspect utility stops processing a table object if an error state is detected. This default behavior lets you detect an erroneous table, but may not report all affected pages of the table object.

To report all pages with errors found, you can specify the `LIMIT ERROR TO ALL` clause. This reports all errors and does not stop after a certain number is reached. If there are many defective pages, the runtime of the `INSPECT` command and the size of the output file increase, but the result contains the information needed by support personnel for further investigations.

The command sequence without limit to the default number of errors for the entire database is as follows:

```
db2 "inspect check database for error state all limit error to all results keep inspect.out"

db2inspf inspect.out inspect.txt
```

Checksum Checking During Backup

During Db2 backup operations, data is by default not checked for errors. You can, however, optionally enable checksum checking during the backup. The registry variable `DB2_BCKP_PAGE_VERIFICATION` specifies whether DMS and AS pages are validated during the backup. It applies to all operating systems. Possible values are *FALSE* (default) or *TRUE*. You can change the setting of `DB2_BCKP_PAGE_VERIFICATION` online, that is, without having to restart the instance. As of Db2 11.1 Modification Pack 1 Fix Pack 1, `DB2_BCKP_PAGE_VERIFICATION` is set by default (under the aggregate registry variable `DB2_WORKLOAD=SAP`).

> **i** Note
>
> Checksum checking only applies to DMS data pages and not to LOB data. LOB pages do not have checksums.

If the backup check detects a potential problem, it aborts the backup operation with an error. In such a case, make sure that you check your database for errors using the `db2dart` or `INSPECT` command as soon as possible.

In some rare cases, the backup check might report false positives, that is, it might indicate a possible error although there is none. This is due to the fact that the backup works on raw data and does not follow the pointers of the metadata. It therefore cannot differentiate between LOB pages and other pages. In the unlikely case that a LOB page looks as if it were a DMS data page with a checksum, the backup check might try to do a check for checksums and wrongly indicate that the checksum is invalid. You should then restore the backup to a different machine and check it with `db2dart` to verify if there is an actual error.

If you do not use the Db2 backup command for database backups, for example, if you use manual flash copy backups, you can perform a Db2 backup to the `NULL` device (on UNIX: `/dev/null`, on Windows: `NUL`) if you want to execute a checksum check with the Db2 backup.

9 Copying an SAP System Using Db2 Tools

With a homogeneous system copy, you can create a copy of a production system on a test system. Although you can perform a homogeneous system copy by exporting the source system and by creating the target system with the exported content, database-specific methods like backup and restore or the use of a split image are often more efficient to copy the database content from the source system to the target system.

Be aware, however, that if you use the backup and restore method, you cannot change the database schema. The database schema will be the same as the database schema of the source system. In addition, a database backup includes all objects in the database. If you have installed multiple SAP systems in the same database, you cannot copy only an individual component to another system.

> **i Note**
>
> If you want to refresh your test system regularly with data from the production system, you might consider exchanging only the database content instead of first deleting the entire SAP system and then reinstalling it with the SAP installation tool.
>
> Note, however, that the SAP installation tool makes additional adaptations to the database, such as granting appropriate user rights and transferring ownership of database objects. If you do not use the SAP installation tool for your system copy, you need to perform these steps manually. We recommend that you use the SAP installation tool.

The following sections provide information about how to copy the database using database-specific tools and methods:

- Redirected Restore [page 125]
- Building a Target System from a Split Image [page 126]
- Relocating Database Container [page 127]

> **i Note**
>
> The database-specific methods described in this document focus **only** on the creation of the **target database** as part of a homogeneous system copy.
>
> To perform a complete **homogeneous system copy** of an SAP system, you have to create the target SAP system using the homogeneous system copy option of the respective SAP installation tool.
>
> For more information, see the homogeneous and heterogeneous system copy guide under References [page 228] in the appendix.

9.1 Redirected Restore

Creating the Redirected Restore Script from a Backup Image

You restore a Db2 backup from the source system on the target system with adaptations. This process is called redirected restore. To perform a redirected restore, you can also create a redirected restore script from an

existing backup image. No access to the source system is necessary. You only require an existing offline or online backup image to perform the redirected restore.

For more information, see Performing a redirected restore using an automatically generated script in the IBM documentation.

Exporting and Importing the Database Manager Configuration (Recommended)

To synchronize the database manager (DBM) configuration parameters between the DBMs of the source and the target system, you can export the DBM configuration from the source system and import the settings on the target system.

1. To export the DBM configuration, enter the following command on the source system host:
   `db2cfexp <filename> BACKUP`
2. To import the exported DBM configuration parameters on the target host, enter the following command:
   `db2cfimp <filename>`

This procedure also saves and restores the Db2 profile registry settings.

9.2 Building a Target System from a Split Image

Modern hardware technologies like `EMC Timefinder®` or `IBM ESS®` provide the basis for fast backups or database clone creation. Using these technologies, you can split up an entire file system very fast. This split image can then be used to create a copy of the database on a separate machine or in a different directory of the same machine.

In the context of a homogenous system copy, you can use the `db2inidb` tool with the `as snapshot` option, for example, to create database clones for quality assurance or test systems.

For more information, see db2inidb Option: as snapshot [page 120].

> ⚠ Caution
>
> The procedures provided in these sections are **only** intended for **experienced** Db2 database administrators.

For more information about `db2inidb`, go to the IBM documentation at https://www.ibm.com/docs/en/db2 .

9.3 Relocating Database Containers

With `brdb6brt` patch 5 or higher, you can create relocation files to move existing containers to other directories using the `db2relocatedb` tool. Furthermore, you can use these relocation files to initialize mirrored databases with a modified container layout using the `db2inidb` tool and its parameter `RELOCATE`.

Moving Existing Containers to Other Directories for DMS Tablespaces

> ⚠ Caution
>
> The following steps **do not** apply to automatic storage tablespaces.

1. To create the relocation file, enter the following command:
   `brdb6brt -s <DBSID> -bm RETRIEVE_RELOCATE`
   The text file `<DBSID>_NODExxxx.scr` is generated. You have to modify it according to your requirements.
2. To update the internal container path information of the database using the `db2relocatedb` tool, enter the following command:
   `db2relocatedb -f <DBSID>_NODExxxx.scr`
3. To initialize the mirrored database, for example, to create a database clone using the `db2inidb` tool, enter the following commands:
   `db2inidb <NEW_DBSID> as snapshot relocate using <OLD_DBSID>_NODExxxx.scr`

Changing the Storage Paths for Automatic Storage Tablespaces in Databases Enabled for Automatic Storage Management

If your database is enabled for Db2's automatic storage management, it can have automatic storage tablespaces as well as regular DMS tablespaces. The database has one or more storage paths (which are database parameters) and automatically allocates space to the automatic storage tablespaces. The regular DMS tablespaces are handled as described earlier in this section under *Moving Existing Containers to Other Directories for DMS Tablespaces*.

To change the storage paths for the automatic storage tablespaces, proceed as follows:

1. To create the relocation file, enter the following command:
   `brdb6brt -s <DBSID> -bm RETRIEVE_RELOCATE`
   The file `<DBSID>_NODExxxx.scr` is generated.
2. Edit the file `<DBSID>_NODExxxx.scr` and change the automatic storage paths for the automatic storage tablespaces.
3. To update the internal container path information of the database using the `db2relocatedb` tool, enter the following command:
   `db2relocatedb -f <DBSID>_NODExxxx.scr`

> **i Note**
>
> As of Db2 10.1, storage paths are assigned to a storage group rather than to the database. Nevertheless, existing storage paths of any storage group are displayed in the relocation file and can be changed there.

Changing Text During Script Generation

`brdb6brt` creates the scripts that are used to perform a redirected restore and the text files that are used to relocate the database (`relocate DB` file). You then have to adapt the respective script or text file according to your requirements.

With `brdb6brt` patch 5 or higher, parameter `-replace <ReplaceDefinition>` was introduced. You use this parameter to adjust the script output during its generation instead of adapting the output manually afterwards.

> **Example**
>
> You use this parameter to change the target database name from PRD to QAS and the container location from `/db2/PRD` to `/db2/QAS` as follows:
>
> ```
> brdb6brt -s PRD -bm RETRIEVE -replace
> ```
>
> ```
> PRD=QAS,db2prd=db2qas
> ```

10 Database Locking Mechanisms

10.1 Locking Concepts

Db2 uses locks to protect data integrity and ensure isolation when multiple concurrent applications access the same data. Applications can influence the locking behavior of Db2 by selecting a specific isolation level. Possible isolation levels are standardized in the ANSI/ISO SQL standard, but Db2 uses its own terminology for the isolation levels.

The following table summarizes the possible isolation levels and lists the possible concurrency problems within each level:

ANSI/ISO Isolation Level	Db2 Terminology	Possible Concurrency Problems
Read Uncommitted	Uncommited read (UR)	Dirty reads, non-repeatable reads, phantoms
Read Committed	Cursor stability (CS)	Non-repeatable reads, phantoms
Repeatable Read	Read stability (RS)	phantoms
Serializable	Repeatable read (RR)	-

The use of higher isolation levels reduces the concurrency on the database and increases the risk of a deadlock.

Db2 keeps track of its locks with the help of an in-memory lock list. There is one lock list per database partition. The size of the lock list is determined by the database configuration parameter `LOCKLIST`. Db2 uses row-level locking by default, but it can also acquire locks on complete tables, table blocks (MDC), tablespaces, buffer pools, and databases.

Locks are released at the end of a database transaction (at `COMMIT` or `ROLLBACK`). Db2 uses various types of locks, for example, shared locks, update locks, or exclusive locks. Locks are also acquired during read operations with isolation level CS or higher. This means that a transaction that issues only `SELECT` statements also needs to be completed with a `COMMIT` or `ROLLBACK` to release all locks.

Lock waits occur if an application tries to acquire a lock currently held by another application. The time an application may wait for a lock to become available is defined by the database configuration parameter `LOCKTIMEOUT`. If an application waits longer than the `LOCKTIMEOUT` limit, the database transaction is rolled back and the application receives *SQLCODE -911* with reason code *68*. This situation is called a lock timeout.

Deadlocks can occur if two or more applications wait for each other in a cyclic chain of locks. Db2 has an internal deadlock detector that helps resolve deadlock situations. If the deadlock detector finds a deadlock, it arbitrarily chooses one of the participating transactions and rolls it back. The application receives *SQLCODE*

-911 with reason code *2*. How often the deadlock detector is running is determined by the database configuration parameter `DLCHKTIME`.

If Db2 does not find enough space in the lock list when it tries to acquire more locks, a lock escalation is triggered. During a lock escalation, Db2 converts the row-level locks into a table-level lock to free up space in the lock list. Such a lock escalation not only degrades concurrency because the complete table is locked now, but it is also likely to result in a deadlock.

Lock escalations should be avoided. They typically occur if an application makes many changes in one single big transaction instead of breaking the workload into smaller pieces with more frequent commits.

> **i** Note
>
> With Db2 11.1 Mod 2 Fix Pack 2, the Db2 registry variable `DB2_AVOID_LOCK_ESCALATION` is introduced. If it is set to ON, which is the default for SAP installations (DB2_WORKLOAD=SAP), Db2 will **not** perform lock escalation. Instead, the SQL error *SQL0912N* is returned to the application that requested the lock that would normally result in lock escalation.The application is able to either COMMIT or ROLLBACK which will free the locks held by this application.

In SAP installations, the Db2 aggregate registry variable `DB2_WORKLOAD` is set to value *SAP*. It also implicitly activates the settings of the following registry variables that affect the locking behavior:

- `DB2_EVALUNCOMMITTED`
  For more information, see Evaluate uncommitted data through lock deferral in the IBM documentation.
- `DB2_SKIPINSERTED`
  For more information, see Performance variables in the IBM documentation.

These variables improve the concurrency.

You can configure Db2 to use the new currently committed (CC) semantics for queries with isolation level CS. With currently committed semantics, readers do not wait for writers to release row locks. Instead, readers see data that is based on the currently committed version, that is, the version of the row prior to the start of the write operation. The use of currently committed semantics is recommended for SAP NetWeaver 7.0 and higher.

More Information

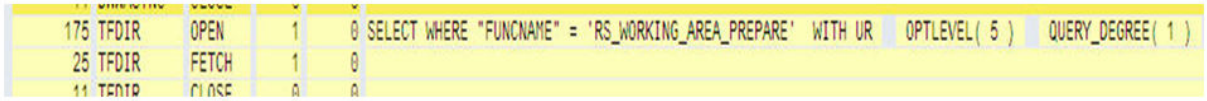
Isolation levels in the IBM documentation

10.2 Locking Mechanisms in an SAP Environment

SAP applications use the uncommitted read (UR) isolation level with IBM Db2. SAP applications protect themselves against uncommitted changes by logical locks on application level. These logical locks are managed by the SAP enqueue server through a central lock table. They are fully portable and uniform across database vendors.

Only in specific cases, AS ABAP uses the cursor stability (CS) isolation level to isolate a query against concurrent changes. In addition, AS ABAP might use the read stability (RS) isolation level in exceptional cases for reads from SAP cluster tables and for reads from the table buffer component.

As of SAP NetWeaver 7.0 ABAP, the isolation level is always appended to the SQL statement using an isolation clause (`WITH <iso_level>`). This lets you see the specified isolation level in the SQL trace (transaction `ST05`). The following is an excerpt of an SQL trace where the `WITH UR` (uncommitted read) clause is appended to the `SELECT` statement:



SQL Trace with “WITH UR” Clause

More Information

SAP Note 1514016

10.3 Monitoring Lock Activity and Deadlocks

10.3.1 Monitoring Lock Activity and Deadlocks Using the DBA Cockpit

The DBA Cockpit offers various functions for monitoring locks and deadlocks. To access these functions, call transaction `DBACOCKPIT` in your SAP system and choose one of the following screens in the DBA Cockpit:

> **i** Note
>
> As of Db2 11.1 Mod 2 Fix Pack 2, Db2 does not perform lock escalation anymore due to the new Db2 registry variable `DB2_AVOID_LOCK_ESCALATION` that is set to ON by default in SAP installations. For more information, see Locking Concepts [page 129].

- *Performance* > *Database*
  On the *Database* screen, choose a line in the overview table and the *Locks and Deadlocks* tab page appears in the content detail area.
  Use this function to get an overview of the size and current use of the lock list, the lock wait situations, and the lock escalations that have occurred since the DBMS start or the last reset of the values.
- *Diagnostics* > *Lock-Wait Events* or *Lock Waits and Deadlocks* (as of Enhancement Package 2 for SAP NetWeaver 7.0)
  Use these screens to check whether there currently is a lock wait or deadlock situation in the SAP database.

For more information, see the DBA Cockpit documentation [page 228].

10.3.2 Monitoring Lock Activity and Deadlocks on the Db2 Command Line

To monitor locks and deadlocks on the Db2 command line, you can use, for example, the following tools:

- `db6util` tool
- Table function `SNAP_GET_LOCKWAIT`
- Monitoring utility `db2pd`
- DBSL deadlock trace
- Lock event monitor

db6util Tool

During the execution of a database transaction, the DBSL of the ABAP kernel passes SAP-specific information on to the database, which you can display with the `db6util` tool. You can use the `db6util` tool to list lock wait situations or deadlocks that currently exist in the database.

To show lock waits and deadlocks, use option `-sl`. To show only deadlocks, use option `-sd`.

You can call the `db6util` tool in such a way that its execution is repeated several times. To let the `db6util` tool check, for example, 20 times for lock waits and deadlocks with a pause of 10 seconds between the checks, enter the following command:

```
db6util -sl 10 20
```

The output, which you can redirect to a file using the option `-w`, contains an ASCII graph that shows the dependency of all processes participating in the lock wait or deadlock situation.

The following is an example of a `db6util` output that shows information about a simple lock wait situation:

```
SNAPSHOT TAKEN AT: 20070921165804
---------------------------------
No deadlocks were detected

LOCK WAITS:
----------
  18                30
 (PID:843936)  <-- (PID:831690)
 disp+work         disp+work

DETAILED INFORMATION ABOUT LOCKED PROCESSES:
-------------------------------------------
  ID    PID      APPL-NAME    HOSTNAME(PART)  MODE RMODE OBJECT TABLE
  18    843936   disp+work    10.17.202.102(0)
  Status   : UOW Executing
  User Id  : TESTUSER1
  Wkstn    : is0016
  Appl.    : SE38
  Acc. Info: SAPLS38E
  Last SQL : UPDATE "ZJOTEST1" SET "C2" = ? WHERE "C1" = ? OPTLEVEL(
             5 ) -- QUERY_DEGREE( 1 ) -- LOCATION( ZJOLOCK2, 95 ) --
             SYSTEM( DEF, SAPDEF )
  30    831690   disp+work    10.17.202.102(0)  X    X    ROW   SAPDEF.ZJOTEST1
  Status   : Lock Waiting (11 seconds)
  User Id  : TESTUSER2
  Wkstn    : is0016
```

```
Appl.    : SE38
Acc. Info: SAPLS38E
Last SQL : SELECT * FROM "ZJOTEST1" WHERE "C1" = ? OPTIMIZE FOR 1
           ROWS WITH RS USE AND KEEP EXCLUSIVE LOCKS -- OPTLEVEL( 5 )
           -- QUERY_DEGREE( 1 ) -- LOCATION( !JOLOCK1, 75 ) --
           SYSTEM( DEF, SAPDEF )
```

In our example, the connection with ID *18* of the ABAP work process with process ID *843936* is currently in a database transaction. The work process executed an update on table `ZJOTEST1` as the last SQL statement. The connection with ID *30* of the work process with process ID *831690* has been waiting for 11 seconds after it tried to acquire an exclusive row-level lock on the same table row of table `ZJOTEST1`, which is currently locked by the update operation. You can see that the two work processes involved in the lock situation are in SAP transaction SE38 and that they were started by the SAP users TESTUSER1 and TESTUSER2. You also get information about the active SAP report SAPLS384. This helps you associate lock situations to certain SAP locations.

If you use `db2pd` or `SNAP_GET_LOCKWAIT`, note that they don't provide this information and that you need further tools or table functions in that case.

For more information, see db6util – Tool to Assist Database Administration [page 214] and SAP Note 327595.

Table Function SNAP_GET_LOCKWAIT

To retrieve information about lock waits using SQL, you can use Db2's table function `SNAP_GET_LOCKWAIT` and the administrative view `SNAPLOCKWAIT`.

The following statement, for example, returns information about lock waits of the currently connected database, its current partition, and the agent ID of the participating agents:

```
SELECT AGENT_ID, LOCK_MODE, LOCK_OBJECT_TYPE, AGENT_ID_HOLDING_LK,
LOCK_MODE_REQUESTED FROM TABLE(SNAP_GET_LOCKWAIT('',-1)) AS T
```

The output of this SQL statement can look as follows:

AGENT_ID	LOCK_MODE	LOCK_OBJECT_TYPE	AGENT_ID_HOLDING_LK	LOCK_MODE_REQUESTED
26	X	ROW_LOCK	24	X

Output Example

For more information, see SNAPSHOT_LOCKWAIT table function in the IBM documentation.

Monitoring Utility db2pd

To gather information about lock waits and deadlocks, you can also use the Db2 monitoring utility `db2pd`. This utility reads the information directly from the shared memory of the database and therefore is very fast.

A `db2pd` command looks, for example, as follows:

```
db2pd -db <dbsid> -locks showlocks wait
```

The output of this command can look as follows:

```
Database Partition 0 -- Database DEF -- Active -- Up 0 days 01:19:40


Locks:
Address            TranHdl    Lockname                   Type       Mode Sts Owner
Dur HoldCount  Att  ReleaseFlg

0x0780000020910540 17         00060011000000002417C000552 Row          ..X  W   8
1   0          0x00 0x00008000  TbspaceID 6

   TableID 17     PartitionID 0 Page 147836 Slot 5

0x078000002091BB00 8          00060011000000002417C000552 Row          ..X  G   8
255 0          0x00 0x40000000  TbspaceID 6

   TableID 17     PartitionID 0 Page 147836 Slot 5
```

Output Example of db2pd

The corresponding table name for the shown table ID was retrieved, for example, from the system catalog view `SYSCAT.TABLES`.

For more information, see Problem Determination Tool db2pd [page 165] and, in the IBM documentation, db2pd - Monitor and troubleshoot DB2 database command.

DBSL Deadlock Trace

In case of a deadlock situation, you can use the `db6util` tool to show all deadlock participants together with their last executed SQL statements. However, in some situations, it is necessary to log **all** active SQL statements of the last database transaction. To do so, you can use the DBSL deadlock trace. It logs all active SQL statements of the last database transaction of every work process into a file.

For more information about using the DBSL deadlock trace, see DBSL Deadlock Trace [page 180] and SAP Note 175036.

Lock Event Monitor

The lock event monitor collects information about lock timeouts, deadlocks, and lock waits, and writes this information to tables in the database. Use the DBA Cockpit to display information collected by the Db2 lock event monitor. For more information about the DBA Cockpit, see References [page 228].

11 Performance Considerations

You can access all monitoring and database administration tasks using the DBA Cockpit.

The availability of Db2 monitoring functions in the DBA Cockpit depend on your SAP basis release and your support package level. If you do not have the most current SAP basis release, some of the functions described here might not be available.

To take advantage of the latest monitoring functions, we recommend that you use SAP Solution Manager as the central monitoring system of your system landscape and that you run the DBA Cockpit as part of SAP Solution Manager. You can access all databases that you want to monitor remotely from the DBA Cockpit on the SAP Solution Manager system.

Performance monitoring and tuning basically consists of the following tasks:

- Monitoring Database Performance [page 135]
- Monitoring Dynamic SQL Statements [page 138]
- Updating Statistics for Database Tables [page 138]
- Reorganization of Database Objects [page 141]
- Monitoring Jobs [page 144]
- Monitoring Network Time [page 144]
- Monitoring IO Throughput [page 145]

These sections focus on a few performance indicators and provide basic information about performance analysis.

More Information

- For detailed information about all available monitoring data and instructions on how to correlate and evaluate this data, see Database monitoring in the IBM documentation.
- DBA Cockpit documentation in References [page 228]

11.1 Monitoring Database Performance

Db2 monitors are used to collect detailed information about resource usage. As of Enhancement Package 2 for SAP Netweaver 7.0 SP6 and Db2 V9.7, you can display an overview of the time that is spent in the database using the *Time Spent Analysis* screen in the *Performance* task area of the DBA Cockpit. You can use the data provided on this screen as a starting point for the analysis of time-based problems of your database, such as the following:

- Processing time in various components of the engine
- Wait times for resources

- I/O times

The advantage of time spent monitoring is that you do not only have to rely on standard database key performance indicators (such as the buffer pool hit ratio) but you can also identify how much time is really spent on different kinds of database operations.

For more information, see the DBA Cockpit documentation [page 228].

Checking the Overall Database Performance

You can analyze performance key figures and exceptional situations for the database on the screens in the *Performance* task area of the DBA Cockpit.

In the content detail area of the *Database* screen, you can check key figures on the respective tab pages for the areas listed in the following table:

Performance Area	Description
Buffer pool quality	Buffer pools are database objects that are used to cache database pages in the memory. If the data page of an object is placed in a buffer pool, physical I/O access to disks is avoided. You can assign buffer pools within the tablespace definition to cache data of a particular tablespace. Every Db2 database must have a buffer pool. For each new database, Db2 defines the `IBMDEFAULTBP` buffer pool, which is the default buffer pool for the database. **i Note** The view on the *Buffer Pool* tab page provides an **overall view** of the database, that is, of **all buffer pools**. It only makes sense to interpret the data provided on this tab page if your databases uses **only one buffer pool**. If your database uses more than one buffer pool, choose *Performance* > *Buffer Pools* instead to display performance figures per buffer pool. Make sure that the buffer quality is within the recommended range for all buffer pools.

Performance Area	Description
Locks and deadlock situations	Db2 captures information about locks held by applications on database objects and records lock escalations and deadlock events using deadlock event monitors. You can access this information on the *Locks and Deadlocks* tab page in the content detail area of the *Database* screen. As of Enhancement Package 2 for SAP Netweaver 7.0 SP6 and as of Db2 V9.7, you can also use the following screens in the *Diagnostics* task area of the DBA Cockpit for a detailed analysis: • *Lock Waits and Deadlocks* (SAP GUI) • *Lock-Wait Events* (Web browser-based version of the DBA Cockpit) **i Note** As of Db2 11.1 Mod 2 Fix Pack 2, Db2 does not perform lock escalation anymore due to the new Db2 registry variable `DB2_AVOID_LOCK_ESCALATION` that is set to ON by default in SAP installations. For more information, see Locking Concepts [page 129].
Sort overflows	The Db2 Snapshot Monitor provides cumulative information about the number of heaps used, overflows, and the performance of sorts. These snapshots are used by the SAP performance monitor to display sort overflows. You can access this information on the *Sorts* tab page in the content detail area of the *Database* screen.
Cache qualities	Db2 uses separate caches for reading the system catalog and for SQL statement preparation and execution, which are the catalog cache and the package cache. For both caches the caching ratio must be observed. You can access this information on the *Cache* tab page in the content detail area of the *Database* screen.

i Note

Monitoring data has usually been gathered since DBM start and might therefore not reflect short-term degradations of the database performance.

As of Enhancement Package 2 for SAP Netweaver 7.0 SP7, you can analyze monitoring data that is based on a local history. This local history is gathered for the DBA Cockpit by the data collection framework (DCF). This way, you are not only able to see current snapshot information but you can also specify analysis time frames.

The data collection framework is the infrastructure for collecting time-based database metrics. For more information about the data collection framework, see the DBA Cockpit documentation [page 228].

11.2 Monitoring Dynamic SQL Statements

Use

The Db2 statement cache stores packages and statistics for frequently used SQL statements. By examining the contents of this cache, you can identify the dynamic SQL statements that are most frequently executed and the queries that consume the most resources. With this information, you can examine the most frequently executed and most expensive SQL operations to determine whether SQL tuning might improve database performance.

Procedure

To access information about the dynamic SQL cache, call the DBA Cockpit and choose *Performance* > *SQL Cache* as described in the DBA Cockpit documentation.

Checking the Access Plan Using the EXPLAIN Function

After you have identified statements that are either called very often or have exceptional execution times, you can check the quality of the access plan chosen by Db2 with the `EXPLAIN` function. The `EXPLAIN` function provides a detailed analysis of SQL statements, for example, information about how Db2 uses indexes to access the data.

To display the access plan for the statement execution, select the SQL statement and choose the *EXPLAIN* pushbutton. To display the ABAP source code from where the SQL statement was executed, you can choose the *Show Source* pushbutton.

More Information

For more information about the `EXPLAIN` function, see the DBA Cockpit documentation under References [page 228] and the IBM documentation at https://www.ibm.com/docs/en/db2 .

11.3 Updating Statistics for Database Tables

The Db2 optimizer uses statistics to determine the best access path. These statistics are stored in the Db2 system catalog. Once user tables have been modified to a significant degree, the statistics data in the system catalog needs to be updated with the latest information to allow the Db2 optimizer to select the best possible access plan.

The statistics in the Db2 system catalog are updated during statistics collection runs of the `RUNSTATS` utility. Db2 automatic table maintenance checks every two hours if statistics are up-to-date and collects new statistics if needed. Db2 real-time statistics allows for even more current statistics information.

In a partitioned database system, statistics are collected only on one database partition. Global table statistics for an entire partitioned table are derived by multiplying the values obtained at the database partition where the collection was executed with the number of database partitions in the database partition group over which the table is partitioned. The resulting calculated global statistics information is then stored in the catalog tables.

More Information

Updating Statistics Using Automatic RUNSTATS [page 139]

11.3.1 Updating Statistics Using Automatic RUNSTATS

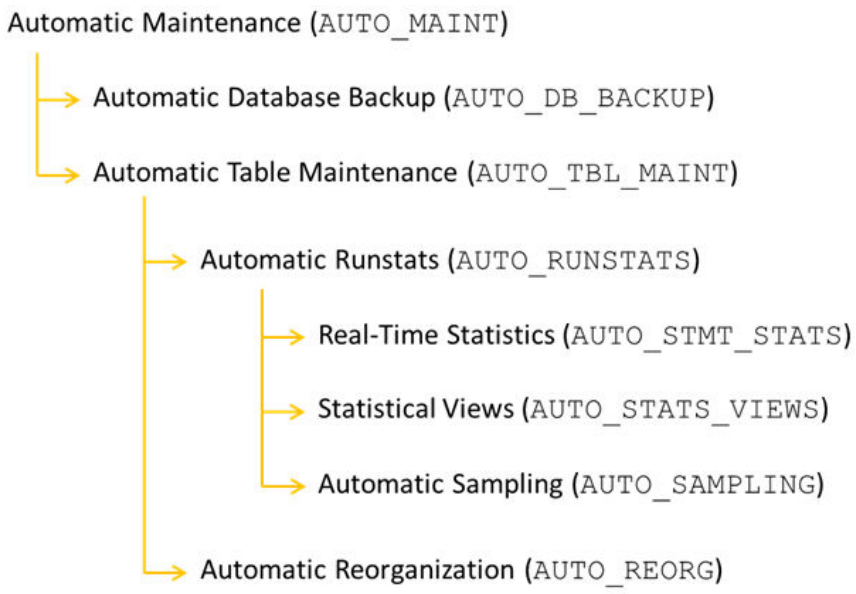
Automatic `RUNSTATS` is part of a completely automated table maintenance solution. Based on the workload, Db2 determines which statistics are required and automatically performs a `RUNSTATS` in the background periodically to update statistics when required. It is mandatory that you enable Db2 automatic `RUNSTATS` for your SAP system.

Enabling Automatic RUNSTATS

Automatic `RUNSTATS` is enabled by default. This means that the following database configuration parameters are set to `ON`:

```
db2 update db cfg for <dbsid> using AUTO_MAINT ON
db2 update db cfg for <dbsid> using AUTO_TBL_MAINT ON
db2 update db cfg for <dbsid> using AUTO_RUNSTATS ON
```

The following figure shows the hierarchy of automatic maintenance commands for statistics collection and their dependencies:



Hierarchy of Db2 Automatic Maintenance Commands for Statistics Collection

Parent parameters in the hierarchy take precedence over child parameters. If you set `AUTO_MAINT` to *OFF*, the settings for child parameters such as `AUTO_RUNSTATS` are irrelevant.

The table maintenance parameter `AUTO_RUNSTATS` enables or disables automatic `RUNSTATS` for a database. To specify the automated behavior, you can use a `RUNSTATS` policy, which is a defined set of rules or guidelines.

With the `real-time statistics` feature the database gathers new table statistics automatically whenever they are needed to optimize and run a query. To enable real-time statistics, you use the `AUTO_STMT_STATS` parameter, which is a child parameter of `AUTO_RUNSTATS`. We recommend that you set the `AUTO_STMT_STATS` parameter to *ON*.

As of Db2 10.1, the additional parameters `AUTO_SAMPLING` and `AUTO_STATS_VIEWS` have been introduced, which are also child parameters of `AUTO_RUNSTATS`:

- `AUTO_STATS_VIEWS` enables the automatic statistics collection on statistical views.
- `AUTO_SAMPLING` controls whether the automatic statistics collection makes use of sampling when collecting statistics for a large table.

Configuring Automatic RUNSTATS

If automatic `RUNSTATS` is enabled, you can configure it in the DBA Cockpit.

Automatic `RUNSTATS` does not generate statistics for tables with the `VOLATILE` attribute. In an SAP ABAP system, there is a predefined set of such tables that are set to `VOLATILE` by default. To display these tables, choose *Configuration* > *Special Tables Regarding RUNSTATS* in the DBA Cockpit.

> **Note**
>
> Prior to Enhancement Package 2 for SAP NetWeaver 7.0, you need to schedule the job `REORGCHK for All Tables` in the DBA Planning Calendar to evaluate the results of the updated statistics for further processing by SAP tools. This additional job is obsolete as of SAP NetWeaver 7.0 Enhancement Package 2.

For more information, see the DBA Cockpit documentation under References [page 228] in the appendix.

11.4 Reorganization of Database Objects

Reorganizing database objects optimizes the physical layout of database objects, such as tables, indexes, and LONG/LOB data.

You can perform a reorganization using the `DB2 REORG` utility to achieve the following:

- Free disk space by making a table and/or its indexes more compact
- Optimize the I/O
- Compress a table for the first time or optimize the compression rate of an already compressed table

> **Note**
>
> The regular reorganization of tables or indexes is neither necessary nor useful in Db2 databases. Reorganizations put a strain on the system and can affect the system availability. Therefore, you should only consider a reorganization if you are sure that it results in measurable benefits.

In addition to "real" reorganizations, there are also maintenance measures for Db2 objects that are executed using the `DB2 REORG` utility. These maintenance measures are:

- Removing pseudo-deleted keys in indexes (`REORG CLEANUP`)
- Releasing completely empty extents after deletions (`REORG RECLAIM`)

Although these maintenance measures are executed using the `REORG` utility, they are no reorganizations in the true sense but a local maintenance of the database object.

> **Note**
>
> If Db2 indicates the need for these maintenance measures, you should always carry them out.

You should automate these maintenance measures by activating Db2 automatic reorganization (`DB2 AUTO REORG`). As of Db2 10.1 and higher, it is mandatory to enable Db2 automatic reorganization.

For more information, see Using Automatic REORG [page 142] and **SAP Note** 975352.

11.4.1 Using Automatic REORG

We recommend that you use the DBA Cockpit to configure automatic reorganizations. To do so, start the DBA Cockpit and choose *Configuration* > *Automatic Maintenance* on the *Database* tab page of the DBA Cockpit or in the SAP GUI navigation frame and go to the *Automatic REORG* tab page.

Prerequisites

- We recommend that you use `DB2 AUTO REORG` for the following Db2 releases:
  - Db2 Version 9.7 FP 4 and higher
  - As of Db2 10.1, you should **always** activate `DB2 AUTO_REORG`.
- You have configured the settings on the *Automatic Maintenance* screen in the DBA Cockpit, in particular:
  - The online/offline window
  - The automatic `REORG` policy
- You have activated `AUTO REORG` by setting the following database configuration parameters:
  - *AUTO_MAINT = ON*
  - *AUTO_TBL_MAINT = ON*
  - *AUTO_REORG = ON*
- Make sure that in the DBA Planning Calendar (transaction DB13), you have **not** scheduled the jobs `Auto_Reorg`, `Reorg_Flag`, or `Reorg_Tbspc` because these jobs are outdated.

Configuration of the Online and Offline Maintenance Window

In the online window, `DB2 AUTO REORG` carries out the following actions:

- Index cleanup
- Reclamation of free index and table space
- Online index reorganization (if allowed)

In the offline window, offline reorganizations are performed. With the SAP-recommended `AUTO REORG` policy, only index cleanups and reclamation of free space are performed.

> → Recommendation
>
> Do not configure an offline window, but configure a 24-hour/7-day online window only.

The Automatic Reorganization Policy

The automatic reorganization policy regulates which objects should be maintained by `DB2 AUTO REORG`. It is an XML document that is stored in Db2 and is interpreted by Db2 automatic table maintenance.

> ⚠ Caution
>
> You must have a suitable AUTO REORG policy installed before you activate DB2 AUTO REORG. Otherwise, DB2 AUTO REORG might execute operations that impair your database availability.

> → Recommendation
>
> We recommend that you use the *Automatic Maintenance* screen in the DBA Cockpit to configure the AUTO REORG policy.

As of Db2 10.1, Db2 automatically installs the SAP default AUTO REORG policy if AUTO REORG is enabled and no AUTO REORG policy is installed. If you upgrade to Db2 10.1 from previous Db2 releases, the `db6_update_db` script updates your AUTO REORG policy when you run it as part of the upgrade post-processing. Therefore, always make sure that a suitable AUTO REORG policy has been installed for Db2 10.1 databases. Nevertheless, we recommend that you check if your installed AUTO REORG policy meets your requirements.

The policy has various parameters that control the behavior of AUTO REORG. You can easily maintain these parameters in the DBA Cockpit. Depending on the support package level, not all of the policy elements might be available.

Scope of DB2 AUTO REORG under DB2_WORKLOAD=SAP

Under the *DB2_WORKLOAD=SAP* aggregate registry variable, DB2 AUTO REORG provides the following functions:

For Db2 Version 9.7

- In the offline window:
  An offline table reorganization is executed on a table if useful and if the table is smaller than the maximum table size. The AUTO REORG policy recommended by SAP excludes offline REORG operations from being executed.
- In the online window:
  - An online index reorganization is executed if Db2 indicates the need and if the underlying table is smaller than the maximum table size. The AUTO REORG policy recommended by SAP excludes online rebuild index REORG operations from being executed.
  - Index cleanup if indicated by Db2
  - As of Db2 V9.7 FP4 and higher: Index cleanup for volatile tables if the parameter *Number Index Pseudo Empty Pages for Volatile Tables* is defined in the automatic REORG policy.
  - Db2 V9.7 only: MDC reclaim based on runtime information

Additional Functions as of Db2 10.1

- Db2 installs the SAP default AUTO REORG policy at the first AUTO REORG execution if AUTO REORG is enabled and no AUTO REORG policy has been installed so far.
- ITC reclaim based on runtime information
- Index reclaim based on runtime information

More Information

For more information about reorganizations in Db2 and about `DB2 AUTO REORG`, see SAP Note 975352 and the DBA Cockpit documentation under References [page 228].

11.5 Monitoring Jobs

You can monitor the runtime of all background jobs and identify long-running jobs by calling the DBA Cockpit and choosing *Jobs* > *DBA Log*.

> **Note**
>
> If background jobs have been scheduled using the DBA Planning Calendar, you can display these jobs by choosing *Jobs* > *DBA Planning Calendar*. In the DBA Planning Calendar, you will find more detailed logs.

For more information, see the DBA Cockpit documentation [page 228].

11.6 Monitoring Network Time

The execution time of an SQL statement executed on an SAP application server is mainly determined by the database time and the network time for communication between database server and application server. If the network connection is slow, the overall application server performance might be dominated by network times.

For an ad-hoc test of the network connection between database server and application server, you can use the `-ping` option of the `db6util` tool. `db6util` connects to the database server, triggers some communication roundtrips between database server and application server, and displays the average communication time and the standard deviation of communication times. If the average communication time is higher than 300 microseconds, check your network connection (a value lower than 300 microseconds is considered good). Unstable network connections can show as communication times with a high standard deviation.

The `db6util -ping` results can change depending on the application server, or after a database switch to another database member or to a standby database.

The following is an example of `db6util -ping` output:

> **Output Code**

```
> db6util -ping
This is the DB6 (DB2 UDB) utility program for SAP kernel 745.
(I) Option "-ping 1 1" .
Ping start timestamp: 20160808131522
-----------------------------------------------------
Average ping time:                   22 micro seconds
Standard deviation of ping times:     5 micro seconds
(I) Connecting to ABC as user sapabc.
```

```
(I) Successfully connected to database ABC .
(I) Setting current schema to SAPABC .
(I) Setting current path to SYSTEM PATH, 'SAPTOOLS' , 'SAPABC' .
(I) Ping start timestamp 20160808131522.
(I) Disconnected from database.
```

For more information about the `db6util` tool and its syntax, see db6util Tool - Command Line Parameters [page 212].

Alternatively, if you are using the DBA Cockpit, you can go to the *Performance* > *Network Statistics* screen of the DBA Cockpit to get details about the communication between the database server and application server to analyze your network performance. For more information, see the DBA Cockpit documentation (under References [page 228] in the appendix) and our video on network statistics in the DBA Cockpit.

11.7 Monitoring I/O Throughput

To access information about the I/O, call the DBA Cockpit and choose *Performance* > *Database* in the navigation frame (SAP GUI) or on the *Database* tab page (Web browser).

In the content detail area of the *Database* screen, you can analyze the following I/O key figures:

- *Synchronous I/O*
  Synchronous I/O takes places if a page needs to be read from disk because it was not found in the buffer pools.
- *Asynchronous I/O*
  Asynchronous writes are performed by I/O cleaners to rewrite changed pages in the buffer pool. Asynchronous reads are performed by I/O servers that prefetch data to be processed.
- *Direct I/O*
  Direct I/O is used to read `LOB` and `LONG VARCHAR` data. Direct I/O does not use the buffer pools but always reads data directly from disk.
- *Logging*
  I/O to the log files is critical for system performance. Changes are logged constantly during transaction processing (write-ahead logging). In addition, a synchronous write is performed at the end of a transaction to ensure that changes are durable. You can use these figures to determine if the logging I/O is a potential bottleneck in your system.

You can check the average I/O times to identify device-specific I/O problems.

12 High Availability

High availability refers to the ability of a system to ensure a certain degree of operational continuity even after a failure of some components.

> **i Note**
>
> In this documentation, we describe the high availability of IBM Db2 ESE installations. High availability with a Db2 pureScale system works differently. For more information about running Db2 with the pureScale Feature, see the relevant installation guide *Running an SAP System on IBM Db2 with the Db2 pureScale Feature* at https://help.sap.com/viewer/p/DB6.

The concept of high availability is based on redundancies of all components. There are different levels of redundancies between hardware (for example, power supply, disks, network adapters, standby server) and software components (for example, database server, application server).

> **i Note**
>
> We refer to the database server as a "software instance".

The following high availability concepts are based on a cluster manager. A cluster manager is a software that is running on a set of cluster nodes and that ensures the availability of services by monitoring, restarting or moving the services at the nodes.

To set up the cluster in a transparent way for all applications, a virtual IP address is used. A virtual IP address is an IP address that is not bound to a specific host. The cluster manager assigns the virtual IP address to the node on which the Db2 software is currently running. An SAP system only knows the virtual host name or the virtual IP address of the database cluster.

Due to the database reconnect feature of the SAP application server, a failure of the database server and a subsequent failover that is controlled by the cluster manager are almost transparent to the clients. The running transactions in the work processes are terminated. The end users of transactions in the affected dialog processes receive a message. SAP NetWeaver reconnects to the database server as soon as it is available again.

Related Information

Setup Types for High Availability [page 147]
Cluster Management Software [page 150]
SAP Adaptive Computing [page 158]

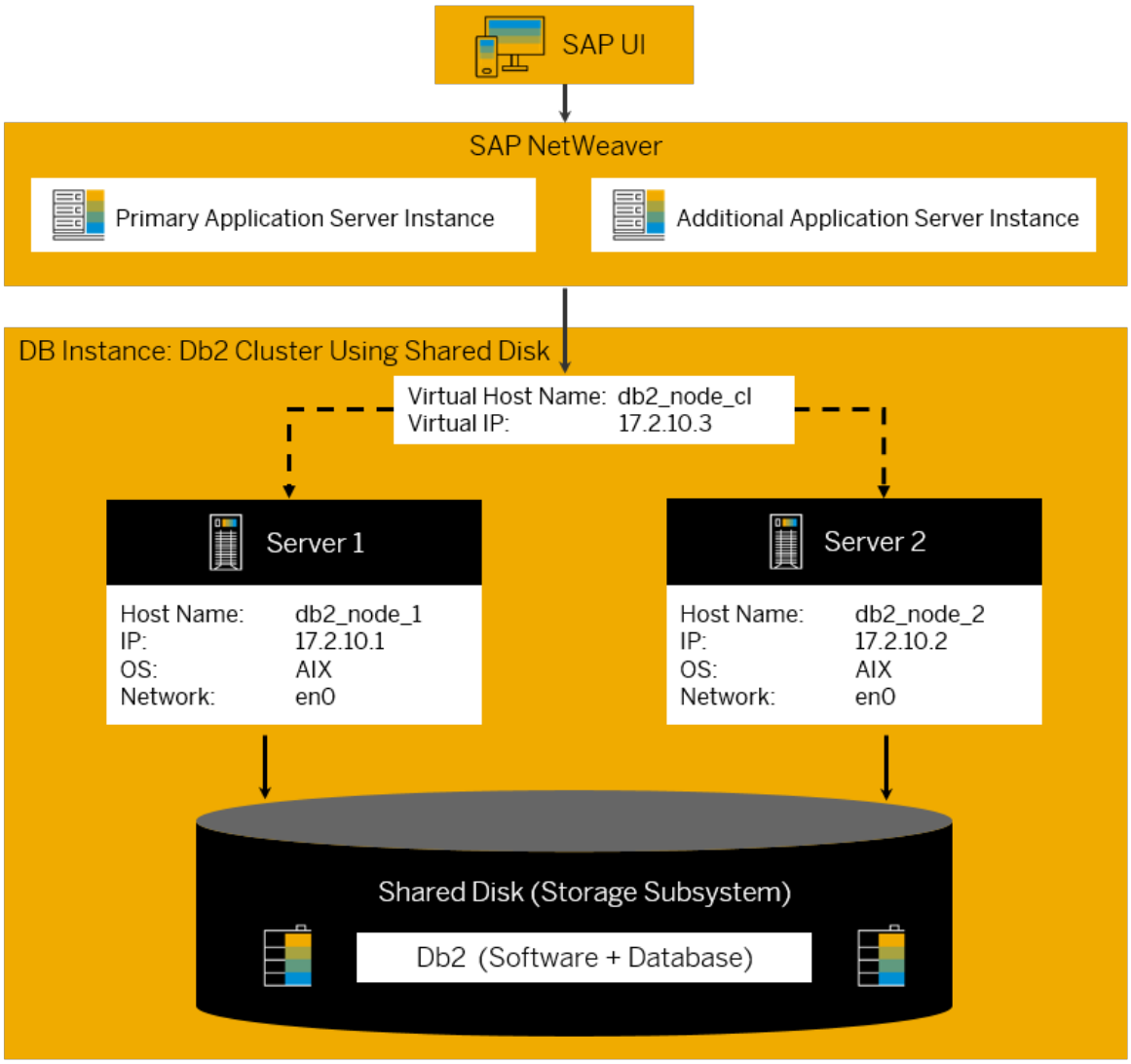
12.1 Setup Types for High Availability

There are two setup types to make a Db2 server highly available using a cluster manager:

- Shared-Disk Approach
  The database is located on a shared disk. The disk is shared between two servers. In the event of a failure of one server, the other server assigns the virtual IP to the network adapter, mounts the shared disk and starts the Db2 instance and database.
- High Availability and Disaster Recovery (HADR)
  HADR is a Db2 replication feature to make the database server highly available (HADR can also be referred to as “shared-nothing approach”). In an HADR scenario, you have two separate Db2 database servers: a primary and a standby database server. Both will be kept in sync and in the event of a failure of the primary database server, the standby database server takes over the workload.

Shared-Disk Approach

The following figure shows a setup of two database servers that share the database software and the database itself on a shared disk:



Shared-Disk Approach

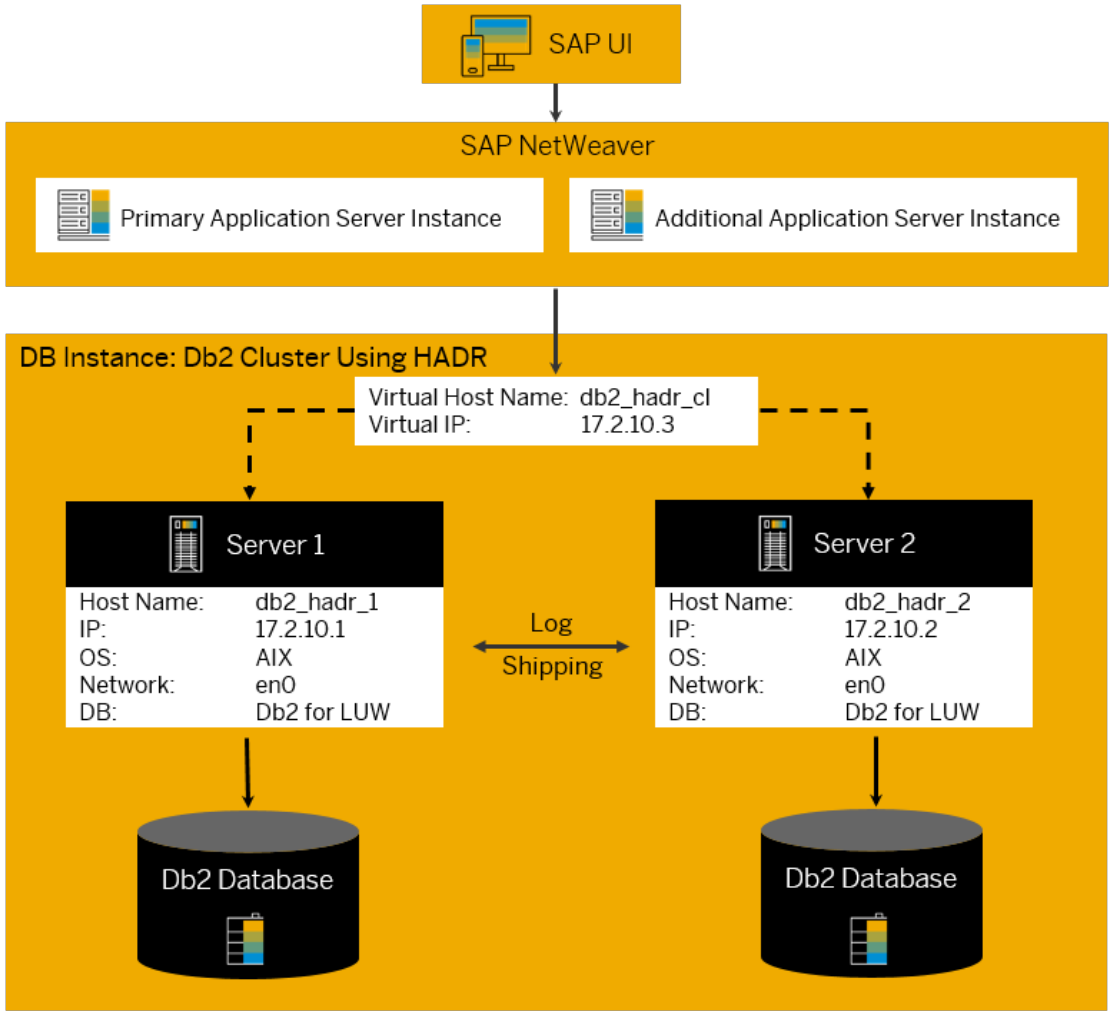
Both database servers have access to the shared disk but only one of them is running at a time and only one of them is connected to the shared disk. The SAP application server instances are running on different machines.

In the case of a failure of server 1, the cluster manager detects the failure, assigns the virtual IP address to a network adapter on server 2 (*en0*), mounts the shared disk and starts the Db2 instance on server 2. During the activation of the database, Db2 triggers a crash recovery to bring the database into a consistent and usable state. All open transactions from server 1 are rolled back and all committed transactions that were still in the memory when the crash occurred are completed.

> **i Note**
>
> The complete crash recovery process is a lot more time-consuming than the HADR approach, which is described in the following.

High Availability Disaster Recovery (HADR)

The following figure shows a setup of two database servers (in this scenario, database server stands for the physical server and the database). Both database servers have their own storage and are up and running.



HADR Setup

In HADR, one server acts as the primary server. This means that all clients are connected to this server. All transactions are written to log files. The log data is transferred to the second database server, that is, the standby server, using TCP/IP. The standby server updates the local database by rolling forward the transferred log records. So, the standby server is kept in sync with the primary server.

> ⚠ Caution
>
> HADR is a replication feature **only**. For failure detection and automation, you need a cluster manager, which monitors the two database servers and initiates the failover to the standby server.

In the event of a crash of the primary database server, the cluster manager initiates the HADR takeover by the standby server and also ensures that the virtual IP address is assigned to the new primary server.

If the primary server crashes, the takeover will only take a few seconds because the second database server is already running and the database itself is already consistent.

In the HADR scenario, the distance between the two database servers can be much larger than in the shared-disk setup because the two database servers only need a network connection to each other. Therefore, you can place them at different locations to protect your data, for example, against fire.

As of Db2 10.1 you can have multiple HADR standby servers. This allows you to have a local HADR standby for fast failover and additional HADR standby servers in other locations for disaster recovery purposes.

For more information, see SAP Note 1612105.

12.2 Cluster Management Software

Cluster management software supports high-availability setups, for example, by providing monitoring to detect system failures and by providing mechanisms to initiate actions without any user intervention.

SAP offers installation support for the following cluster management software, depending on your operating system platform:

- Tivoli System Automation for Multiplatforms (SA MP) – AIX and Linux only
  SA MP is a high-availability cluster solution that provides several monitoring mechanisms to detect system failures and also provides a set of rules to initiate the correct action without any user intervention. The set of rules is called a policy describing the relationships between applications or resources. This policy provides SA MP with extensive up-to-date information about the system landscape so that it can restart the resource on the current node or move the whole application to another cluster node.
  The SA MP license for the database server is included in the Db2 license.
  For more information, see the latest version of the SAP installation guide IBM Db2 High Availability Solution: IBM Tivoli System Automation for Multiplatforms.
- Pacemaker cluster software – Linux only
  Pacemaker is a high-availability cluster solution that provides monitoring mechanisms to detect system failures and that provides a set of resources, constraints, and rules to initiate the correct action without any user intervention. There's a version of Pacemaker available that is integrated into IBM Db2. This version is derived from the ClusterLabs open source project but is enhanced by specific Db2 resource agents and is integrated with Db2 functions such as db2stop or db2start.
  For more information, see Installing Pacemaker with IBM Db2 [page 151].
- Microsoft Cluster Server (MSCS) - Windows only
  For more information about MSCS in an SAP environment, see the relevant installation guide on SAP Help Portal.

12.3 Installing Pacemaker with IBM Db2

Get an overview of the steps you need to perform to install the Pacemaker cluster software with IBM Db2.

Context

Pacemaker is a high-availability cluster solution. You can install a Pacemaker version that is integrated with IBM Db2.

> **i** Note
>
> This overview is about the Pacemaker cluster software. For more information about installing the Tivoli System Automation for Multiplatforms (SA MP), see the SAP installation guide IBM Db2 High Availability Solution: IBM Tivoli System Automation for Multiplatforms.
>
> For more information about installing Microsoft Cluster Server, see the relevant chapters in the installation guide for SAP systems on Windows.

Procedure

1. Set up a standby database server using a homogeneous system copy.

   For more information, see Setting Up a Standby Database Server Using Homogeneous System Copy [page 152].
2. Configure the HADR pair.

   For more information, see Configuring the HADR Pair [page 153].
3. Install and configure the Pacemaker cluster.

   For more information about installing and configuring Pacemaker with Db2, see the IBM documentation Installing the Pacemaker cluster software stack | IBM Db2 and SAP Notes 3100330 and 3100287.
4. Set up a Virtual Host Name and IP Address.

   We recommend that you use virtual IP addresses instead of the Db2 automatic client reroute feature. For more information, see SAP Note 1568539 and Virtual Host Name and IP Address [page 155].

12.3.1 Setting Up a Standby Database Server Using Homogeneous System Copy

If you set up the failover cluster based on the Db2 feature HADR, you must create a standby database as a copy of the primary database. Use the SAP homogeneous system copy for setting up the standby database server.

Procedure

1. Log on to the first node as user `db2<dbsid>`.
2. Enable archiving for the logs, for example, to a disk folder using the following command:

   ```
   db2 "update db cfg for <dbsid> using LOGARCHMETH1 DISK:/db2/<DBSID>/
   log_archive/"
   ```

   > ⚠ Caution
   >
   > Both the primary and the standby database needs to be able to retrieve log files from all the log archive locations to which either of the databases might archive log files.
   >
   > The log archiving is only performed by the primary database. If you change the HADR roles of the database servers or if a failure occurs, the new primary database is responsible for log archiving. If you have set up different log archive locations, your logs might be archived twice and, in the case of local or remote catch-up, you might have to copy the archived logs manually from the old primary server to the active log location of the new primary server.
   >
   > To configure log archiving for HADR, we recommend that you configure both the primary and the standby database to have automatic log retrieval capability from all log archive locations.

3. Create the directory `/db2/<DBSID>/backup` using the following command:

   ```
   mkdir /db2/<DBSID>/backup
   ```

4. Create a full database backup:

   ```
   db2 "backup db <dbsid> to /db2/<DBSID>/backup compress"
   ```

5. Log on to the second node as user `root`.
6. Create the directory `/db2/<DBSID>/backup`:

   ```
   mkdir /db2/<DBSID>/backup
   ```

7. Transfer the backup from the first node to the second node to the directory `/db2/<DBSID>/backup`:

   ```
   scp root@<hostname>:/db2/<DBSID>/backup/* /db2/<DBSID>/backup/
   ```

8. Start the SAP software provisioning manager for installing SAP systems.
9. In the software provisioning manager, choose *Software Lifecycle Tasks* > *System Copy* > *Target System* > *Distributed System* > *<Your setup type>* > *Database Instance*
10. For the copy method, choose homogeneous system copy so that you can use your backup to restore the backup on the standby server.
11. On the *Summary* screen, revise the database communication parameters for the database communication port and the password phrase to ensure that you have the same values as on the primary database server.

12. Start the installation process.
13. When you reach the exit step to restore the database for the homogeneous system copy, exit the installer.

    You restore the database from the primary host. All subsequent installation phases have already been executed on the primary database server.
14. Log on to the second node as user `db2<dbsid>`.
15. Restore the database backup using the following command:

    ```
    db2 "restore db <dbsid> from /db2/<DBSID>/backup replace history file"
    ```
16. Verify that your database is in rollforward pending state using the following command:

    ```
    db2 "get db cfg for <dbsid" | grep Rollforward
    ```

12.3.2 Configuring the HADR Pair

Prerequisites

Before you start to configure the HADR feature, please read SAP Note 1612105.

Procedure

1. Add the IP addresses to the hostname mappings in the `/etc/hosts` file of each database server (primary and standby).
2. Verify that you can find the following entries on both servers using the `# cat /etc/hosts` command:

   ```
   192.168.1.16 <PrimaryHostName>.<DOMAIN> <PrimaryHostName>
   ```

   ```
   192.168.1.17 <StandByHostName>.<DOMAIN> <StandByHostName>
   ```
3. Add the service ports `<SID>_HADR_1` and `<SID>_HADR_2` for HADR to both hosts in the HADR cluster to the `/etc/services` files.
4. Verify that you can find the following entries on both servers using the `cat /etc/services | grep -i HADR` command:

   ```
   <SID>_HADR_1 5951/tcp
   ```

   ```
   <SID>_HADR_2 5952/tcp
   ```
5. Configure the HADR primary database using the following commands:

   ```
   db2 UPDATE DB CFG FOR <SID> USING HADR_LOCAL_HOST <PrimaryHostName>
   ```

   ```
   db2 UPDATE DB CFG FOR <SID> USING HADR_LOCAL_SVC <SID>_HADR_1
   ```

   ```
   db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_HOST <StandByHostName>
   ```

   ```
   db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_SVC <SID>_HADR_2
   ```

```
db2 UPDATE DB CFG FOR <SID> USING HADR_REMOTE_INST db2<sid>
db2 UPDATE DB CFG FOR <SID> USING INDEXREC RESTART LOGINDEXBUILD ON
db2 UPDATE DB CFG FOR <SID> USING HADR_TIMEOUT 120
db2 UPDATE DB CFG FOR <SID> USING HADR_SYNCMODE NEARSYNC
db2 UPDATE DB CFG FOR <SID> USING HADR_SPOOL_LIMIT AUTOMATIC
db2 UPDATE DB CFG FOR <SID> USING HADR_PEER_WINDOW 240
```

6. Configure the HADR standby database using the following commands:

```
db2 UPDATE DB CFG FOR ADH USING HADR_LOCAL_HOST <StandByHostName>
db2 UPDATE DB CFG FOR ADH USING HADR_LOCAL_SVC <SID>_HADR_2
db2 UPDATE DB CFG FOR ADH USING HADR_REMOTE_HOST < PrimaryHostName>
db2 UPDATE DB CFG FOR ADH USING HADR_REMOTE_SVC <SID>_HADR_1
db2 UPDATE DB CFG FOR ADH USING HADR_REMOTE_INST db2<sid>
db2 UPDATE DB CFG FOR ADH USING HADR_TIMEOUT 120
db2 UPDATE DB CFG FOR ADH USING HADR_SYNCMODE NEARSYNC
db2 UPDATE DB CFG FOR ADH USING HADR_SPOOL_LIMIT AUTOMATIC
db2 UPDATE DB CFG FOR ADH USING HADR_PEER_WINDOW 240
db2 UPDATE DB CFG FOR ADH USING INDEXREC RESTART LOGINDEXBUILD ON
```

> **i Note**
>
> The values for `HADR_TIMEOUT`, `HADR_SYNCMODE`, `HADR_SPOOL_LIMIT` and `HADR_PEER_WINDOW` depend on your environment and needs. For more information about these parameters, see the Db2 HADR Wiki .

7. Start HADR first on the newly created standby database using the following command:

```
db2 start hadr on db <SID> as standby
```

8. Start HADR on the primary database using the following command:

```
db2 start hadr on db <SID> as primary
```

Results

You can now continue with the installation and configuration of the Pacemaker cluster software.

For more information about installing and configuring Pacemaker with Db2, see the IBM documentation Installing the Pacemaker cluster software stack | IBM Db2 and SAP Notes 3100330 and 3100287.

12.3.3 Virtual Host Name and IP Address

The virtual host name and virtual IP address are used to access the cluster.

The virtual host name is a reference on the DNS server or to the virtual IP address in the `/etc/hosts` files. The cluster binds the virtual IP address to the active cluster node. All clients must refer either to the virtual host name or directly to the virtual IP address. Therefore, the clients always connect to the node of the cluster where the clustered Db2 or SAP instance is currently running.

To set up such an environment, you can choose between the following two options:

Reuse of Host Name and IP Address of Single Server [page 155]

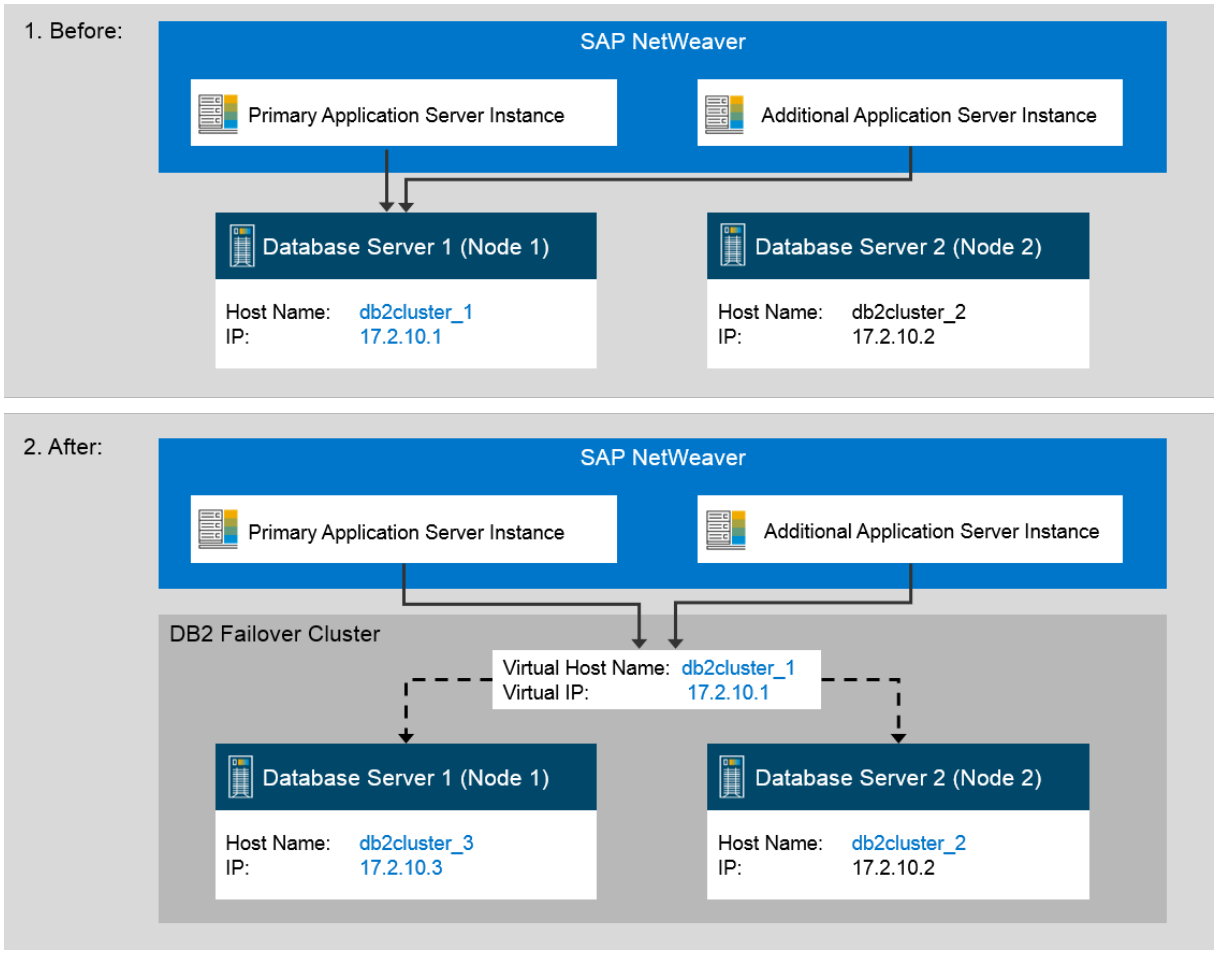
Setup of a New Virtual Host Name and Virtual IP Address [page 156]

> **i** Note
>
> For a new installation or system copy, we recommend that you always use the second option where you set up a new virtual host name and new virtual IP address. The reuse option is only recommended for an existing SAP system that you want to make highly available.

12.3.3.1 Reuse of Host Name and IP Address of Single Server

To reuse the host name and the IP address of a single database server as virtual host name and virtual IP address, you have to replace the physical host name and IP address of the single database server with a new one because you specify the old host name and IP address as the new virtual ones.

The advantage of this approach is that you do not have to change the references to the database host or SAP instance host of your SAP system landscape and of additional monitoring tools. However, additional services that are installed on the server are not available anymore on the specified host name or IP address.



The host name `db2cluster_1` of the single database server is reused as virtual host name, and the IP address *17.2.10.1* is reused as virtual IP address. To avoid naming conflicts, the database server needs a new network identity. Therefore, the database host name is changed to `db2cluster_3` and the IP address to *17.2.10.3*.

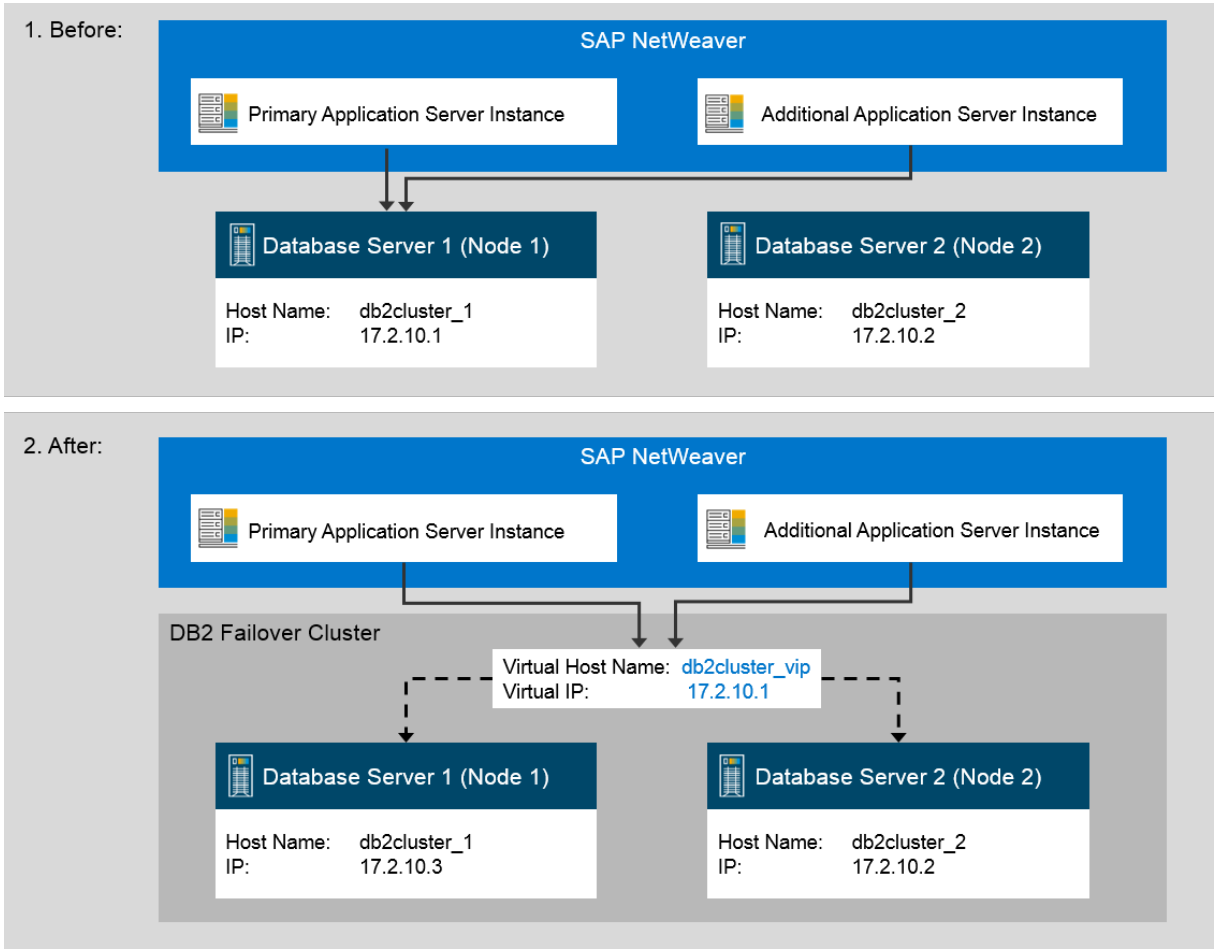
The host name and IP address of the second database server (node 2) remain the same and both database servers are now addressed by the new virtual host name and IP address.

You have to change the entries of the host name in `db2nodes.cfg` to the new physical host name of node 1. This file is located in the Db2 instance directory `/db2/db2<sapsid>/sqllib`.

12.3.3.2 Setup of a New Virtual Host Name and Virtual IP Address

To set up a new virtual host name and new virtual IP address, you have to specify the new host name and IP address in your networking environment.

The following figure shows an example of how to set up a new virtual host name or virtual IP address before and after the database cluster was set up:



The database server keeps its host name `db2cluster_1` and its IP address 17.2.10.1. The new virtual host name `db2cluster_vip` and the new virtual IP address 17.2.10.3 are set for the failover cluster. If you already have an SAP NetWeaver system running on this database server (node 1), you have to change all references to the database host in the SAP system landscape and in the monitoring tools.

To connect to the new database, you must make the following changes manually in the SAP system configuration:

1. Change the parameters `SAPDBHOST` and `j2ee/dbhost` in the default profile that is located in `/sapmnt/<SAPSID>/profile/DEFAULT.PFL`. The database host name must be replaced. According to the example above, change the value from `db2cluster_1` to `db2cluster_vip`.
2. Change the Db2 CLI driver configuration file that is located in `/sapmnt/<SAPSID>/global/db6/db2cli.ini`. To do so, replace the entry `Hostname= db2cluster_1` with `Hostname= db2cluster_vip`.

12.4 SAP Adaptive Computing

Adaptive Computing Controller (AAC) is a tool that provides a single point of control allowing you to operate, observe, and manage your adaptive computing landscape. With ACC, you can flexibly assign computing resources to distribute workload for processing to any server.

The ACC framework is designed to manage a pool of (heterogeneous) servers. Through the movement of services between the servers, you can optimize the capacity utilization and you can activate additional servers on demand.

> **i** Note
>
> The ACC framework does **not** support automated failover. Therefore, it is not a cluster management software to support high availability. However, you can use ACC to easily restart a crashed Db2 database server on a different host after manual intervention.

13 Upgrading or Updating the Database to a Higher Version or Fix Pack Level

Database Upgrade to a Higher Db2 Version

If you have an SAP system running on IBM Db2 and want to upgrade to a higher Db2 version, follow the instructions in the relevant upgrade guide that you can find on our SAP on IBM Db2 page on SAP Help Portal .

Database Update to a Higher Fix Pack Level

If you have an SAP system running on IBM Db2 and want to update the database to a higher Db2 version, follow the instructions in this documentation:

- Updating the Db2 Fix Pack Level [page 159]
- Rolling Update of the Db2 Fix Pack Level [page 160]

We recommend that you also check whether your SAP system running on IBM Db2 is on the right Fix Pack level and which recommended Fix Pack levels are available (see SAP Note 101809). For this purpose, you can use the Db2 Fix Pack level check that is available with SAP Note **2989894** and with the support packages mentioned in this SAP Note.

13.1 Updating the Db2 Fix Pack Level

If you have an SAP system running on IBM Db2 and want to upgrade to a higher Fix Pack level of the same Db2 major release, follow the instructions in the relevant SAP Notes listed below.

These SAP Notes describe how to update the Db2 Fix Pack level of your database engine as a first step, and how to update the Fix Pack level of your DB2 client software as a second step. The SAP kernel requires that the Db2 client has the same major release version as the database server of the SAP system. However, Db2 clients of a lower Fix Pack level are accepted.

Database Version	Operating System	SAP Note
Db2 11.5	UNIX and Linux	2841297
	Windows	2841330

Database Version	Operating System	SAP Note
Db2 11.1	UNIX and Linux	2303756
	Windows	2303766
Db2 10.5	UNIX and Linux	1871003
	Windows	1871004
Db2 10.1	UNIX and Linux	1708037
	Windows	1708038
Db2 9.7	UNIX and Linux	1363169
	Windows	1363170

Related Information

Rolling Update of the Db2 Fix Pack Level [page 160]

13.2 Rolling Update of the Db2 Fix Pack Level

When Can I Do a Rolling Update?

A rolling Db2 Fix Pack update is possible in high availability setups with HADR or Db2 pureScale. In these setups, you can perform the Fix Pack update with minimal or even no downtime. In a rolling update, your database cluster doesn't need to be completely taken offline, but you shut down, upgrade, and bring back online the components one at a time, that is, they are never all offline at the same time.

Rolling Update and SAP Kernel

During a rolling Fix Pack update, your SAP kernel must be able to connect to both Fix Pack levels. Since the SAP kernel only supports downlevel clients (that is, clients of the same or lower Fix Pack level than the Db2 server), you first have to complete the database software update of all of your Db2 members or databases in a cluster before you can update the Db2 client Fix Pack level.

After you have completed the rolling Fix Pack update of your database servers, you can also perform a rolling update of the Db2 client software if you have two or more SAP application servers.

Related Information

Rolling Update in a Db2 pureScale Cluster [page 161]
Rolling Update in a Db2 HADR Cluster [page 161]
Rolling Update of the Db2 Client Fix Pack Level [page 162]

13.2.1 Rolling Update in a Db2 pureScale Cluster

You can perform a rolling update of the database server software in a Db2 pureScale cluster. To do so, follow the instructions in Online fix pack updates in Db2 pureScale environments in the IBM documentation.

13.2.2 Rolling Update in a Db2 HADR Cluster

Context

With a rolling update, you can perform a complete update of the database Fix Pack level of all databases in your HADR cluster almost without any visible restrictions for your SAP end users.

Procedure

1. Install the Db2 software with the new Fix Pack level into a new directory on both the HADR primary and HADR secondary server.
2. Shut down the HADR secondary database and switch to the new Fix Pack level using `db2iupdt`.
3. Restart and resync the HADR secondary database server. The HADR secondary database instance now runs with the new Fix Pack level while the HADR primary database instance still runs with the old Fix Pack level.
4. Perform a role switch between the HADR secondary and HADR primary server. To minimize the downtime for this planned takeover operation, you can use the “Graceful Maintenance Tool” as described in Graceful Maintenance Tool (GMT) for SAP Business Continuity During Database Maintenance [page 191] and SAP Note 1530812.
5. Shut down the former HADR primary and switch to the new Fix Pack level using `db2iupdt`.
6. Restart and resync the former HADR primary database server. The HADR primary database instance and the HADR secondary database instance now run with the new Fix Pack level.
7. If necessary, you can perform another role switch and assign the old roles to your HADR databases. Again, you can use the “Graceful Maintenance Tool” as described in Graceful Maintenance Tool (GMT) for

SAP Business Continuity During Database Maintenance [page 191] and SAP Note 1530812 to minimize downtime.

13.2.3 Rolling Update of the Db2 Client Fix Pack Level

A rolling update of the Db2 client Fix Pack level is possible if you have more than one SAP application server installed. In this case, you can update the centrally installed Db2 client software and then restart the SAP application servers one after the other so that they pick up the new Db2 client software.

As described in Db2 Client Connectivity [page 19], the Db2 client software is installed into a subdirectory of the share `/usr/sap/<SAPSID>/SYS/global/db6`. During the startup of each application server, the client software from the global share is copied to a local directory on the application server, for example, `/usr/sap/<SAPSID>/<Instance Name>/exe`. This allows for a rolling update of the Db2 client software.

You can run an SAP system with Db2 server software of a given Fix Pack level and with Db2 client software of the same major Db2 release, but with an older Fix Pack level. While this setup is completely supported, we recommend that you run your system on the same Db2 client and server software level.

Procedure

1. Install the new Db2 client software Fix Pack using the `db6_update_client` script while your SAP system is running. The script installs the new client software level into the shared global directory. After doing so, your SAP application servers will still use the old client software Fix Pack level from their local directories.
2. Then restart one application server after the other to make them copy and pick up the new client software Fix Pack level.

14 Troubleshooting and Support

Have you run into problems in your system environment and need help with troubleshooting? Have a look at the topics in this section to find out how you can approach and analyze problems, and which tools you can use. Keep in mind, however, since problem scenarios may vary, these are only guidelines and no guaranteed solutions for your specific case.

For additional information about the Db2 tools, see also the IBM documentation for your database version.

14.1 Collecting Db2 Data and Identifying Basic Problems

14.1.1 Diagnostic Directory db2dump and Diagnostic Files

For problems that are detected by the Db2 engine, check the `db2dump` directory for problem-related data. The `db2dump` directory contains various files with diagnostic data.

The `db2dump` directory is configured using the `DIAGPATH` parameter in the database manager configuration. This parameter allows you to specify the directory names per database member and host name. For more information, see the IBM documentation.

If problems occur frequently, the amount of data that is written to the `db2dump` directory can increase quickly. We recommend that you check the file system where the `db2dump` directory is located on a regular basis to prevent the files from becoming too large. You should also move data that is no longer required to another location.

> **i Note**
>
> To avoid a negative impact on other parts of the system, make sure that the `db2dump` directory resides in its own file system.

The following files in the `db2dump` directory are particularly important:

- Db2 diagnostic log (`db2diag.log`)
- Db2 notification log
  On UNIX and Linux, the data is written to the file `<Db2 instance name>.nfy`, for example, `db2abc.nfy`.
  On Windows, the notification records are written to the *Event Viewer* in the *Application* section.

The following database manager configuration parameters influence the diagnostic log information:

Parameter	Description
DIAGPATH	Path to which diagnostic information is written

Parameter	Description
ALT_DIAGPATH	Alternative diagnostic path that is used when the primary diagnostic path is not available
DIAGLEVEL	Type of diagnostic information recorded in the `db2diag.log` A default value of *3* is appropriate for normal operation.
NOTIFYLEVEL	Type of notification message recorded in the notification log A default value of *3* is appropriate for normal operation.
DIAGSIZE	Controls the file sizes of `db2diag.log` and the notification log

For more information about how to manage these diagnostic files, see Diagnostic Tool db2diag [page 164].

14.1.2 Db2 Tools for Collecting and Analyzing Diagnostic Data

14.1.2.1 Diagnostic Tool db2diag

The `db2diag` tool helps you reduce the amount of data in the `db2diag.log`.

You can use the `db2diag` tool to:

- Filter the Db2 diagnostic data using various filtering options
- Search for occurrences of specified error codes
- Format the content of the Db2 diagnostic files
- Restrict search operations to a certain time interval, to certain process IDs (PIDs), thread IDs (TIDs), or partitions
- Interpret Db2 error return codes
- Archive a Db2 diagnostic file

To become familiar with the `db2diag` tool, call the help function by entering the following command:

```
db2diag -h
```

To call the detailed version providing all possible options, enter the following command:

```
db2diag –h all
```

To retrieve help information about one or more options, enter the following command:

```
db2diag –h <option>[,<option>[,…]]
```

> Example
>
> ```
> db2diag –h rc
> ```

14.1.2.2 Problem Determination Tool db2pd

Using the `db2pd` tool, you can retrieve information from many different areas in Db2 while minimizing the negative impact on the database environment. The `db2pd` tool allows you to look into areas of Db2 that are otherwise hidden, for example, the Db2 catalog cache.

> ### Example
>
> You can use the `db2pd` tool instead of a snapshot in the following situation:
>
> To investigate database problems, support teams frequently ask for snapshots. Running a snapshot, however, requires the allocation of Db2 resources. An application snapshot, for example, consistently retrieves all data from all application processes for a given time stamp. In addition, some processing of the Db2 engine is locked to get consistent data.
>
> For troubleshooting, however, consistency of the dumped data is often not necessary. Therefore, using the `db2pd` tool instead of running a snapshot can be more convenient. The `db2pd` tool directly reads the shared memory areas of the Db2 processes in a dirty-read manner preventing resources from being locked.

The following table shows some of the options that you can use with the `db2pd` tool:

Command Line Options	Description
`db2pd -help`	Retrieves help information
`db2pd -db <dbsid> -applications`	Retrieves application information
`db2pd -db <dbsid> -locks`	Retrieves information about existing locks
`db2pd -osinfo`	Retrieves operating system-specific information
`db2pd -db <dbsid> -catalogcache`	Retrieves contents of the catalog cache
`db2pd -db <dbsid> -logs`	Retrieves information about database log files

14.1.2.3 db2support Utility for Problem Analysis and Environment Data Collection

To collect information about a Db2 problem, one of the most important Db2 utilities that you have to run is `db2support`. The `db2support` utility is part of the Db2 software and uses, for example, certain functions of the Db2 explain facility to retrieve data from the Db2 optimizer.

> ### → Recommendation
>
> Since the Db2 instance owner `db2<dbsid>` has the maximum authorization in the database, we recommend that you run the `db2support` utility as user `db2<dbsid>`.

Collecting General Diagnostic Data Using db2support

To collect a general set of data, for example, in the case of a database crash, proceed as follows:

1. Log on to the database server as user `db2<dbsid>`.
2. On the command line, enter the following command:
   `db2support <path> -d <dbsid> -c -s -f [-o <output file>]`
   The following table explains some of the available parameters:

Parameter	Description
<path>	Path to which the output file is written
-d <dbsid>	Name of the database, which can be the same as the SAP system ID
-g	Copies all dumps from the `db2dump` directory to an archive This parameter is **not** available anymore as of Db2 10.1.
-c	Connects to the database to collect additional information **i Note** If you cannot connect to your database, omit this option.
-s	Collects database system information
-f	Unattended mode
-o <output file>	Name of the file to be created in the provided path By default, the file name is *db2support.zip*.

Investigating the Performance of an SQL Statement

To investigate the performance of an SQL statement, information about, for example, the following is required:

- Structure of tables involved and their indexes
- Statistics information about the tables involved
- The access plan that the Db2 optimizer used for this statement

The `db2support` utility collects data that allows a first thorough analysis. Based on this data, the support team either provides suggestions to improve the performance of the statement or requests additional information.

To collect optimizer data for a statement, the `db2support` utility executes the Db2 explain facility. As a prerequisite, you must create a file, insert the statement into this file, and make sure that the statement terminates with a semicolon.

If the tables that are specified in the statement text do not explicitly include the table schema as it is usually the case in an SAP environment, provide the table schema by entering the following command:

```
db2support <path> -d <dbsid> -c -s -f -cs <schema> -sf <statement_file>
```

where <schema> is the table schema and <statement_file> is the file that contains the statement text.

As a result, the `db2support.zip` file is created that contains the `db2supp_opt.zip` archive. `db2supp_opt.zip` contains data of the Db2 optimizer.

> **i Note**
>
> The `db2support` utility does not clearly report success or failure of the optimizer data collection. Therefore, make sure that you check if `db2supp_opt.zip` contains the file `exfmt_badquery.opt_out`. This is a good indicator that optimizer data has been collected.
>
> For troubleshooting during the process of collecting data of the Db2 optimizer, check the Db2 optimizer screen. You can find the screen output of the Db2 optimizer in the `optimizer.log` file in the `db2support.zip` file. In many cases, messages that are issued while the SQL statement that is contained in the `bad_query.sql` file is invoked lead you to the source of the problem. You can find these messages in the `optimizer.log` file.

More Information

- Collecting environment information with the db2support command in the IBM Db2 documentation
- SAP Note 83819: *DB6: Collecting support data*

14.1.2.4 db2fodc Tool for First Occurrence Data Capture

You can also use the `db2fodc` tool for the collection of data to investigate problems such as performance issues or database hang situations.

> **→ Recommendation**
>
> Since the Db2 instance owner `db2<dbsid>` has the maximum authorization in the database, we recommend that you run the `db2fodc` utility as user `db2<dbsid>`.

`db2fodc` collects and dumps the required diagnostic data into the `db2dump` directory. After having run the `db2fodc` utility, you use the `db2support` utility to collect the dumped data and to forward it to the SAP Db2 support team for further analysis as described in db2support Utility for Problem Analysis and Environment Data Collection [page 165].

> **⚠ Caution**
>
> Executing the `db2fodc` utility can severely impact the performance of your database system. Especially the keyword *full*, which is available for some parameters of the tool, might bring a system to a standstill, particularly during peak load times. You can interrupt the execution by choosing **`Ctrl+C`** at any time while the utility is running.

Examples of Using db2fodc

To collect data from a specific database (`SAMPLE`) in a potential database hang situation, enter the following command as user `db2<dbsid>`:

```
db2fodc -db SAMPLE -hang
```

To collect data from a specific database in the case of a performance issue, enter the following command as user `db2<dbsid>`:

```
db2fodc -db SAMPLE -perf
```

To display all options available for the `db2fodc` utility, enter the following command as user `db2<dbsid>`:

```
db2fodc -help
```

More Information

db2fodc - Db2 first occurrence data collection command in the IBM documentation

14.2 Troubleshooting Using the DBA Cockpit

14.2.1 EXPLAIN Function

Using the `EXPLAIN` function in the DBA Cockpit, you can check the optimizer access plan of an SQL statement. In addition, you can investigate how the behavior of the Db2 optimizer changes if, for example, an SQL statement was modified or the used parameters were changed.

There are several ways to access the `EXPLAIN` function in the DBA Cockpit. You can, for example, go to *Diagnostics* > *EXPLAIN* in the navigation frame of the DBA Cockpit (SAP GUI version) and enter a statement. To analyze the access plan in more detail, you can, for example:

- Change the used optimization levels and specify optimization guidelines
- Retrieve Db2 catalog information for a table or index involved in the optimizer access plan
- Collect and download support data that is required to analyze the optimizer access plan if requested by the SAP support team

In addition, with SAP Notes 3049243, 3094083, and 3038068, or the related support packages, you can also use the following functions:

- Display the execution plan of an already prepared SQL statement from the SQL cache using the *Explain from Cache* function.
- Display the execution plan of an SQL statement that violated a threshold defined in workload management using the *EXPLAIN from Activity* function
- SAP GUI version only:
  Download of the execution plan in `db2exfmt` format

For more information about the `EXPLAIN` function, see the DBA Cockpit documentation under References [page 228] in the appendix.

14.2.2 Index Advisor

If you want to investigate a poorly performing SQL statement using the `EXPLAIN` function, a frequently used method to improve the runtime is to use one or more additional indexes on the tables involved in this statement.

Creating new indexes for test purposes includes costly disk space consumption and usually an update of table and index statistics. Using the index advisor, you can test this scenario in the `EXPLAIN` pretending that the indexes to be tested already exist.

> **i Note**
>
> The statistics of recommended and user-defined indexes are estimated. The estimation is based on current statistics that are available for the currently existing tables and indexes. Therefore, after the advised indexes have been created, the optimizer access plan might differ from the one that was based on virtual indexes.

You can access the index advisor by choosing *Diagnostics* > *Index Advisor* on the *Database* tab page (Web Dynpro) or in the navigation frame (SAP GUI) of the DBA Cockpit. On the *Index Advisor* screen, you can, for example:

- Recommend indexes for a given SQL statement by choosing *Recommend Indexes*
- Add user-defined virtual indexes by choosing *Add Virtual Index*

For a new `EXPLAIN` run, you can now use existing indexes only, existing and recommended indexes, or existing, recommended, and user-defined indexes.

To check which optimizer access plan is better, compare the results.

> **⚠ Caution**
>
> The index advisor sometimes recommends adding *include columns* to existing unique indexes. To create an index with *include columns*, use storage parameter functions included in SAP transaction `SE14` (*ABAP Dictionary - Database Utility*).
>
> Make sure that you are logged on to the SAP database as the database connect user. Otherwise, the SAP system cannot use this particular index.

For more information about the index advisor, see the DBA Cockpit documentation under References [page 228] in the appendix.

14.2.3 Analysis of Lock-Wait Events

As of Enhancement Package 2 for SAP NetWeaver 7.0 SP7, you can use the lock-wait event monitor in the DBA Cockpit to analyze locks and deadlocks. On a detailed level, the lock-wait event monitor traces locks

and deadlock situations that occur on your system. In the event of lock timeouts, very long-lasting locks, or deadlocks , lock-related context information is dumped into the appropriate event monitor so that you can analyze it at a later point in time.

A database deadlock causes an error in processing. Therefore, this incident is usually reported in the SAP system log (`SYSLOG`) and in the SAP developer trace file of the victim process.

> → Recommendation
>
> We recommend that you also check the `db2diag.log` for related entries at the time of the error.

You can access the lock-wait event monitor by calling the DBA Cockpit and, on the *Database* tab page, choosing *Diagnostics* > *Lock-Wait Events*.

For more information, see the DBA Cockpit documentation under References [page 228] in the appendix .

14.3 SQL Trace for SAP ABAP and Java Systems

The SAP system provides tracing tools that you can use to trace SQL commands executed by the SAP system. This can be useful to examine slow programs in the SAP system.

> i Note
>
> Do not use the SQL traces to examine general performance issues of your SAP system. Since the traces can have a negative impact on the system performance, you should turn them off after you have reproduced the issue.

The following are the most important and easiest-to-use tracing options that are available in the SAP system:

- SQL Trace for SAP ABAP Systems (Transaction ST05) [page 170]
- SQL Trace for SAP Java Systems [page 171] (Open SQL Monitoring Web application)

14.3.1 SQL Trace for SAP ABAP Systems (Transaction ST05)

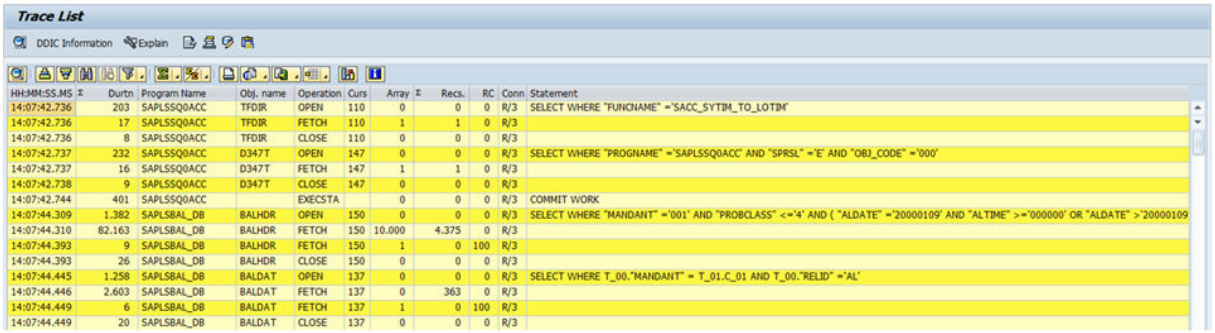
Use

With transaction `ST05`, you can trace SQL statements that are used in ABAP programs. You can turn the trace on for a certain user, transaction, program, or process in your system.

The trace output provides the following information and options for single SQL statements:

- If the SQL statement was written in OPEN SQL, it's converted to a standard SQL statement.
- If the statement is using parameter markers, the values are displayed.
- Based on the trace output, you can call the EXPLAIN function in the DBA Cockpit for a single statement.
- You can go to the ABAP coding where the SQL statement was executed.

Example



Trace List

DDIC Information | Explain

HH:MM:SS.MS	Durtn	Program Name	Obj. name	Operation	Curs	Array	Recs.	RC	Conn	Statement
14:07:42.736	203	SAPLSSQ0ACC	TFDIR	OPEN	110	0	0	0	R/3	SELECT WHERE "FUNCNAME" ='SACC_SYTIM_TO_LOTIM'
14:07:42.736	17	SAPLSSQ0ACC	TFDIR	FETCH	110	1	1	0	R/3	
14:07:42.736	8	SAPLSSQ0ACC	TFDIR	CLOSE	110	0	0	0	R/3	
14:07:42.737	232	SAPLSSQ0ACC	D347T	OPEN	147	0	0	0	R/3	SELECT WHERE "PROGNAME" ='SAPLSSQ0ACC' AND "SPRSL" ='E' AND "OBJ_CODE" ='000'
14:07:42.737	16	SAPLSSQ0ACC	D347T	FETCH	147	1	1	0	R/3	
14:07:42.738	9	SAPLSSQ0ACC	D347T	CLOSE	147	0	0	0	R/3	
14:07:42.744	401	SAPLSSQ0ACC		EXECSTA		0	0	0	R/3	COMMIT WORK
14:07:44.309	1.382	SAPLSBAL_DB	BALHDR	OPEN	150	0	0	0	R/3	SELECT WHERE "MANDANT" ='001' AND "PROBCLASS" <='4' AND ("ALDATE" ='20000109' AND "ALTIME" >='000000' OR "ALDATE" >'20000109
14:07:44.310	82.163	SAPLSBAL_DB	BALHDR	FETCH	150	10.000	4.375	0	R/3	
14:07:44.393	9	SAPLSBAL_DB	BALHDR	FETCH	150	1	0	100	R/3	
14:07:44.393	26	SAPLSBAL_DB	BALHDR	CLOSE	150	0	0	0	R/3	
14:07:44.445	1.258	SAPLSBAL_DB	BALDAT	OPEN	137	0	0	0	R/3	SELECT WHERE T_00."MANDANT" = T_01.C_01 AND T_00."RELID" ='AL'
14:07:44.446	2.603	SAPLSBAL_DB	BALDAT	FETCH	137	0	363	0	R/3	
14:07:44.449	6	SAPLSBAL_DB	BALDAT	FETCH	137	1	0	100	R/3	
14:07:44.449	20	SAPLSBAL_DB	BALDAT	CLOSE	137	0	0	0	R/3	

Example of ABAP Trace File (ST05)

More Information

- EXPLAIN Function [page 168]
- Performance Trace Overview on SAP Help Portal (for example, for SAP NetWeaver 7.4)

14.3.2 SQL Trace for SAP Java Systems

Use

You can access the SQL trace for SAP Java systems by using the Open SQL Monitoring Web application. The trace can be switched on and off dynamically. You can set different filters for the SQL trace. For example, you can filter by user, by HTTP session ID, by the minimum duration of an SQL statement, or by a particular application.

Example

SQLTrace Evaluation: List for trace id 20150109144520867 from node 2737750.

Filter | List of Traces | Trace Status | Refresh

Running on node 2737750. Output limited to 5000 records in total.

Time	Duration [microsec]	DB Session	J2EE user Application	Jdbc method Id	No.	Result	Statement
14:45:24,149	29		Administrator sap.com/tc~lm~itsam~ui~mainframe~wd	PooledConnection.setTransactionIsolation(int level)			setTransactionIsolation(1)
14:45:24,161	6		Administrator sap.com/tc~lm~webadmin~mainframe~persist~ear	PooledConnection.setAutoCommit(boolean bAutoCommit)			setAutoCommit(false)
14:45:24,162	455	10.76.201.220.33726.15010913355	Administrator sap.com/tc~lm~webadmin~mainframe~persist~ear	PreparedStatement.executeQuery() STMT:1243217256, RS:1685210698			SELECT "F"."ID" "F_ID_id","F"."USERID" "F_USERID_user1d","F"."APPLICATION_ID" "F_APPLICATION_ID_appl...
14:45:24,162	8	10.76.201.220.33726.15010913355	Administrator sap.com/tc~lm~webadmin~mainframe~persist~ear	ResultSet.next() RS:1685210698		false	next()
14:45:24,162	4	10.76.201.220.33726.15010913355	Administrator sap.com/tc~lm~webadmin~mainframe~persist~ear	ResultSet.close() RS:1685210698			close()
14:45:24,163	5	10.76.201.220.33726.15010913355	Administrator sap.com/tc~lm~webadmin~mainframe~persist~ear	PreparedStatement.clearBatch() STMT:1243217256			clearBatch()
14:45:24,163	703	10.76.201.220.33726.15010913355	Administrator sap.com/tc~lm~itsam~ui~mainframe~wd	Connection.commit()			commit()
14:45:24,177	284	10.76.201.220.33726.15010913355	Administrator sap.com/tc~lm~webadmin~mainframe~persist~ear	PreparedStatement.executeQuery() STMT:779375176, RS:1671115602			SELECT "P"."PID" "P_PID_pid","P"."PAID" "P_PAID_paid","P"."EAPPLID" "P_EAPPLID_eapplid","P"."PKEY" "...
14:45:24,177	182	10.76.201.220.33726.15010913355	Administrator sap.com/tc~lm~webadmin~mainframe~persist~ear	ResultSet.next() RS:1671115602		false	next()
14:45:24,178	2	10.76.201.220.33726.15010913355	Administrator sap.com/tc~lm~webadmin~mainframe~persist~ear	ResultSet.close() RS:1671115602			close()
14:45:24,178	2	10.76.201.220.33726.15010913355	Administrator sap.com/tc~lm~webadmin~mainframe~persist~ear	PreparedStatement.clearBatch() STMT:779375176			clearBatch()
14:45:24,178	119	10.76.201.220.33726.15010913355	Administrator sap.com/tc~lm~itsam~ui~mainframe~wd	Connection.commit()			commit()

Example of SQL Trace for SAP Java Systems

More Information

SQL Trace on SAP Help Portal

14.4 Tracing the SAP Database Interface (DBSL)

14.4.1 What Is the DBSL?

The lowest software layer between SAP kernel and the database is the database service layer, also called DBSL.

Since this layer is not database vendor-independent, there is a separate DBSL shared library for each supported database platform that can be loaded by the SAP kernel when needed.

These libraries are named `db<dbs>slib` with an operating system-dependent suffix. For Db2 for Linux, UNIX, and Windows, the library name is `dbdb6slib`.

> **i** Note
>
> The SAP kernel program that loads the DBSL shared library must have the same kernel version as the DBSL shared library itself.

What Does the DBSL Do?

The tasks performed by the DBSL for Db2 include, for example, the following:

- Loading of the database client libraries
- Maintenance of database connections

- Final translation of ABAP SQL statements into a database vendor-specific SQL format
- SQL error handling
- Tracing of SQL statements
- Collection of network statistics
- Handling of virtual tables

DBSL Trace

The DBSL for Db2 provides specific traces that allow you to analyze problems. That's why in the event of a customer incident, SAP support staff may instruct you to activate those traces and to provide the trace output for analysis.

The DBSL trace sequentially dumps all activities that are passed from the SAP applications to the Db2 database using the Db2 call level interface (CLI). Since a sequential trace in a production ABAP system can easily become very large, you can set a filter to minimize the trace output. You can trace, for example, the SQL activity on certain tables only, which often makes the DBSL trace easier to use than a Db2 CLI trace. In an SAP system, the DBSL trace can be dynamically activated.

DBSL Deadlock Trace

The DBSL deadlock trace is designed to trace the SQL activity of deadlock participants (assuming that all those participants are SAP programs). To simplify the interpretation of the DBSL deadlock trace output, you need to collect `db6util -sl` information while the deadlock trace is running.

> **i** Note
>
> The DBSL trace tools mentioned here only apply to SAP ABAP systems. For SAP Java systems, you must activate the JDBC trace (for more information, see JDBC Trace [page 188]).

More Information

- Activating the DBSL Trace [page 174]
- DBSL Deadlock Trace [page 180]
- SAP Note 31707: *DB6: DBSL trace for performance/error analysis*
- SAP Note 175036: *DB6: The DBSL deadlock trace*
- SAP Note 327595: *Analysis of database lock situations using db6util*

14.4.2 DBSL Trace

14.4.2.1 Activating the DBSL Trace

Use

You can activate the DBSL trace for the following:

- Selected SAP work processes
- All SAP work processes of an SAP application server
- All SAP work processes of an SAP logon session

Procedure

Activating the DBSL Trace for Selected SAP Work Processes

1. On the application server where the SAP work process to be traced is running, call transaction SM50.
2. Select the SAP work processes that you want to trace.
3. Choose *Process* > *Trace* > *Components*.
   The *Change Trace Components* dialog box appears.
4. Use trace level 2 or 3 for component *Database (DBSL)*.
5. Deselect the other components.
   The trace information is written to the SAP developer trace files.

> **i** Note
>
> To deactivate the DBSL trace, repeat steps 1 to 3. On the *Change Trace Components* dialog box, choose *Default Values*.

Activating the DBSL Trace for All SAP Work Processes of an SAP Application Server

You can activate the DBSL trace for all SAP work processes of an SAP application server by setting the appropriate parameters in the SAP instance profile `dbs/db6`.

The following table lists the supported parameters and settings:

SAP Instance Profile Parameter	Description
`dbs/db6/dbsl_trace = <tracelevel>`	The following values are possible: • *0* or *1* The DBSL trace is turned off. This is the SAP default value. If an SQL statement returns an error, it is written to the SAP developer trace file of the SAP work process together with the affected SQL statement. • *2* Trace level 2 writes the SQL statements that the SAP application sends to the database. The input parameters that are passed to or retrieved from the database are **not** dumped. If you require this information to investigate a certain problem, use trace level 3. • *3* Trace level 3 writes the SQL statements, the parameters for the parameter makers as well as the values returned by the database into the trace file.
`dbs/db6/dbsl_trace_dir = <trace directory>`	Specifies an alternative trace directory. The default value is as follows: • **Linux and UNIX:** `/tmp/TraceFiles` • **Windows:** `drive>:\usr\sap\TraceFiles` User `<sapsid>adm` must have write access to the trace directory. Trace data is written to files (for example, `TraceFile <pid>.txt`) that are located in a subdirectory of the trace directory. This subdirectory carries the name of the SAP system. If the subdirectory does not exist yet, it is created as soon as the trace becomes active.
`dbs/db6/dbsl_trace_flush = <value>`	The default value is *0*. If `dbsl_trace_flush` is set to a value other than *0*, the trace output is flushed to disk after each trace operation. Flushing the trace on every write is expensive and has a significant negative impact on trace performance. Only use this parameter when advised by SAP support.
`dbs/db6/dbsl_trace_string = <string1>[;<string2>[…]]`	Restricts the trace to SQL statements that contain the specified search strings.

SAP Instance Profile Parameter	Description
`dbs/db6/dbsl_trace_iocount = <count>`	With array operations, the trace displays only the first number of operations defined by `<count>`. The default value is *5*.
`dbs/db6/dbsl_trace_time = <runtime>`	Only operations with a runtime of at least `<runtime>` milliseconds are written to the trace files.
`dbs/db6/dbsl_trace_str_len = <maximum trace column length>`	By default, the content of the data fields is displayed up to a maximum length of 64 bytes. To change the maximum trace column length, use this parameter.

With transaction RZ11, you can change the DBSL trace settings dynamically (that is, without restarting the application server) by changing the values of the parameters listed in the table above. By default, the changes are **only** valid for the current SAP application server. Optionally, you can turn on the trace for all application servers at a time.

The changes are **not** saved in the SAP instance profiles and are lost after you restart the SAP application servers.

Activating the DBSL Trace for All Work Processes of an SAP Logon Session

The previously described methods to trace the DBSL only work for SAP work processes. However, the DBSL is also used by SAP programs, for example, `R3trans`, `R3load` or `tp`, which cannot be influenced using transaction SM50 or SAP instance profile parameters.

To trace these programs, activate the DBSL trace using the environment variable `DB6_DBSL_TRACE`.

You can set the environment variable as follows:

- In the terminal window where you start the program to be traced
- In the user profile of user `<sapsid>adm`
- On Windows, in the registry under the key
  `HKEY_LOCAL_MACHINE\Software\SAP\<SAPSID>\Environment`
  Enter trace environment variables as `STRING_VALUE`.

> **i** Note
>
> To activate the trace, you have to restart the application server.

The following options of `DB6_DBSL_TRACE` are supported:

- `DB6_DBSL_TRACE`
- `DB6_DBSL_TRACE_DIR`
- `DB6_DBSL_TRACE_FLUSH`
- `DB6_DBSL_TRACE_STRING`
- `DB6_DBSL_TRACE_IOCOUNT`
- `DB6_DBSL_TRACE_TIME`
- `DB6_DBSL_TRACE_STR_LEN`

For a description of these parameters, see the table above in section *Activating the DBSL Trace for All SAP Work Processes of an SAP Application Server*.

To test whether the trace is active, enter the following command:

```
R3trans -d
```

14.4.2.2 Trace File Format

The following section provides information about and examples of the type of data that is included in a trace file and what the various areas of a trace file can look like.

Trace data is usually displayed in table columns. The following table lists the column types that are contained in a trace file, depending on the area:

Column	Description
`CON`	Database connection that a statement is using
`Hstmt`	CLI statement handle (reference throughout the trace)
`c_id`	Cache ID of the statement in the DBSL cache
`Statement`	Statement text and parameters set for a statement See *Example 3 - Current Open Connections and Statement Cache of the DBSL* below
`toplevel caller`	Topmost DBSL function that calls the CLI layer
`CLI function`	Indicates if the CLI function was called
`sql_rc`	SQL return code that is retrieved by the CLI function
`rows`	Lists the number of rows affected
`additional info`	Provides additional information
`CL`	Classifies the statement runtime
`timestamp`	Time stamp when the CLI function was called
`elapsed time`	Elapsed time since the CLI function was called If the value for elapsed time is high in the sense of the *classification* as shown below in *Example 2 – Legend*, the trace writes the appropriate classification token (`1!,…,5!`) into the *CL* field.

Trace File Areas

The following examples show what various areas of a trace file can look like and how they are related.

Example 1 - Trace File Parameter Settings

This example shows the area where you can find the configuration settings for the DBSL trace:

```
&
& DB2 UDB DbSl trace
& ------------------
&
& PID 13872 on fmhsuse11, system N71
&
& DBSL_TRACE          = 2
& DBSL_TRACE_DIR      = /tmp/TraceFiles/N71
& DBSL_TRACE_FLUSH    = 0
& DBSL_TRACE_STRING   =
& DBSL_TRACE_NWSTATS  = 0
& DBSL_TRACE_IOCOUNT  = 5
& DBSL_TRACE_TIME     = 0 micro seconds
& DBSL_TRACE_STR_LEN  = 64
&
```

Example 1 - Trace File Parameter Settings

Example 2 - Legend

The following area of the trace file contains a description of most of the columns and fields that contain trace information from the trace file:

```
&
& LEGEND:
& -------
&
& CON   : connection handle
&
& hstmt : CLI statement handle
&
& c_id  : cache_id
&
& rows  : rows affected ( or 0 if n/a)
&
& timestamp: time of execution in format DD.MM hh:mm:ss.usec
&
& elapsed time: duration of execution in format mmm:ss.usec
&
& CL Classification of the elapsed time:
&       1!    50,000usec <= elapsed time <    125,000usec
&       2!   125,000usec <= elapsed time <    250,000usec
&       3!   250,000usec <= elapsed time <    500,000usec
&       4!   500,000usec <= elapsed time <  1,000,000usec
&       5!                 elapsed time >= 1,000,000usec
&
```

Example 2 - Legend

Example 3 - Current Open Connections and Statement Cache of the DBSL

> **i** Note
>
> This area only exists in the trace file if a trace was turned on dynamically.

```
& List of open connections:
&
&   CON =  0, DB = N71, USER = SAPN71, SCHEMA = SAPN71
&   CON =  1, DB = N71, USER = SAPN71, SCHEMA = SAPN71
&   CON =  2, DB = N71, USER = SAPN71, SCHEMA = SAPN71
&
& dumping statement caches for all connections ...
```

Open Connections and Statement Cache of the DBSL

The statements are listed together with a statement handle. The trace data that is dumped afterwards (see the next figure) only refers to this statement handle - except when a new statement is prepared or the statement execution fails with an error.

The statement cache is dumped in the columns `CON`, `hstmt`, `c_id`, and `Statement`:

```
& dumping statement caches for all connections ...
&
& CON| hstmt |c_id|Statement
& ---+------+----+-------------------------------------------------------------------------------------------------------------------------------
&+  0| 1:1   |   0|     SELECT * FROM "TBTCP" WHERE "JOBNAME" = ? AND "JOBCOUNT" = ?  WITH CS
&+  0| 1:1   |   0|       cursor type=NO_HOLD, isolation=RC, cc_release=YES, optlevel=5, degree=1, op_type=18, read_only=1
&+  0| 1:1   |   0|       ABAP location info: 'SAPMSSY2', 7389
&+  0| 1:2   |   1|     INSERT INTO "TBTCO" VALUES( ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
&+  0| 1:2   |   1|      ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ? )
&
& ... end of statement cache dump
```

Example 3 - Part II

Example 4 – General Trace Data

The following example shows an excerpt of general trace data:

```
& ... end of statement cache dump
&
&
& CON| hstmt |c_id|toplevel caller     |CLI function        |sql_rc|rows   |additional info        |CL| timestamp           |elapsed time
& ---+------+----+-------------------+--------------------+------+-------+-----------------------+--+---------------------+------------
&   0| 1:61  |  60|DbSlModifyDB6       |SQLFreeHandle       |     0|      0|SQL_HANDLE_STMT        |  |11.12 16:35:36.746489|000:00.000049
&   0| 1:61  |  60|DbSlModifyDB6       |SQLAllocHandle      |     0|      0|SQL_HANDLE_STMT        |  |11.12 16:35:38.073712|000:00.000019
&   0| 1:61  |  60|DbSlModifyDB6       |SQLExtendedPrepare  |     0|      0|                       |  |11.12 16:35:38.073819|000:00.034180
&+  0| 1:61  |  60|     UPDATE "TPFYPROPTY" SET "SUSR" = ?, "SDATE" = ?, "STIME" = ? WHERE "PARANAME" = ?
&+  0| 1:61  |  60|       cursor type=NO_HOLD, isolation=UR, cc_release=YES, optlevel=5, degree=1, op_type=19, read_only=0
&+  0| 1:61  |  60|       ABAP location info: 'SAPLTHFB', 9666
&   0| 1:61  |  60|DbSlModifyDB6       |SQLExtendedBind     |     0|      0|BindParameter          |  |11.12 16:35:38.108128|000:00.000006
&   0| 1:61  |  60|DbSlModifyDB6       |SQLExecute          |     0|      1|                       |  |11.12 16:35:38.108200|000:00.000658
&   0| 1:61  |  60|DbSlModifyDB6       |SQLRowCount         |     0|      1|                       |  |11.12 16:35:38.119023|
```

Example 4 - General Trace Data

The header information describes the format of the first line of a single trace record. Subsequent lines are appended to columns `CON`, `hstmt`, and `c_id`. The first line of a trace record starts with `&`, while subsequent lines start with `&+`.

14.4.3 DBSL Deadlock Trace

In the case of a deadlock situation, the DBSL deadlock trace logs all active SQL statements of the last database transaction of every work process into a file. This means that all SQL statements of the current database unit of work (UOW) of the processes that are involved in a deadlock situation are dumped. The dump files allow you to analyze the order in which the SQL statements are submitted to the database by the processes involved.

The DBSL deadlock trace logs the SQL statements of the following applications:

- Applications that receive an SQL911E error
- Applications that use SQL statements exceeding the time that was specified by SAP profile parameter `dbs/db6/dbsl_trace_deadlock_time` (in seconds) or by environment variable `DBS_DB6_DBSL_TRACE_DEADLOCK_TIME`.
  As a result, there is a high probability that all database transactions participating in a deadlock situation are captured.

The following sections provide information about how you activate and use the DBSL deadlock trace as well as examples of trace files.

14.4.3.1 Activating the DBSL Deadlock Trace

You can activate the DBSL deadlock trace using one of the following options:

- Parameter `dbs/db6/dbsl_trace_deadlock_time`
  To activate the deadlock trace, set parameter `dbs/db6/dbsl_trace_deadlock_time` in the SAP instance profile to a value that is higher than *0*. You can set this parameter directly in the SAP instance profile or dynamically using transaction RZ11.
  The parameter specifies the threshold for the runtime of a statement. This runtime determines if a transaction is to be traced. The value of `dbs/db6/dbsl_trace_deadlock_time` must be large enough so that only statements running for a long time are written to the log file. However, the value must be considerably lower than the one of the database configuration parameter `DLCHKTIME`. Parameter `DLCHKTIME` defines the frequency at which the database manager checks for deadlocks among all the applications that are connected to a database. By default, the value is set to *10000*, which is 10 seconds. For tracing purposes, this interval is too short. We recommend that you increase the value of `DLCHKTIME` so that the time interval is extended to a range of, for example, a few minutes. Since `DLCHKTIME` is a dynamic parameter, the changes become effective immediately.

  > **Example**
  >
  > To get a time interval of 5 minutes, set `DLCHKTIME` to the appropriate value in milliseconds (that is, 300000) as follows:
  >
  > ```
  > db2 update db cfg for db <dbsid> using dlchktime 300000 immediate
  > ```
  >
  > A good value for `dbs/db6/dbsl_trace_deadlock_time` corresponding to this setting is *20*.

- Environment variable `DBS_DB6_DBSL_TRACE_DEADLOCK_TIME`
  You use this environment variable if you want to trace other programs, for example, `R3trans` (which cannot be influenced by transaction SM50 or the SAP instance profile). To activate the trace, set it to a value higher than *0*. This value specifies the time an SQL statement has to run so that a trace file is generated. To deactivate the trace, set environment variable `DBS_DB6_DBSL_TRACE_DEADLOCK_TIME` to *0* or do not set it at all.

14.4.3.1.1 Examples of DBSL Deadlock Trace Files

The following is an example of two deadlock traces for two different work processes:

```
& CON| hstmt |c_id|toplevel caller     |CLI function        |sql_rc|rows   |additional info        |CL| timestamp
|elapsed time
& ---+-------+----+--------------------+--------------------+------+-------+-----------------------+--+-------------
-------+-------------
&   0| 1:22  |  21|DbSlReadDB6         |SQLExtendedPrepare  |     0|      0|                       |  |11.10
10:13:32.107869|000:00.028919
&+  0| 1:22  |  21|     SELECT * FROM "ZJOTEST1" WHERE "C1" = ?  OPTIMIZE FOR 1 ROWS WITH RS USE AND KEEP EXCLUSIVE
LOCKS
&+  0| 1:22  |  21|       cursor type=NO_HOLD, isolation=RR, cc_release=NO, optlevel=5, degree=1, buffer_lobs=YES,
op_type=29, reopt=0
&+  0| 1:22  |  21|       ABAP location info: 'ZJOLOCK1', 77
&   0| 1:22  |  21|DbSlReadDB6         |SQLExecute          |     0|      1|                       |  |11.10
10:13:32.136964|000:00.000211
&+  0| 1:22  |  21|     row 1:             1 LONG            I=4        2
&   0| 1:22  |  21|DbSlReadDB6         |SQLFreeStmt         |     0|      0|SQL_CLOSE              |  |11.10
10:13:32.142473|000:00.000141
&   0| 1:24  |  23|DbSlModifyDB6       |SQLExtendedPrepare  |     0|      0|                       |  |11.10
10:13:38.000854|000:00.003017
&+  0| 1:24  |  23|     UPDATE "ZJOTEST2" SET "C2" = ? WHERE "C1" = ?
&+  0| 1:24  |  23|       cursor type=NO_HOLD, isolation=UR, cc_release=NO, optlevel=5, degree=1, buffer_lobs=NO,
op_type=19, reopt=0
&+  0| 1:24  |  23|       ABAP location info: 'ZJOLOCK1', 102
&   0| 1:24  |  23|DbSlModifyDB6       |SQLExecute          |    -1|      1|                       |5!|11.10
10:13:38.004027|000:07.144983
&+  0| 1:24  |  23|     row 1: [ERR]       1 WCHAR           I=28      "deadlock test1"
&+  0| 1:24  |  23|                        2 LONG            I=4        9
```

Trace 1

```
& CON| hstmt |c_id|toplevel caller     |CLI function        |sql_rc|rows   |additional info        |CL|
timestamp           |elapsed time

& ---+-------+----+--------------------+--------------------+------+-------+-----------------------+--+---------
-----------+-------------

&   0| 1:154 | 153|DbSlReadDB6         |SQLExtendedPrepare  |     0|      0|                       |  |11.10
10:13:30.841549|000:00.011026

&+  0| 1:154 | 153|     SELECT * FROM "ZJOTEST2" WHERE "C1" = ?  OPTIMIZE FOR 1 ROWS WITH RS USE AND KEEP
EXCLUSIVE LOCKS

&+  0| 1:154 | 153|       cursor type=NO_HOLD, isolation=RR, cc_release=NO, optlevel=5, degree=1,
buffer_lobs=YES, op_type=29, reopt=0

&+  0| 1:154 | 153|       ABAP location info: 'ZJOLOCK2', 77

&   0| 1:154 | 153|DbSlReadDB6         |SQLExecute          |     0|      1|                       |  |11.10
10:13:30.852770|000:00.000136

&+  0| 1:154 | 153|     row 1:             1 LONG            I=4        9

&   0| 1:154 | 153|DbSlReadDB6         |SQLFreeStmt         |     0|      0|SQL_CLOSE              |  |11.10
10:13:30.853495|000:00.000433

&   0| 1:226 | 225|DbSlModifyDB6       |SQLExtendedPrepare  |     0|      0|                       |  |11.10
10:13:36.002837|000:00.010660

&+  0| 1:226 | 225|     UPDATE "ZJOTEST1" SET "C2" = ? WHERE "C1" = ?

&+  0| 1:226 | 225|       cursor type=NO_HOLD, isolation=UR, cc_release=NO, optlevel=5, degree=1,
buffer_lobs=YES, op_type=19, reopt=0

&+  0| 1:226 | 225|       ABAP location info: 'ZJOLOCK2', 102

&   0| 1:226 | 225|DbSlModifyDB6       |SQLExecute          |     0|      1|                       |5!|11.10
10:13:36.013684|000:09.151306

&+  0| 1:226 | 225|     row 1:             1 WCHAR           I=28      "deadlock test2"

&+  0| 1:226 | 225|                        2 LONG            I=4        2
```

Trace 2

The deadlock shown here occurs because both applications first perform a SELECT SINGLE FOR UPDATE operation on two different tables, which results in an exclusive lock on the selected rows. Afterwards, each application tries to update the row in the table on which the other application is already holding a lock.

14.4.3.2 Using the DBSL Deadlock Trace Together With the db6util Tool

The DBSL deadlock trace does not show which processes are waiting for each other. Since deadlocks can be accompanied by a general contention of resources on the database, this can result in a very high number of work processes being in a lock wait or deadlock situation. Without additional information, you might not be able to identify the real deadlock with only the data provided by the DBSL deadlock trace.

To be able to retrieve the missing data, we therefore recommend that you run the `db6util` tool together with the DBSL deadlock trace. You execute it on operating system level, which is particularly useful if you cannot log on to the SAP system (for example, due to lock contention).

`db6util` runs application snapshots at regular intervals and checks them for lock-wait situations and deadlocks. `db6util` provides you with the IDs of the processes involved in the deadlock. The process IDs also point to the trace files because the process IDs are included in the trace file names.

`db6util` can run for days with a moderate amount of space required for logging and a minimum impact on system performance.

Constraints

For the use of `db6util`, the following constraints apply:

- Since `db6util` checks the system at regular time intervals, it might possibly miss deadlocks. The same is true for the DBSL Deadlock Trace.
- The tracking mechanism via Db2 application snapshots implies that the database monitors of the system need to be turned on.
- `db6util` only shows the last statement executed by the processes involved. The transaction history is not included in the snapshot data.
- By default, SAP uses parameter markers for non-BW SQL statements. In its output, `db6util` shows the prepared statement with parameter markers. Information about actual values is not tracked.

14.5 Db2 Traces

The Db2 software provides the following trace facilities to track a variety of problems that might occur when you run or develop Db2 applications:

- Db2 Trace Facility db2trc [page 185]
- Db2 CLI Trace [page 186]
- JDBC Trace [page 188]

Before you run a trace, you must consider the following:

- In most cases, system performance decreases.
- Running a trace is a time-consuming task and requires careful planning.
- The amount of trace data can be very large.
  Therefore, use the smallest possible scenario that reproduces the problem. You might have to minimize or even stop other activities while you re-create the problem situation with the trace activated.
- Interpreting trace data properly requires detailed inside knowledge of Db2. Therefore, traces are usually only run at the request of Db2 or SAP support teams.
- Problems that were observed might disappear during attempts to reproduce the problem situation with the trace turned on due to different timing conditions.

More Information

- Basic trace diagnostics in the IBM documentation
- **SAP Note** 38513: *Database trace (db2trc) with DB malfunctions*

- **SAP Note** 1486120: *DB6: Data Collection in a CLI driver environment*
- **SAP Note** 850486: *DB6 and DB2/390: Java Database Connectivity (JDBC) trace*

14.5.1 Db2 Trace Facility db2trc

The Db2 trace is controlled by the `db2trc` command. You must run `db2trc` on the database server as the instance owner. To transform the trace data into a readable form, you have to format the trace data on the database system where the trace data was collected.

To get an overview of the available options, run the trace facility without any options.

Location of Trace Data

When the trace is activated, options for the location of trace data are:

- Shared memory
  The amount of memory that is available for tracing is limited by the amount of available shared memory. The advantage is that having a limited memory has the least negative impact on system performance. The disadvantage is that depending on the amount of shared memory, either the trace periodically overwrites itself (including the relevant parts of the trace data) or the trace ends after the available memory area has been filled.

  > **i** Note
  >
  > The available amount of memory is usually not large enough for the purpose of tracing. Therefore, tracing in shared memory is mainly used when Db2 performance is traced as described later in this section.

  Once the problem has been traced, trace data first needs to be dumped before the trace is turned off.
- Disk
  Tracing on disk decreases the database system performance a lot more than tracing in the shared memory. The possible amount of trace data is limited by the available disk space. Usually, support teams ask for a Db2 trace on disk.

Running db2trc on Disk

1. To activate the trace on the database server and to write the trace data in the file `db2trc.dmp`, enter the following command as the instance owner:
   ```
   db2trc on -f db2trc.dmp
   ```
2. Reproduce the problem.
3. Turn off the trace by entering the following command:
   ```
   db2trc off
   ```

4. Split `db2trc.dmp` into the files `db2trc.fmt` and `db2trc.flw` and format them by entering the following command:
   ```
   db2trc fmt db2trc.dmp db2trc.fmt
   db2trc flw db2trc.dmp db2trc.flw
   ```
   You can now analyze the trace data in the `db2trc.fmt` and `db2trc.flw` trace files.

Tracing Db2 Performance Using db2trc

1. To activate the trace, enter the following command:
   ```
   db2trc on -perfcount
   ```
2. Since the trace writes to the memory, you need to explicitly dump the data to disk before turning the trace off again. To do so, enter the following commands:
   ```
   db2trc dmp db2trc.perfdmp
   db2trc off
   ```

   > **i** Note
   >
   > If you turn off the trace before the data was dumped, the trace data is lost.
3. To format the trace data, enter the following command:
   ```
   db2trc perffmt db2trc.perfdmp db2trc.perffmt
   ```

   > **⚠** Caution
   >
   > You cannot use the performance trace in combination with any other option because then option **`-perfcount`** is ignored.

14.5.2 Db2 CLI Trace

The Db2 CLI trace traces all activities of the Call Level Interface (CLI), that is, the SQL interface of Db2 used by the SAP ABAP kernel. You run the Db2 CLI trace on the machine where applications that use the SAP database interface (DBSL) are running. The trace becomes active for an application process when the process connects to the database after the trace was turned on. It is also possible to activate the trace for a process when it is already connected to the database.

Configuring the Db2 CLI Trace

You can configure the Db2 CLI trace by editing the Db2 CLI configuration file `db2cli.ini` manually. The `db2cli.ini` file is located in the SAP system global directory under `.../global/db6/`:

> **✤** Example
>
> Linux/UNIX: `/usr/sap/<SID>/SYS/global/db6/db2cli.ini`
>
> Windows: `\\<SAPGLOBALHOST>\sapmnt\<SID>\SYS\global\db6\db2cli.ini`

In the `db2cli.ini` file, the actual configuration is done by editing or adding the section `[common]`. The format for section `[common]` is as follows:

```
[common]
<parameter_1>=<value_1>
<parameter_2>=<value_2>
```

> **Example**
>
> ```
> [common]
> Trace=1
> Traceflush=1
> Tracepathname=/tmp/CLItrace
> Tracerefreshinterval=60
> ```

The following configuration parameters are supported:

Parameter	Description
TRACE	If TRACE is set to `1`, the trace is activated. After the process that is being traced has finished, do not forget to deactivate the trace by setting **`TRACE=0`**. > **Note** > If TRACEREFRESHINTERVAL is not set, keep in mind that tracing continues until the application processes that are traced are disconnected.
TRACEFILENAME	Specifies the path of the file that contains all trace data
TRACEPATHNAME	Specifies the directory where trace files are stored > **→ Recommendation** > We strongly recommend that you have one trace file per application process. To this, set CLI parameter TRACEPATHNAME to the name of an existing directory (`TRACEPATHNAME=<directory>`).
TRACEFLUSH	If this parameter is set to `1`, it forces a write to disk for each entry. To avoid problems, for example, that trace data is not flushed to disk and is lost if an application process crashes, make sure that you set TRACEFLUSH to `1`.

Parameter	Description
TRACEREFRESHINTERVAL	Specifies the time in seconds after which the CLI configuration is reread by the CLI application. To activate this parameter, you must restart the applications. Therefore, the parameter allows a dynamic activation of the trace.

Running the Db2 CLI Trace

1. Log on as `<sapsid>adm` user.
2. To activate the trace, you set the trace parameter in the CLI configuration to 1 by maintaining section `[common]` of the `db2cli.ini` file so that it contains the following entry:
   `TRACE=1`
3. After the trace has been activated, you can check that it is active by entering the following command:
   `R3trans -x`
   If the trace is active, R3trans generates one trace file.

   > **i** Note
   >
   > The trace files are always named using the following pattern:
   >
   > `p<pid>t<tid>.cli`
   >
   > If the CLI trace is turned on dynamically, there are two files per `<pid>`. One contains the trace data, the other the statement cache.
4. Reproduce the problem.
5. To deactivate the trace, set the TRACE parameter in section `[common]` as follows:
   `TRACE=0`

14.5.3 JDBC Trace

Use

The JDBC trace traces the JDB interface that SAP Java applications use to access the database. To activate the trace, you can use one of the following methods:

- Connect URL
- Db2 JDBC global properties file

Activating the JDBC Trace Using the Connect URL

1. Open the SAP J2EE Configuration tool.
2. In the navigation frame, choose *Secure Store*.
   The connect URL looks as follows:
   `jdbc:db2://<host>:<port>/<DBSID>[:<parameter>=<value>;[...]]`

3. Add the following string to the URL:
   `traceDirectory=<directory>`
   `<directory>` must be an existing directory. The SAP system administrator (`<sapsid>adm`) must have write access to this. For each application process, one file is written to this directory. The following naming conventions are used for the file names: `_driver_<n>`, where `<n>` can be `1, 2, 3`…
4. To restrict the amount of output created, add the following string to the URL:
   `traceLevel=65`
   You must separate the specified parameter settings with a semicolon. In addition, make sure that the connect URL is terminated with a semicolon. Otherwise, the Java applications fail to connect to the database.

> **i Note**
>
> To activate or deactivate the JDBC trace using this method, you have to restart the AS Java.

> **Example**
>
> **`jdbc:db2://serverxy:5678/J01:traceDirectory=/tmp/JDBCTrace;traceLevel=-65;`**

Activating the JDBC Trace Using the Db2 JDBC Global Properties File

1. Create a properties file for the Db2 JDBC driver.

> **→ Recommendation**
>
> We recommend that you call the file `jcc.properties` and that you store it in the `/global/db6` directory.

2. Add the following lines to the `jcc.properties` file:
   `db2.jcc.traceDirectory=<path to trace directory>`
   `db2.jcc.traceLevel=0`
3. Make the properties file known to the Java environment using the SAP J2EE Configuration tool:
   1. In the navigation frame of the SAP J2EE Configuration tool, choose *Cluster_data* > *instance_<SAP Java instance>* > *server_<server number>*
   2. On the *General and Bootstrap* tab page, add the following parameter entry:
      `-Ddb2.jcc.propertiesFile=<path to global properties file>`
   3. Save your entries.
4. To activate the JDBC trace, restart the AS Java.
5. To dynamically activate the trace, modify the global properties file as follows:
   `db2jcc.tracelevel=<trace_level>`

> **→ Recommendation**
>
> We recommend that you use trace level 65.

> **i Note**
>
> To deactivate the trace, set the trace level to **`0`**.

More Information

SAP Note 850486: *DB6 and DB2/390: Java Database Connectivity (JDBC) trace*

15 SAP Tools

15.1 Graceful Maintenance Tool (GMT) for SAP Business Continuity During Database Maintenance

The Graceful Maintenance Tool (GMT) is based on the micro-outage feature of SAP on IBM Db2. You can use it to pause SAP applications for a short time, which enables you to execute short database maintenance tasks in a template-based and automated behavior. Using the GMT only takes a few minutes, and you do not have to stop any SAP NetWeaver ABAP application servers.

Related Information

SAP Micro-Outage Feature [page 191]
Graceful Maintenance Tool (GMT) [page 192]

15.1.1 SAP Micro-Outage Feature

The micro-outage feature is part of the SAP Database Shared Library (DBSL) for IBM Db2 for Linux, UNIX, and Windows. For information about how to use this feature, see SAP Note 1434153. This SAP Note describes all the steps that are required to activate, use, and deactivate the micro-outage feature. Once the micro-outage feature has been activated, all ABAP application servers disconnect and fall into sleep at the next transaction boundary. When all connections are closed, you can execute short maintenance tasks and afterwards deactivate the micro-outage feature. After the deactivation, the SAP application servers reconnect to the database and continue processing without any errors.

The micro-outage feature is only available for SAP NetWeaver ABAP servers. You can use it to decrease the number of canceled transactions during a planned maintenance. However, there is no guarantee that all connections can be closed until the maintenance task starts. Note the following restrictions when using the micro-outage feature:

- The maintenance task should not take longer than the time set by `rdisp/max_wprun_time` (the default is 300 seconds). If it takes longer, the dispatcher restarts dialog work processes.
- During the maintenance task, you must not update the Db2 client software if its libraries are loaded by an active SAP application server.
- The closing of the database connection always occurs only at a database transaction limit. Long-running database transactions, for example, background jobs or processes that are waiting for a enqueue lock, are not interrupted.
- Only the database connections of the local SAP system are closed. External connections, for example, from an SAP Solution Manager system, are not interrupted.

- Secondary database connections of the work processes are not closed if they have an open transaction.
- The micro-outage feature affects only the ABAP application server. Database connections from the Java stack are not interrupted.

Related Information

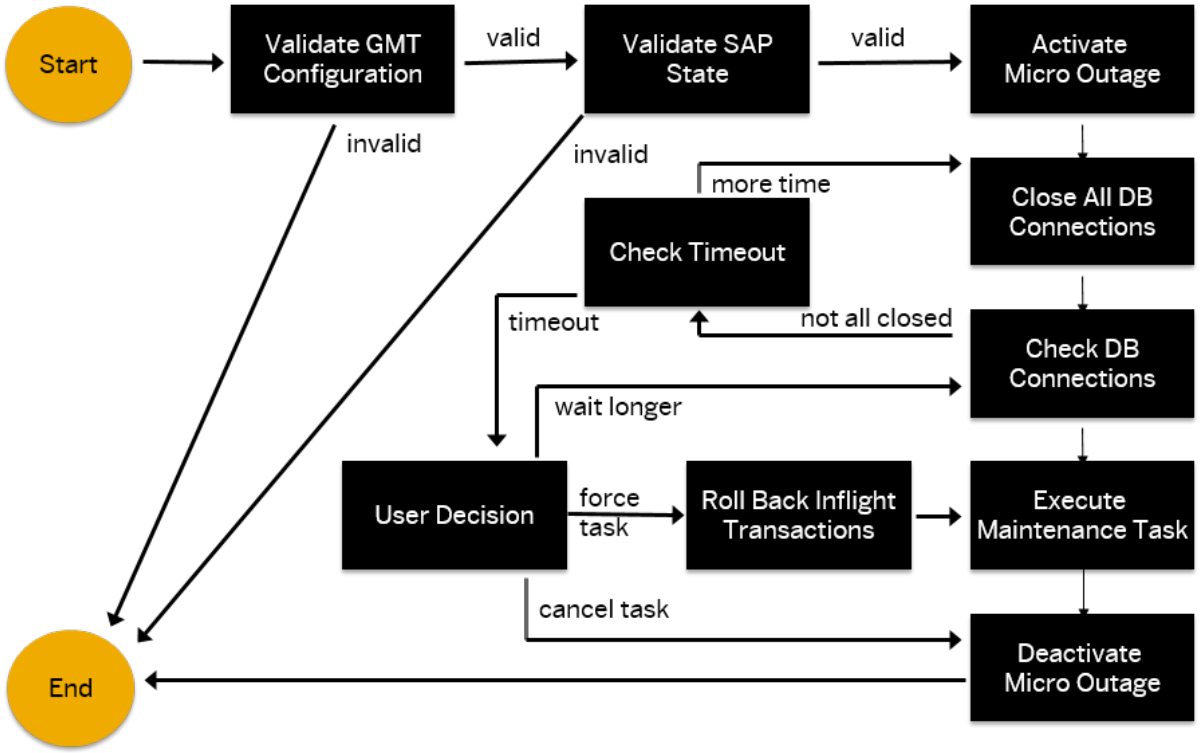
Graceful Maintenance Tool (GMT) [page 192]

15.1.2 Graceful Maintenance Tool (GMT)

The GMT enables you to make optimal use of the micro-outage feature. The GMT works like a wrapper around the micro-outage feature and takes care of prerequisites, constraints, and other special circumstances for the use of the micro-outage feature, such as identifying the exact workflow and the appropriate tasks. The GMT checks the state of the system and the prerequisites. If these checks do not return any errors, the GMT automatically activates the micro-outage feature.

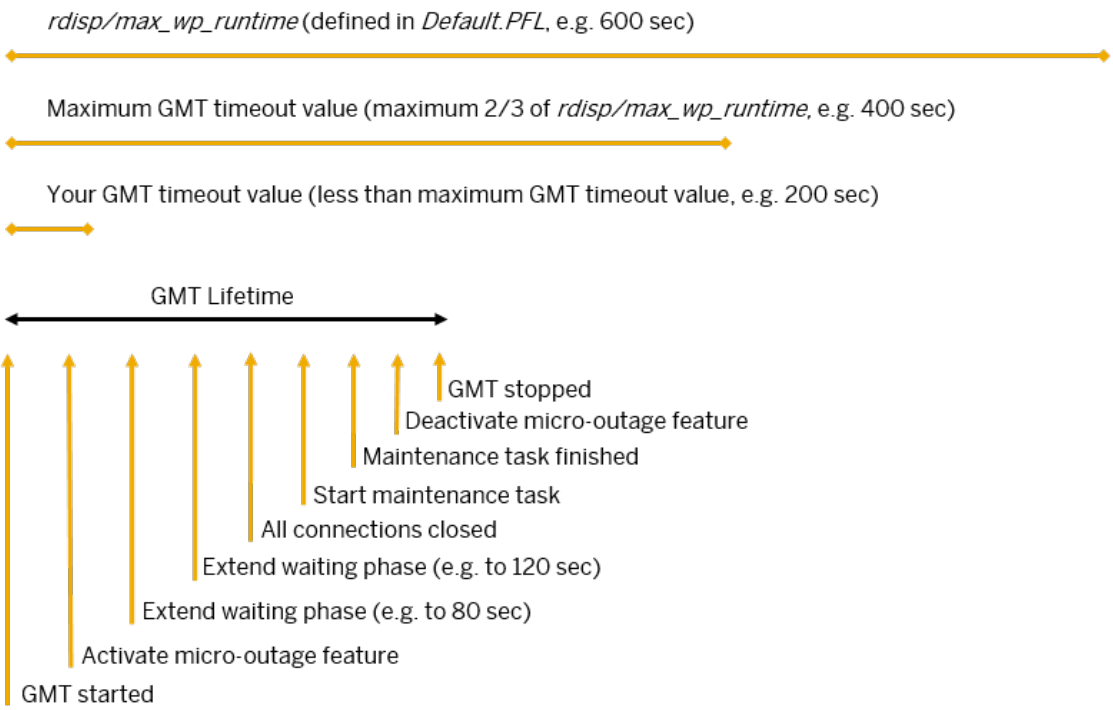
The GMT monitors the disconnection process and observes the maximum allowable maintenance time. If the micro-outage feature manages to close all connections in the defined time window, the GMT silently executes the maintenance task, logs all information, and, after the task has been finished, deactivates the micro-outage feature so that SAP users can continue their work.

The following figure shows the GMT workflow steps:



GMT Lifetime

All these steps are performed automatically. You can define your own maintenance task as described later in the GMT examples in the following sections. Be aware that your maintenance task should not take too much time because of the maximum runtime defined by the profile parameter `rdisp/max_wp_runtime` (see the following figure).



GMT/Micro Outage: Timing

More Information

Prerequisites for Using the GMT [page 193]

Using the GMT [page 194]

GMT Configuration File sapdb2gmt.conf [page 200]

Example Exit Scripts for the GMT [page 201]

15.1.2.1 Prerequisites for Using the GMT

- See SAP Note 1530812. This SAP Note contains the latest version of the GMT and predefined exit scripts for different maintenance tasks. Use at least GMT script version 6.29.
- Implement SAP Note 1907533 in your SAP system. Attached to this SAP Note is a correction instruction with several functions and reports. These routines are required for the communication of the GMT script with the SAP system using SAP batch events. You have to register the SAP events by executing the report `RSDBA_GMT_REGISTER_EVENTS` as described in the SAP Note.

15.1.2.2 Using the GMT

You use the GMT by performing the following steps:

1. Start the GMT.
2. Configure the GMT.
3. Validate the SAP system before using the micro-outage feature.
4. Initialize the graceful maintenance task.

Starting the GMT

You can start the GMT in interactive or silent mode:

In interactive mode, the following GMT call generates a default log file (`sapdb2gmt.log`) and a default configuration file (`sapdb2gmt.conf`) in the local directory:

```
root# ./sapdb2gmt.sh
```

You can also specify your own log file and configuration file and store them in different locations. The last argument is for improving the trace level to the highest level:

```
root# ./ sapdb2gmt.sh –l ./my/GMT.log –f ./my/GMT.conf –tl 1
```

Alternatively, you can use the script in silent mode and choose one of the following options:

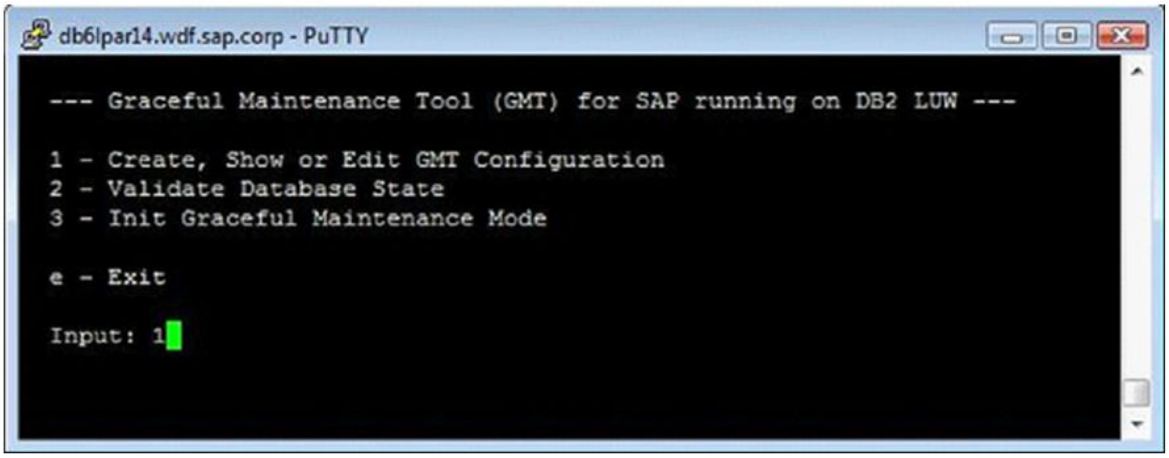
```
root# ./sapdb2gmt.sh -gmt –show -silent [abort|force]
```

```
root# ./sapdb2gmt.sh -gmt –validate -silent [abort|force]
```

```
root# ./sapdb2gmt.sh -gmt –init [–p <password>] -silent [abort|force]
```

Configuring the GMT

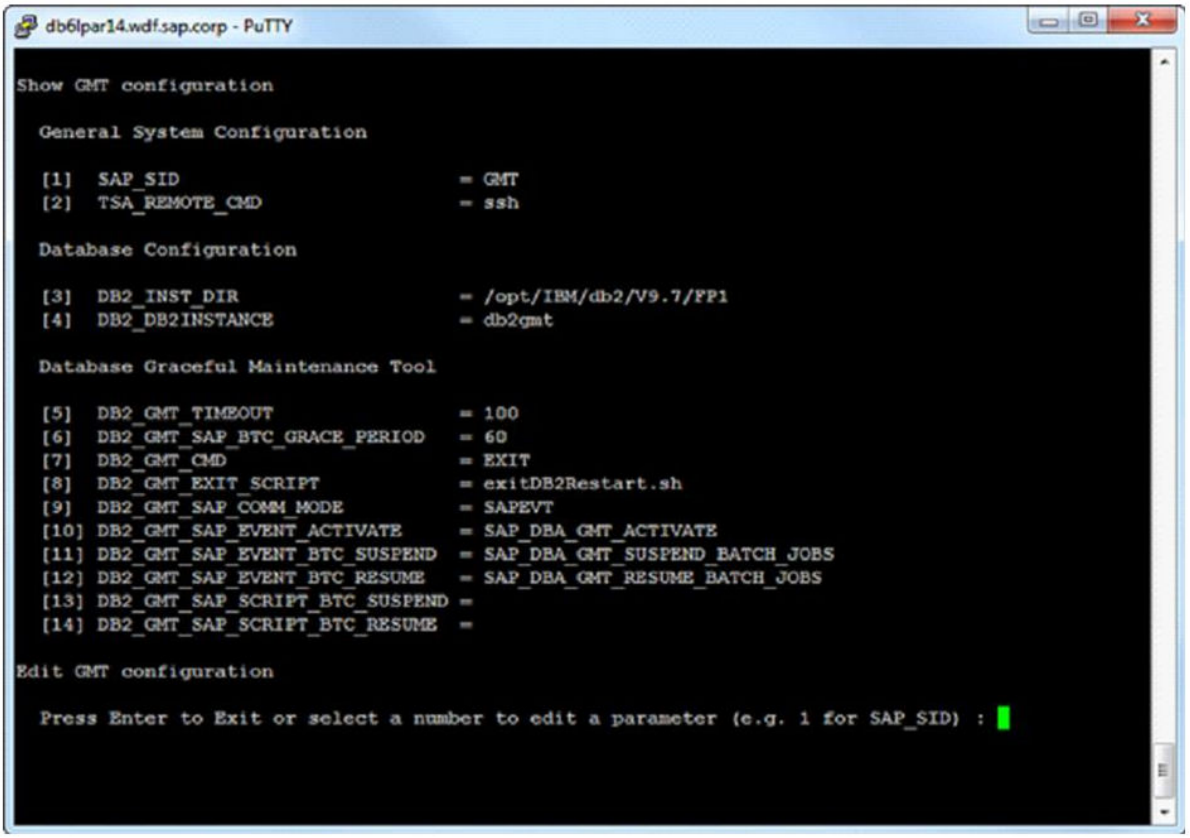
Before you can initiate a graceful maintenance task, the GMT needs some information about your SAP system. To obtain the necessary information, choose option *1 - Create, Show, or Edit GMT Configuration* from the main menu:



GMT Main Menu

Using this option, the script automatically detects most of the required values or uses the default values, for example, a timeout value of 100. You have to call this option at least once to configure the GMT properly.

Additionally, you can use this option to change the values in the GMT configuration file as listed in the following figure:



GMT Configuration

For a list of all parameters that are required for the Graceful Maintenance Tool and that are part of the GMT configuration file, see GMT Configuration File sapdb2gmt.conf [page 200].

In the following, some parameters and their meaning for the GMT script are described in more detail.

GMT Commands

The GMT script can be used with the following commands:

- `DB2_GMT_CMD=EXIT`
  The GMT script automates the DBSL micro-outage usage. The actual maintenance task is performed using an exit script. This is also supported on Db2 HADR primary servers.
- `DB2_GMT_CMD=CLUSTER_FAILOVER`
  The GMT script performs an automated Db2 HADR cluster failover. This maintenance task can be executed from any Db2 cluster server. In this way, the GMT script replaces what was formerly known as “Graceful Cluster Switch”.

Communication Modes Between the GMT and the SAP System

As to communication with the SAP system, you can choose between the following options for the settings of the configuration parameter `DB2_GMT_SAP_COMM_MODE`:

- *SAPEVT* (SAP batch events)
  The GMT script uses the executable `sapevt` to raise SAP batch events in the SAP system. This mode neither requires `startrfc` nor additional user credentials. We recommend that you choose this communication mode.
  You can configure the SAP batch events parameters `DB2_GMT_SAP_EVENT_*` individually. If nothing is customized, the script uses the default names used by the report `RSDBA_GMT_REGISTER_EVENTS`.
- *RFC* (SAP RFC)
  The GMT script uses the deprecated `startrfc` executable. This executable is not delivered anymore in current SAP kernel packages. It has to be deployed manually from older SAP kernel packages and requires user credentials for the RFC call.

Handling of Background Jobs

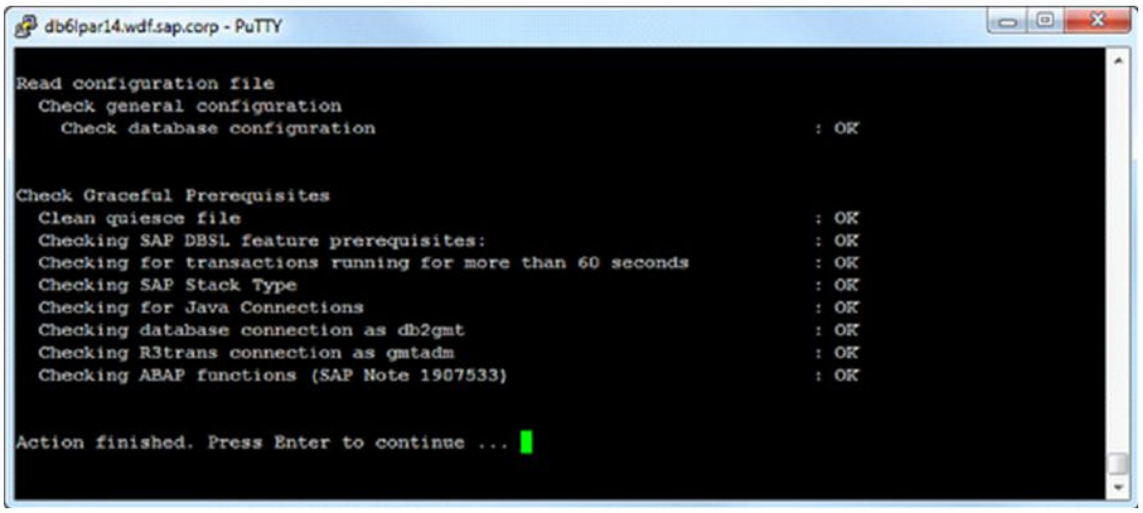
In some SAP applications, a dispatcher background job is executed every minute. This background job schedules other background jobs on demand to process the workload. For maintenance with the GMT, this is an unpredictable workload that can cause long-running transactions. To quiesce those background application patterns, the GMT script suspends the SAP background processing as follows:

1. The script raises the configured SAP batch event `DB2_GMT_SAP_EVENT_BTC_SUSPEND`.
2. The script calls the optional external script configured as `DB2_GMT_SAP_SCRIPT_BTC_SUSPEND` for external schedulers.
3. The script waits for the time set by `DB2_GMT_SAP_BTC_GRACE_PERIOD`. The background jobs have this period of time to finish their activities before the GMT script activates the DBSL micro outage.

After the maintenance was performed or in case of an abortion of the GMT script, the GMT script resumes the SAP background processing by raising the SAP batch event `DB2_GMT_SAP_EVENT_BTC_RESUME`. In addition, the script also calls the optional external script configured as `DB2_GMT_SAP_SCRIPT_BTC_RESUME` for external schedulers.

Validating the System

To easily check if your database administrator and your SID administrator can connect to the database, you can use option *2 - Validate Database State* in the main menu. This option is not mandatory because when you initiate the graceful maintenance task, an automated check runs anyway to identify if your SAP system and your database are in a healthy state and if your GMT configuration is valid.



GMT Validation

Initializing the Graceful Maintenance Task

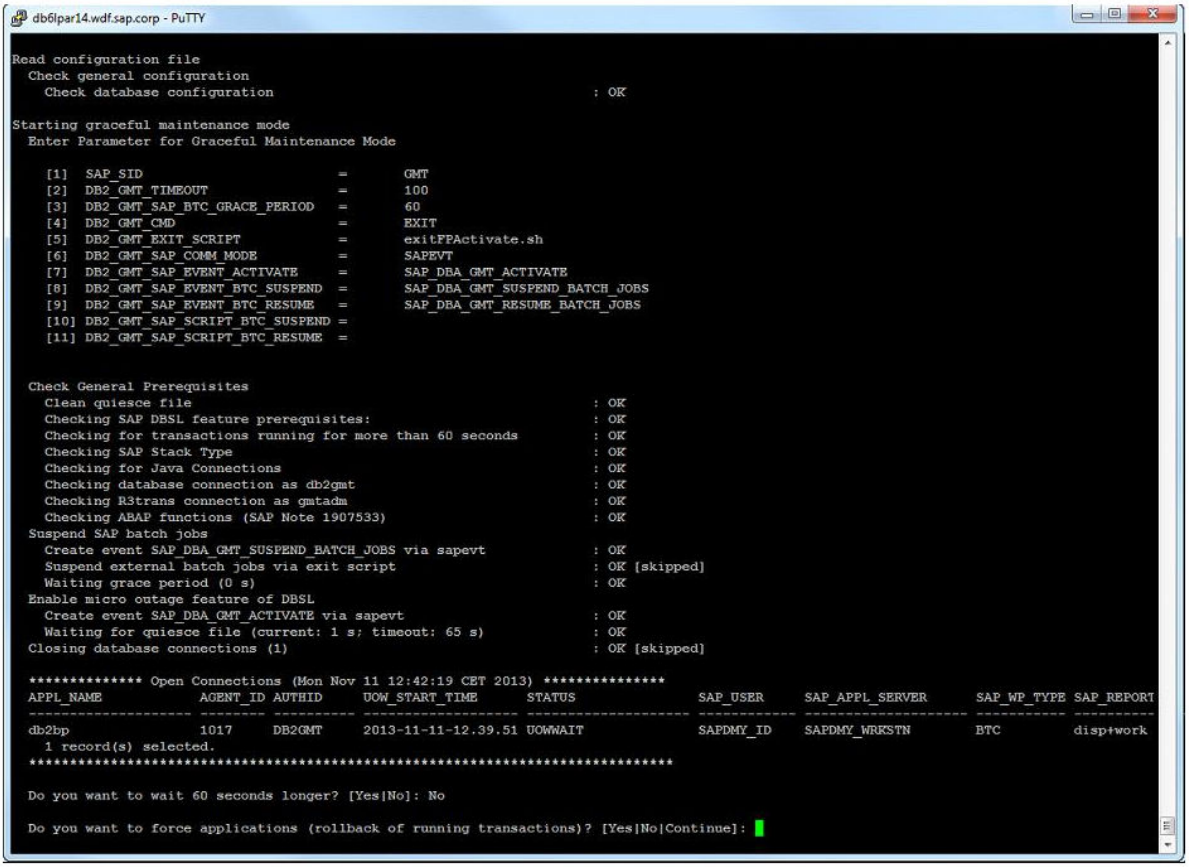
After you have configured the GMT, you can initialize the graceful maintenance task by choosing option *3 - Init Graceful Maintenance Mode* in the main menu. The following configuration file is displayed:

GMT Initialization

The maintenance task can be user-defined, or you can use one of the examples as described in Example Exit Scripts for the GMT [page 201]. To activate, for example, a new Fix Pack level, you can use the predefined exit script `exitFPactivate.sh`. This means that if your database is free of connections, the GMT executes the exit script and waits for the script to finish the maintenance task and to return (with return code *0*).

After the exit script is finished, the GMT disables the graceful maintenance mode, and SAP users can continue their work. It is important that the exit script is well programmed and does not take too much time. The wrapper, that is, the GMT, only warns you if you receive a return code other than zero, but it is your job to handle this exception. No matter what the exit script is doing, if it returns, the GMT disables the micro-outage feature so that the SAP work processes will try to connect to the database within the next 60 seconds.

If there are open connections because not all SAP work processes have disconnected before the GMT timeout, the GMT configuration file looks as follows:



Open Connections

The script provides basic information about these open transactions to help you decide on how to proceed. You can either wait another 60 seconds for the transactions to end or you can proceed with the GMT script. If you proceed, you have the following options:

- *Force Applications [Yes]*
  The GMT script uses `DB2 QUIESCE/UNQUIESCE` to force the open transactions.
- *Abort GMT process [No]*
  The GMT script aborts the processing. As a result, the script only executes all cleanup commands to properly revert the DBSL micro outage.

- *Continue with maintenance [Continue]*
  The GMT script continues to execute the maintenance task. However, you must be sure that the maintenance task can handle such a situation properly. If in doubt, do not use this option.

15.1.2.3 GMT Configuration File sapdb2gmt.conf

The `sapdb2gmt.conf` policy file contains all the parameters that are required for the Graceful Maintenance Tool. The Graceful Maintenance Tool creates the file and sets the parameter values. Usually, you do not have to edit the configuration file manually.

The following list contains all parameters in the GMT configuration file:

General GMT Configuration Parameters

Parameter	Description	Value
`SAP_SID`	SAP system ID	<SAPSID> For example: *GMT*
`TSA_REMOTE_CMD`	Remote shell for Db2 clusters	For example: *ssh*
`DB2_INST_DIR`	Db2 instance directory	*/db2/db2<dbsid>/db2_software*
`DB2_DB2INSTANCE`	Db2 instance name	*db2<dbsid>*
`DB2_GMT_TIMEOUT`	Overall GMT timeout in seconds	*100*
`DB2_GMT_CMD`	GMT command	*EXIT* or *CLUSTER_FAILOVER*
`DB2_GMT_SAP_COMM_MODE`	SAP communication mode	*SAPEVT* or *RFC*

SAP Batch Event Configuration Parameters

Parameter	Description	Value
`DB2_GMT_SAP_EVENT_ACTIVATE`	SAP batch event ID to activate DBSL micro outage	By default: *SAP_DBA_GMT_ACTIVATE*
`DB2_GMT_SAP_EVENT_BTC_SUSPEND`	SAP batch event ID to suspend SAP background processing	By default: *SAP_DBA_GMT_SUSPEND_BATCH_JOBS*
`DB2_GMT_SAP_EVENT_BTC_RESUME`	SAP batch event ID to resume SAP background processing	By default: *SAP_DBA_GMT_RESUME_BATCH_JOBS*

SAP RFC Configuration Parameters

Parameter	Description	Value
`DB2_GMT_AS_HOST`	Host name of the SAP primary application server	For example: *db6lpar14*
`DB2_GMT_AS_NR`	Instance number of the SAP primary application server	For example: *ssh*
`DB2_GMT_USER`	User for RFC calls	*DDIC*
`DB2_GMT_CLIENT`	Client to use for RFC calls	*001*

Configuration Parameters for Background Job Scheduling

Parameter	Description	Value
`DB2_GMT_SAP_BTC_GRACE_PERIOD`	Grace period (in seconds) for background jobs until the DBSL micro outage is activated. If set to *0*, the batch suspend and resume calls are skipped.	By default: *60*
`DB2_GMT_SAP_SCRIPT_BTC_SUSPEND`	Name of the exit script to suspend jobs in an external scheduler	For example: *exitSuspendBtcExternal.sh* (A default value does not exist.)
`DB2_GMT_SAP_SCRIPT_BTC_RESUME`	Name of the exit script to resume jobs in an external scheduler	For example: *exitResumeBtcExternal.sh* (A default value does not exist.)

15.1.2.4 Example Exit Scripts for the GMT

> **i Note**
>
> The exit scripts described here are only examples and not part of the GMT. You have to modify and adapt them according to your own usage scenarios. For the most recent version of the exit scripts, see SAP Note 1530812.

exitDB2Restart.sh

You can use the `exitDB2Restart.sh` script to restart the database and to activate a delayed Db2 configuration value.

The exit script first stops the database instance so that you can execute your offline maintenance task on the database. Then the exit script restarts the database instance and activates the database again.

exitFPActivate.sh

You can use the `exitFPActivate.sh` exit script to activate a new Fix Pack level. To do so, you can choose between the following options:

- `installFixpack`
  You can use `installFixpack` to update your current database software level, which has to be done offline. If you choose this option, you have to add the `installFixpack` call in the script after `db2stop` and before `db2iupdt`. However, keep in mind that this will extend the time of your outage, which you should keep as small as possible. Therefore, we highly recommend that you install the database software in a new directory as described in the following alternative `db2setup` option.
- `db2setup`
  You can use `db2setup` to install the new database software in a new directory while your database is running and just call `db2iupdt` in the outage window. For the `db2iupdt` call, you have to enter the SAP system ID, a log file, and the path of the new software directory. This example script is self-explanatory.

exitSuspendBtcExternal.sh

If a batch grace period is defined, the background jobs are suspended in the SAP system before the micro-outage phase begins. Optionally, you can use the `exitSuspendBtcExternal.sh` script to suspend all jobs from external schedulers. Besides this primary use case, you can also use it for other pre-processing tasks that you want to perform before the GMT starts.

exitResumeBtcExternal.sh

If a batch grace period is defined, the background jobs are resumed in the SAP system after the micro-outage phase ends. Optionally, you can use the `exitResumeBtcExternal.sh` script to resume all jobs from external schedulers. Besides this primary use case, you can also use it for other post-processing tasks that you want to perform after the GMT ends.

exitNoOp.sh

This script does not perform any operations. It is mainly used for GMT test scenarios without database maintenance tasks.

15.2 brdb6brt – Redirected Restore Tool

15.2.1 Using the brdb6brt Tool

You can use the `brdb6brt` redirected restore tool to:

- Perform a simple backup
- Retrieve an overview of the container layout
- Perform a redirected restore
- Change the container layout
- Change the storage path
- Perform a homogeneous system copy
- Create a script for restoring certain tablespaces only
- Check the restore script
- Move existing containers to other directories for DMS tablespaces
- Change text during script generation if required

The following sections provide example commands. For more information about the syntax of `brdb6brt`, see brdb6brt – Command Line Parameters [page 208].

Performing a Simple Backup

You want to make a backup of the entire database to TSM with two sessions. The source database is called `PRD` and the backup is made online.

Enter the following command:

```
brdb6brt -s PRD -bm BACKUP -bpt TSM 2 -ol
```

Retrieving an Overview of the Container Layout

If you want to have an overview of the layout of the database containers, you can create a restore script only.

To do so for database `PRD` as instance owner `db2prd` with password `PASS123`, enter the following command:

```
brdb6brt -s PRD -bm RETRIEVE -user db2prd -using PASS123
```

The script contains the timestamp of the last successful backup from the history file. You can also use this script for a restore operation if you earlier have taken a backup of the database. You have to change the script to the timestamp of the backup image that you want to use.

The created script is based on the current layout of the database. The current layout, however, might deviate from one that existed at the time a backup was taken. Therefore, when you use the created script for a redirected restore you have to make sure that the script is suitable for restoring the backup. As an alternative, you can use the Db2 restore command with the `GENERATE SCRIPT` option to generate a suitable redirected restore script from a backup.

Performing a Redirected Restore

Use the Db2 Command Line Processor (`DB2 CLP`) to perform a redireced restore based on a redirected restore script. Enter the following command:

```
db2 -tvf <script file>
```

The parameters have the following meaning:

Parameter	Meaning
`-t`	Forces the CLP to use a semicolon (;) as terminating character for an SQL statement. The use of this option is mandatory for the execution of the script.
`-v`	Forces the CLP to print each statement on the screen.
`-f <file>`	Forces the CLP to read the statements from the specified script file.

If a backup and restore script of the database `PRD` was created, you should now execute the script by entering the following command:

```
db2 -tvf PRD NODE0000.scr
```

Changing the Container Layout of DMS Tablespaces

During a redirected restore, you can change the container layout of DMS tablespaces of your database. You can change, for example, the number of containers of a tablespace, their sizes, or their location in the file system.

The following procedure is an example of how you can change the container layout and store the backup in three separate directories:

1. To create the backup and the restore script, enter the following command:
   ```
   brdb6brt -s <DBSID> -bm BOTH -bpt Y:\BACKUPS1 Y:\BACKUPS2 Y:\BACKUPS3
   ```
   Since the database is rather large, the backup is split and stored in three separate directories.
2. Edit the script `<DBSID>_NODExxxx.scr` and change the location, size, and number of the container.
3. To change the container layout, restore the database by entering the following command:
   ```
   db2 -tvf <DBSID>_NODExxxx.scr
   ```

Changing Storage Paths for Automatic Storage Tablespaces

> ⚠ Caution
>
> The following procedure **only** applies if you are using **automatic storage tablespaces** and a database for which automatic storage is enabled.

If automatic storage is enabled for your database, you can change the number and locations of the storage paths.

The following procedure is an example of changing the storage paths for the automatic storage tablespaces and storing the backup into three separate directories:

1. To create the backup and the restore script, enter the following command:
   `brdb6brt -s <DBSID> -bm BOTH -bpt Y:\BACKUPS1 Y:\BACKUPS2 Y:\BACKUPS3`
   Since in this example the database is rather large, the backup is split and stored in three separate directories.
2. Edit the `<DBSID>_NODExxxx.scr` script and change the automatic storage paths for the automatic storage tablespaces (*ON* clause).
3. To change the container layout, restore the database by entering the following command:
   `db2 -tvf <DBSID>_NODExxxx.scr`

Automatic storage enabled databases can also have DMS tablespaces without automatic storage in addition. The layout of such DMS tablespaces can be changed as described above in section *Changing the Container Layout for DMS Tablespaces*.

Performing a Homogeneous System Copy

You want to copy your database to another machine. For this purpose, you have to adapt the container locations. To do so for the database `<DBSID>`, proceed as follows:

1. To create the backup and the restore script, enter the following command:
   `brdb6brt -s <DBSID> -bm BOTH -bpt Y:\BACKUPS1 Y:\BACKUPS2 Y:\BACKUPS3`
   Since in this example the database is rather large, the backup is split and stored in three separate directories.
2. Make the backup images and the script available on the target machine by copying them to the machine using `ftp`.
3. Log on to the target machine as user `db2<dbsid>` and edit the script `SDB.scr`.
4. Change the container locations. In addition, you also need to adapt the location of the backup image to the directory or device where it is available on the target machine.
5. Restore the database by entering the following command:
   `db2 -tvf <DBSID>_NODExxxx.scr`

Creating a Script for Restoring Certain Tablespaces Only

You want to back up one or more tablespaces rather than the entire database. The tablespaces for backup are called `USERSPACE1`, `TBSPACE` and `TESTSP2`. The backup is done to TSM (three sessions). In this example, the database name is `PRD`. The restore script is created to restore only the specified tablespaces using the following command:

`brdb6brt -s PRD-bm BOTH -bpt TSM 3 -tbs USERSPACE1 TBSPACE TESTSP2`

Checking the Restore Script

After you edited the script, you can check whether the script would succeed on that machine. The check allows scripts that perform full restores on database or tablespace levels. In addition, restoring to a new or existing database is also considered as an option. The user who runs the check should be the same user who will later run the script using Db2 CLP; in most cases this is the Db2 instance owner (`db2<dbsid>`).

To run a check on the `<DBSID>_NODExxxx.scr` script, enter the following command:

```
brdb6brt -bm CHECK -ip <DBSID>_NODExxxx.scr
```

The output shows possible errors, warnings, and information about the redirected restore operation that the script is going to perform. Possible errors start with `[E]`, warnings with `[W]` and information messages with `[I]`. The output is also saved to a file in the current directory. The file name of the output is the same as the script name, but has the file extension `.chk`.

With the content of the check output, which is mainly error and warning messages, you should be able to find errors in the script or in the database server system, for example, duplicated file names, missing write authorizations, or out-of-space situations. The output file also provides additional information about the used file systems, for example:

- A list of tablespace containers
- Information about used space
- Information about free space
- Information about missing space
- Information about required space in the Db2 log directory

> **i** Note
>
> So before running `brdb6brt`, implement the tablespace container layout (that is, distribute tablespace containers in the file systems) by creating file systems, directories, and links to receive flawless check output.

In a multi-partition database environment, you need to run `brdb6brt` for all partitions of your database. To do so, use the `-nn <node number>` parameter. The scripts created include the partition number, which prevents existing scripts from other database partitions from being overwritten.

> **i** Note
>
> If you use the `-nn all` option, scripts are automatically created for all database partitions.

Moving Existing Containers to Other Directories for DMS Tablespaces

> **⚠** Caution
>
> If you are using automatic storage tablespaces and a database for which automatic storage is also enabled, you **must not** use the following procedure.

With `brdb6brt` patch 5 or higher, you can create relocate scripts to move existing containers to other directories using the `db2relocatedb` tool. Furthermore, you can use these scripts to initialize mirrored databases with a modified container layout using the `db2inidb` tool and the `RELOCATE USING` parameter.

1. To create the relocate script, enter the following command:
   `brdb6brt -s PRD -bm RETRIEVE_RELOCATE`
   Script `PRD_NODE0000.scr` is generated.
2. Modify the script `PRD_NODE0000.scr` according to your requirements.
3. To update the internal container path of the database using the `db2relocatedb` tool, enter the following command:
   `db2relocatedb -f PRD_NODE0000.scr`
4. To initialize the mirrored database, for example, to create a database clone using the `db2inidb` tool, enter the following command:
   `db2inidb <NEW_DBSID> as snapshot relocate using <OLD_DBSID>_NODExxxx.scr`

Changing the Storage Path

> ⚠ Caution
>
> The following procedure **only** applies if you are using automatic storage tablespaces and a database for which automatic storage is also enabled.

If the database is enabled for automatic storage management, it can have automatic storage tablespaces as well as regular DMS tablespaces. The database has one or more storage paths (which are database parameters) and automatically handles the space allocation for the automatic storage tablespaces. For information about how to handle regular DMS tablespaces, see *Changing the Container Layout* earlier in this section.

To change the storage paths for the automatic storage tablespaces, proceed as follows:

1. To create the relocate script, enter the following command:
   `brdb6brt -s PRD -bm RETRIEVE_RELOCATE`
2. Edit the `PRD_NODE0000.scr` script and change the automatic storage paths for the automatic storage tablespaces.
3. To update the internal container path information of the database using the db2relocatedb tool, enter the following command:
   `db2relocatedb -f PRD_NODE0000.scr`

Changing Text During Script Generation

`brdb6brt` creates the scripts that are used to perform a redirected restore and to relocate the database (`relocate DB script`). You then have to adapt the script according to your requirements.

With `brdb6brt` patch 5, a new parameter `-replace <ReplaceDefinition>` was introduced. You can use this parameter to adjust the script output during its generation instead of adapting the output manually afterwards.

> **Example**
>
> You can use this parameter, for example, to change the name of the target database from PRD to QAS and the container location from `/db2/PRD` to `/db2/QAS`:
>
> ```
> brdb6brt -s PRD -bm RETRIEVE -replace PRD=QAS,db2prd=db2qas
> ```

15.2.2 brdb6brt – Tool Command Line Parameters

Use

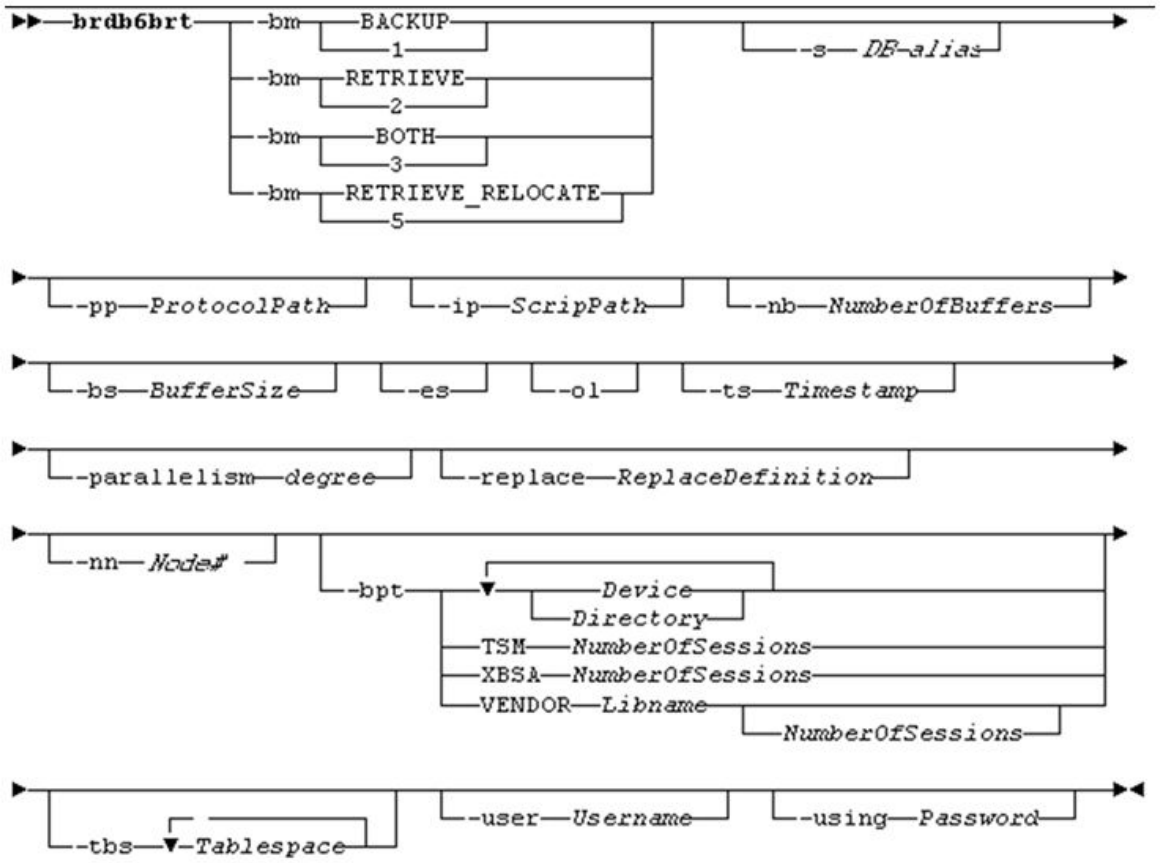
`brdb6brt` runs in the following modes:

- Backup or retrieve
- Check
- Tool information

The following sections provide syntax examples of each mode.

Backup or Retrieve Mode

To create a backup or a restore script, use the following syntax:



brdb6brt – Backup or Retrieve Mode

Parameter	Description
`-V`	Displays the version information (patch level) of `brdb6brt`
`-h`	Displays an overview of the command line options of `brdb6brt`
`-bm BACKUP`	Creates a backup of the specified database only
`-bm RETRIEVE`	Creates the restore script for the specified database only
`-bm BOTH`	Creates a backup of and the restore script for the specified database
`-bm RETRIEVE RELOCATE`	Creates the relocate script for the specified database
`-s <DB-alias>`	Alias of the database for which the backup or restore script should be created

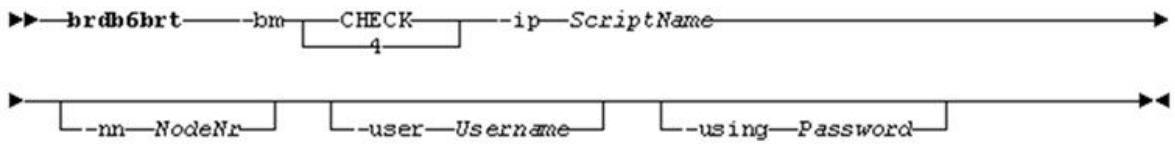
Parameter	Description
`-pp <ProtocolPath>`	Directory where the protocol file for the `brdb6brt` run is written to The default directory is the working directory. The protocol file is named `<SourceDB>.brp` or `<SourceDB>_NODE<NodeNumber>.brp` in a multi-partition database environment.
`-i <ScriptPath>`	Directory where the restore script is written to The default directory is the working directory. The restore script is named `<SourceDB>.scr` or `<SourceDB>_NODE<NodeNumber>.scr` in a multi-partition database environment.
`-nb <NumberOfBuffers>`	Number of buffers that are reserved for the execution of the backup The default value is `2`.
`-bs <BufferSize>`	Size of the buffer for the backup operation The size is measured in allocation units of 4 KB. The default value is `1024`.
`-es`	The restore script is created for experts, that is, only comments that are really needed are included.
`-ol`	Backup operation is done online.
`-ts <Timestamp>`	Only used in retrieve mode If this parameter is specified, the timestamp in the restore script is set to this value, which must have format `YYYYMMDDhhmmss`. The default value is the current date and time or the timestamp of the latest available backup.
`-replace <ReplaceDefinition>`	With this option you replace strings in the generated scripts for redirected restore and relocate. Parameter `<ReplaceDefinition>` must have the format `<orig. string 1>=<repl. string 1>,<orig. string 1>=<repl. string 2>,…`. This option only makes sense if you also specified the following `–bm` options: • **`–bm RETRIEVE`** • **`-bm BOTH`** • **`–bm RETRIEVE RELOCATE`**
`-parallelism <degree>`	Parallelism degree for backup and redirected restore operations
`-nn <NodeNr>`	In a multi-partition database environment, the backup is performed on this node. The restore script is specific to this node and is named `<SourceDB>_NODE<NodeNumber>.scr`.

Parameter	Description
`-nn ALL`	In a multi-partition database environment, it addresses all nodes for the specific operation. If you perform a backup using `-nn ALL`, `brdb6brt` creates a single system view backup.
`-bpt <Device>`	To back up the database to tape, specify a valid tape device. **i Note** You can split the backup into multiple pieces by specifying multiple devices separated by blanks.
`-bpt <Directory>`	To back up the database to a directory, specify a valid directory. **i Note** Make sure that sufficient space is available in the directory for the backup. You can split the backup into multiple pieces by specifying multiple directories separated by blanks.
`-bpt TSM [<NumberOfSessions>]`	To back up the database to TSM, specify the number of sessions (`<NumberOfSessions>`) required for the TSM connection.
`-bpt XBSA [<NumberOfSessions>]`	To back up the database to a XBSA-compliant storage management, specify the number of sessions (`<NumberOfSessions>`) required for the XBSA connection.
`-bpt VENDOR <LibName> [<NumberOfSessions>]`	To back up the database to a vendor product, specify the shared library (required for the backup operation) and – optionally – the number of sessions (`<NumberOfSessions>`) required for the connection to the vendor product.
`-tbs <Tablespace>`	If this option is not specified, a full database backup is performed. However, you can decide to back up only one or more tablespaces of the database by specifying the tablespaces separated by blanks. The restore script is then only created for the specified tablespaces.
`-user <Username>`	Specifies another user with which you can run `brdb6brt`
`-using <Password>`	Password for the specified user

Check Mode

To check whether a given restore script would succeed on the machine where you want to use the restore script, use this syntax.

> **Note**
>
> The user who performs the check mode should be the database instance owner. The terminal output of the check run is written to a protocol file in the current working directory. The name of the protocol file is `<SourceDB>.chk` or `<SourceDB>_NODE<NodeNumber>.chk` depending on the specified script name.



brdb6brt – Check Mode

Parameter	Description
`-bm CHECK`	Checks if a given restore script would succeed on the machine where you run `brdb6brt`
`-ip <ScriptName>`	Name of the restore script to be checked By default, the restore script is named `<SourceDB>.scr` or `<SourceDB>_NODE<NodeNumber>.scr` in a multi-partition environment.
`-nn <NodeNr>`	In a multi-partition environment, the specified node is checked.
`-user <Username>`	Specifies another user with which you can perform this operation
`-using <Password>`	Password for the specified user

More Information

For more information about the latest `brdb6brt` patches, see SAP Note 867914.

15.3 db6util - Tool to Assist Database Administration

15.3.1 db6util Tool - Command Line Parameters

This section provides information about the syntax of `db6util`.

> **Code Syntax**

```
db6util [ -h                                                    ]
        [ -V                                                    ]
        [ -n        <dbname>                                    ]
```

```
                    [ -o        <logfile>                                  ]
                    [ -w        <resultfile>                               ]
                    [ -auth     <username>   [password]   [schema]         ]
                    [ -remote   <dbhost>     <svcename>                    ]
                    [ -r        <tabname>                                  ]
                    [ -rf       <filename>                                 ]
                    [ -rv                                                  ]
                    [ -f                                                   ]
                    [ -dg       <db  parameter>                            ]
                    [ -mg       <dbm parameter>                            ]
                    [ -ping     [sleep time] [number of ping tests]        ]
                    [ -sd       [sleep time] [number of snapshots] [cmd]   ]
                    [ -sl       [sleep time] [number of snapshots] [cmd]   ]
                    [ -lock2key [row lock name]                            ]
                    [ -mvt      <tabname>                                  ]
                    [ -rtvt     <old tablespace> <new tablespace>          ]
```

Parameter	Description
`-h`	Prints help text
`-V`	Prints version information
`-n <dbsid>`	Specifies the database name By default, the value of the environment variable `DB2DBDFT` is used.
`-o`	Specifies the log file The default value is standard output (`stdout`).
`-w`	Specifies the result file The default value is standard output (`stdout`).
`-auth`	Specifies the user authentication If this option is not specified, `db6util` tries to retrieve the `<sapsid>adm` password from the Db2 password service.
`-remote`	Retrieves information about the remote connection If this option is not specified, `db6util` assumes that the database is cataloged on the database client.
`-r`	Performs `RUNSTATS` on a single table and all its indexes
`-rf`	Performs `RUNSTATS` on tables that were provided in a file list
`-rv`	Performs `RUNSTATS` on tables with the `VOLATILE` attribute Tables that are flagged with *ACTIVE = N* in table `DBSTATC` are not affected. The `VOLATILE` attribute is removed after the `RUNSTATS`.
`-f`	Retrieves information about free space in a tablespace
`-dg`	Retrieves database parameters

Parameter	Description
`-mg`	Retrieves database manager parameters
-ping	Pings the database server using CLI and determines the average ping time
`-sd`	Displays an overview of deadlock processes in the application snapshot
`-sl`	Displays an overview of deadlock processes and processes in lock-wait status in the application snapshot
`-lock2key`	Attempts to resolve row locks to key values on tables with unique constraints. You can use this option in addition to the `-sl` and `-sd` options, or you can explicitly supply a row lock name.
`-mvt`	Replaces a virtual table with its corresponding empty table
`-rtvt`	Renames a tablespace in all virtual tables
`-prof`	Inserts the optimization profile from an xml file into table `SYSTOOLS.OPT_PROFILE`

Related Information

Using the db6util Tool [page 214]

15.3.2 Using the db6util Tool

Use

The `db6util` tool contains a collection of utility routines that are mainly used during the SAP system upgrade.

The following `db6util` options are also useful for database administration and troubleshooting and can be entered using the command line. To generate a complete list of all `db6util` options, you can call `db6util –h` from the command line.

The results or messages generated by all `db6util` commands can be redirected by the command options `[ -o <log file> ]` or `[ -w <resultfile> ]`.

Tablespace Free Space

To generate a free space list for all tablespaces, enter the following command:

```
db6util –f
```

Db2 RUNSTATS Options

- To perform `RUNSTATS` on a single table, enter the following command:
  `db6util -r <tabname>`
- To perform `RUNSTATS` on all tables specified in a file, create a file containing a list of tables and enter the following command:
  `db6util -rf <filename>`
- To perform `RUNSTATS` on all tables that were temporarily marked as `VOLATILE` in the database and to remove the `VOLATILE` attribute from the tables after `RUNSTATS` has run, enter the following command:
  `db6util -rv`

> ⚠ Caution
>
> Do not use the **`-rv`** option for systems that are enabled for the Db2 automatic `RUNSTATS` feature. Only use **`db6util -rv`** if requested by the support team.

> i Note
>
> Tables that are marked with an `N` in the `ACTIVE` column in table `DBSTATC` are not affected by this option.

Database Lock Overview

`db6util` helps to analyze database lock-wait situations by extracting all involved processes from an application snapshot and displaying their dependencies in the form of a syntax diagram. Detailed information about those processes, such as the last SQL statements that were executed or lock types, is displayed.

- To display processes that are only involved in a deadlock situation, enter the following command:
  `db6util -sd`
- To display all processes that are involved in a lock-wait situation, enter the following command:
  `db6util -sl`

To take snapshots periodically, you can execute both commands with additional parameters, for example, as follows:

```
db6util -sd [sleep time] [number of snapshots]
```

```
db6util -sl [sleep time] [number of snapshots]
```

More Information

For more information about further use and syntax of the `db6util` tool, see the following sections:

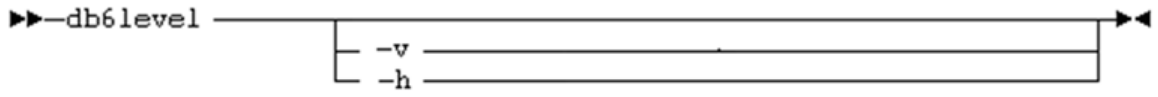
db6util Tool - Command Line Parameters [page 212]

Monitoring Lock Activity and Deadlocks on the Db2 Command Line [page 132]

Monitoring Network Time [page 144]

15.4 db6level - Tool to Check Db2 Client Libraries

The `db6level` tool loads the Db2 client libraries and displays their version. In its output, `db6level` also shows the version of the Db2 software. The syntax of the `db6level` tool looks as follows:



Syntax – db6level

Parameter	Description
`-v`	Prints detailed information about the loaded Db2 libraries ('verbose' mode)
`-h`	Displays an overview of the command line options of `db6level`

You can compare the client version number displayed by `db6level` with the Db2 server version number that is displayed by `db2level`. On an SAP system, the database client must have the same main version as the database server, but the client may have a lower Fix Pack level.

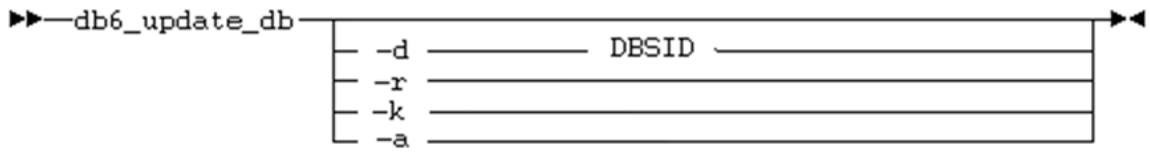
15.5 db6_update_db – Tool to Enable New Features After a Database Upgrade or Fix Pack Installation

Use

After a database upgrade or a Fix Pack installation, you need to use the scripts `db6_update_db.sh` (UNIX) and `db6_update_db.bat` (Windows) to ensure that important SAP-specific updates are applied to your database.

Make sure you read SAP Note 1365982 and always use the latest version of the script that is attached to this SAP Note.

The following section provides information about the syntax of `db6_update_db`.



Syntax of db6_update_db

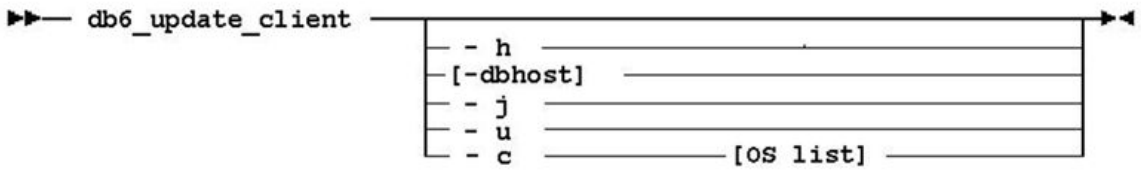
Parameter	Description
`-d <DBSID>`	Database ID `<DBSID>` If this parameter is not set, the value of environment variable `DB2DBDFT` is used.
`-r`	Activates Db2's automatic `RUNSTATS`
`-k`	Specifies that `DB2_WORKLOAD=SAP` is not set
`-a`	Specifies that DMS tablespaces are not enabled for automatic resize
`-m -u`	Specifies that the script is called in the scenario of a **database upgrade** (that is, a change of a major database release) and **not** after a Fix Pack installation.
`-s`	Specifies that the script is called on a HADR standby node
`-enable_roles`	Enables the role-based security concept for SAP systems

More Information

- Role-Based Security Concept for Database Users [page 27]
- For more information about the role-based security concept and separation of duties, see Role-Based Security Concept for Database Users on IBM Db2 for Linux, UNIX, and Windows on SAP Community.
- For more information about the `db6_update_db` script, see SAP Note 1365982.

15.6 db6_update_client - Script to Update the Client Software

You use the scripts `db6_update_client.sh` (UNIX) and `db6_update_client.bat` (Windows) to update or install the Db2 client software. The following section provides information about the syntax of `db6_update_client`.



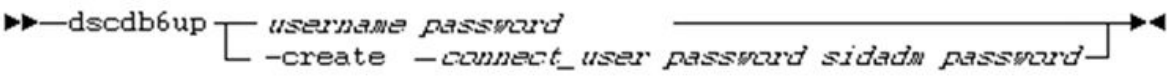
Syntax of db6_update_client

Parameter	Description
`No option`	Description of use is printed. Same as `-h`.
`-h`	Description of use. In addition, prints also allowed operating system parameters.
`-u`	Updates all installed CLI and JDBC drivers in `global/db6` with the versions from the client DVD
`-c <OS_LIST>`	Installs the CLI driver for the provided operating system and installs the JDBC driver
`-dbhost <DBHOSTNAME>`	Database host name This parameter is optional if the database server has several network cards with several database host names connected to them. The provided database host name is used in file `db2cli.ini`. If `<DBHOSTNAME>` is not specified, the value of `uname -n` is used for `db2cli.ini`.
`-j`	**Java systems only:** Only the JDBC driver is copied to the global directory.
`-nodb`	Specify this flag if you call this script on a non-Db2 system and you want to install Db2 CLI/JDBC drivers in the global directory (for example, for remote monitoring). **i** Note You **cannot** use this option in combination with the `-dbhost option`.

15.7 dscdb6up - Tool to Set and Update Passwords

The tool `dscdb6up` sets and updates the password of the SAP system administrator `<sapsid>adm` and the ABAP database connect user `sap<sapsid>` or `sapr3` (for SAP systems originally installed with version 4.6D or lower).

The tool updates the content of the `dscdb6.conf` file. Note that on multi-partition database systems, you have to update the operating system passwords on all database nodes.



Syntax – dscdb6up

Parameter	Description
`<username> <password>`	User name and its new password
`-create` `<connect_user password> <sapsidadm password>`	Overwrites the existing password file, but does not change operating system level passwords

If you do not want to provide the passwords in the OS commands, you can omit them in the parameters and `dscdb6up` will prompt you for the passwords as shown in the following examples:

Example

```
fmhsusell: n71adm 345> dscdb6up sapn71
Enter new password for user "sapn71":
```

Example

```
fmhsusell: n71adm 346> dscdb6up -create
Enter new password for connect_user:
```

15.8 rsecssfx - Tool to Create and Update Secure Storage in the File System

The command-line tool `rsecssfx` creates, updates, and lists entries in the secure storage in the file system.

Prerequisites and Use

For SAP systems running on IBM Db2 for Linux, UNIX, and Windows, secure storage in the file system is used to store the database connect user name and password. The minimum kernel requirement for IBM Db2 for Linux, UNIX, and Windows is 7.49. You use `rsecssfx` to create and update user names and passwords in the secure storage.

Entries in the Secure Storage

There are two database-relevant entries in the secure storage. The name of the connect user is stored in the record key `DB_CONNECT/DEFAULT_DB_USER` and the password of the connect user is stored in the record key `DB_CONNECT/DEFAULT_DB_PASSWORD`.

```
rsecssfx list
|-------------------------------------------------------------------------------
-|
| Record Key                     | Status             | Time Stamp of Last
Update |
|-------------------------------------------------------------------------------
-|
| DB_CONNECT/DEFAULT_DB_PASSWORD | Encrypted          | 2022-09-26  23:54:55
UTC |
|-------------------------------------------------------------------------------
-|
| DB_CONNECT/DEFAULT_DB_USER     | Plaintext          | 2022-09-26  23:54:50
UTC |
|-------------------------------------------------------------------------------
-|
| SYSTEM_PKI/PIN                 | Encrypted          | 2022-09-27  01:21:42
UTC |
|-------------------------------------------------------------------------------
-|
| SYSTEM_PKI/PSE                 | Encrypted (binary) | 2022-09-27  01:21:44
UTC |
```

The entry for the database connect user name can be created or updated using the following command:

```
rsecssfx put DB_CONNECT/DEFAULT_DB_USER sap<sapsid>
```

The entry for the database connect user password can be created or updated using the following command:

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

The name is stored as plain text and the value of the key can be retrieved from the command line:

```
rsecssfx get DB_CONNECT/DEFAULT_DB_USER
Record Key   : DB_CONNECT/DEFAULT_DB_USER
Record Value : sap<sapsid>
Time Stamp   : 2022-09-26  23:54:50  UTC
Host Name    : <hostname>
OS-User      : <sapsid>adm
```

When the key for the password is retrieved, the value isn't shown:

```
rsecssfx get DB_CONNECT/DEFAULT_DB_PASSWORD
Record Key   : DB_CONNECT/DEFAULT_DB_PASSWORD
Record Value : <Encrypted text>
Time Stamp   : 2022-09-26  23:54:55  UTC
Host Name    : <hostname>
OS-User      : <sapsid>adm
```

More Information

For more information about using `rsecssfx` for password management for IBM Db2, see User Authentication Concept for AS ABAP [page 30].

For more information about secure storage and the tool `rsecssfx`, see the SAP NetWeaver security guide, for example, for release 7.5: Administering the Secure Storage in the File System (AS ABAP).

16 SAP-Specific User-Defined Functions and Stored Procedures

User-defined functions (UDFs) and stored procedures are contained in shared libraries. The shared library `db2sap` is part of the Db2 installation. Before you can use the UDFs and stored procedures that are contained in `db2sap`, they must be activated. The UDFs are activated during installation or as part of the `db6_update_db` post-processing.

17 Configuring Tivoli Storage Manager (TSM) / IBM Spectrum Protect

Use

When you perform a database backup or restore, you can specify Tivoli Storage Manager (TSM) to manage the database or tablespace backup. You can also use TSM to manage archived Db2log files. However, before the Db2 database manager `db2<dbsid>` can use TSM, you have to configure TSM as described in the following sections.

The overall procedure has been improved. For example, you do not need to install the TSM BA client anymore.

> → Recommendation
>
> We recommend that you use at least TSM Version 5.4.1.

> i Note
>
> As of Version 7.1.3, IBM Tivoli Storage Manager has been renamed to IBM Spectrum Protect. The names are often used interchangeably, but in this document the product is referred to as *TSM*. The directory name `sqllib/adsm` hasn't changed and still contains the old product name.

Process Flow

1. Install the TSM application programmer interface (API) on the Db2 server.

   > i Note
   >
   > In a multi-partition database environment, you have to install the TSM API on each physical partition host.

2. Configure the TSM files `dsm.opt` (UNIX and Windows) and `dsm.sys` (UNIX only) as described in Configuring TSM Files dsm.opt and dsm.sys [page 224].
3. Configure the TSM environment variables `DSMI_DIR, DSMI_CONFIG`, and `DSMI_LOG` as described in Configuring TSM Environment Variables [page 225].
4. To generate the password file, which is required for the connection between the TSM client and the TSM server, enter the command **`dsmapipw`** as user `root`.
5. To make the environment variables available to Db2, restart Db2.

More Information

For more information about the configuration of the TSM files `dsm.opt` and `dsm.sys` as well as about the configuration of the `DSMI` environment variables, see SAP Note 82029.

17.1 Configuring TSM Files dsm.opt and dsm.sys

On UNIX systems, you must configure the TSM files `dsm.opt` and `dsm.sys`. On Windows systems, you have to configure the TSM file `dsm.opt`.

The files must contain the following parameters:

UNIX

File dsm.opt

Parameter	Value
`SERVERNAME`	`<TSM server name>`

File dsm.sys

Parameter	Value
`SERVERNAME`	`<TSM server name>`
`COMMETHOD`	`TCP/IP`
`TCPPORT`	`1500` (example of TSM port number)
`TCPSERVERADDRESS`	`<TSM server IP address or host name and domain name>`
`PASSWORDACCESS`	`Generate`
`PASSWORDDIR`	`/etc/tsm` (example)

> ⚠ Caution
>
> The entry of `SERVERNAME` in the `dsm.opt` file references the corresponding section in the `dsm.sys` file.

> ⚠ Caution
>
> For parameter `PASSWORDDIR`:
>
> The users `root` and `db2<dbsid>` must be able to access the password directory (for example, `/etc/tsm`). Make sure that this directory has the appropriate authorizations.

Windows

File dsm.opt

Parameter	Value
COMMMETHOD	TCP/IP
TCPPORT	1500
NODENAME	<TSM client>
TCPSERVERADDRESS	<TSM server IP address or host name and domain name>
PASSWORDACCESS	Generate

17.2 Configuring TSM Environment Variables

You require the TSM environment variables `DSMI_DIR`, `DSMI_CONFIG`, and `DSMI_LOG` to specify the location of the TSM API client and its configuration files.

Procedure Under UNIX

You update the files `/db2/db2<dbsid>/sqllib/usercshrc` and `/db2/db2<dbsid>/sqllib/userprofile` by inserting the values of the environment variables as follows:

Environment Variable	Value
DSMI_DIR	/usr/tivoli/tsm/client/api/bin64
DSMI_CONFIG	/db2/db2<dbsid>/sqllib/adsm/dsm.opt
DSMI_LOG	/db2/db2<dbsid>/sqllib

By storing the TSM-related environment variables in these files, you ensure that the environment settings are also available in a multi-partition database environment.

> ⚠ Caution
>
> Make sure that the files `usercshrc` and `userprofile` are sourced when you log on as user `db2<dbsid>` or `<sapsid>adm`.

At the end of the files `.login` and `.profile` in the home directories of `db2<dbsid>`, check for the following entries:

- File `.login`:
  `source /db2/db2<dbsid>/sqllib/db2cshrc`
- File `.profile`:
  `. /db2/db2<dbsid>/sqllib/db2profile`

If they do not yet exist, copy the entries to the `.login` and `.profile` files of user `<sapsid>adm`.

Procedure Under Windows

You set the environment variables in either the global environment or user environment of user `db2<dbsid>` as follows:

Environment Variable	Value
`DSMI_DIR`	`<drive>:\progra~1\tivoli\tsm\api`
`DSMI_CONFIG`	`<drive>:\progra~1\tivoli\tsm\baclient\dsm.opt`
`DSMI_LOG`	`<drive>:\ progra~1\tivoli\tsm\api`

17.3 Configuration Considerations

With TSM, you can use the same or different management classes for archiving backups and log files. The preferable solution, however, is that you use a management class for backups **different** from the one you use for log files. If you use the same management class, backups and log files might be archived to the same tape. In this case, the following disadvantages have to be taken into consideration:

- If you lose a tape, you not only lose a backup but also the log files. If you only lose a backup, you can use an older backup and still perform a rollforward recovery. With log files missing, this is not possible.
- The performance during rollforward recovery is limited because log files are widely spread on the tape if stored on the same tape as backups.
- Sometimes you might be forced to keep offline backups longer than log files. If log files are deleted from tape by TSM, you **cannot** reuse the whole tape because backups still reside on that tape and space reclamation drastically affects the system performance.

→ Recommendation

We recommend that for archiving log files you use a **disk storage pool**. With this pool, you can achieve a better system and rollforward performance if caching on the disk storage pool is switched on.

For backups, we recommend that you **do not** use such a disk storage pool. If the Db2 backup image does not fit into this disk storage pool, TSM will fail. An extremely large disk storage pool would be necessary to avoid this problem.

A Appendix

A.1 References

Are you looking for more documentation? Here's an overview of information sources that are available for SAP systems on IBM Db2 for Linux, UNIX, and Windows.

Documentation by SAP

The following documentation is available on SAP Help Portal and SAP Support Portal:

> → Recommendation
>
> For central access to **all** our documentation, use our SAP on IBM Db2 overview page on SAP Help Portal.

Description	Links
Database Administration Guide for SAP on IBM Db2 for Linux, UNIX, and Windows	https://help.sap.com/viewer/db6_admin
Implementation documentation such as guides for installation, system copy, and SAP system upgrades	Installation guides for SAP Application Server systems based on SAP NetWeaver Software Update Manager (SUM) guides System copy guides
DBA Cockpit documentation	Database Administration Using the DBA Cockpit: IBM Db2 for Linux, UNIX, and Windows
SAP Business Warehouse on IBM Db2 for Linux, UNIX, and Windows: Administration Tasks	Db2 10.5 and higher Db2 10.1 and lower (out of mainstream maintenance)
Database upgrade guides	Db2 11.5 Db2 11.1 DB2 10.5 DB2 10.1 (out of mainstream maintenance) DB2 V9.7 (out of mainstream maintenance)

Description	Links
Running an SAP System on IBM Db2 with the Db2 pureScale Feature	Db2 11.5 Db2 11.1 Db2 10.5 Db2 10.1 (out of mainstream maintenance)
Enabling SAP Business Warehouse Systems to Use IBM Db2 for Linux, UNIX, and Windows as Near-Line Storage (NLS)	http://help.sap.com/viewer/db6_nls
IBM Db2 High Availability Solution: IBM Tivoli System Automation for Multiplatforms	https://help.sap.com/viewer/db6_samp
Central access to SAP Notes	Find SAP Notes on SAP Support Portal
Central access to all Db2 for LUW-related guides (including the above mentioned)	SAP on IBM Db2 overview page

SAP on Db2 for Linux, UNIX, and Windows Community

Check out and participate in our SAP community for IBM Db2. Here you'll find blog posts, Q&As, white papers, videos, and guides.

IBM Db2 Documentation

For IBM product documentation on your Db2 version, go to the IBM Db2 documentation on the IBM website.

A.2 Glossary

This glossary defines terms used in this document or used by support personnel in connection with SAP on IBM Db2 for Linux, UNIX, and Windows. If appropriate, it also includes links to other parts of this documentation, which describe the term in more detail.

Term	Description
`db2sap`	Shared library It is shipped with IBM Db2 for Linux, UNIX, and Windows and contains SAP-specific UDFs and stored procedures.
`<DBSID>` and `<dbsid>`	Database name
`<SAPSID>` and `<sapsid>`	SAP system ID The SAP system IDs and database names may differ. Therefore, you need to differentiate between `<sapsid>` and `<dbsid>`. **⚠ Caution** The IDs and the SAP database names are case-sensitive. This also applies to user IDs and groups (`db2<dbsid>`, `<sapsid>adm`, `sapr3`, `sap<sapsid>`, `sap<sapsid>db`, `SAPservice<SAPSID>`) as well as directory names.
archiving archival	Moving or copying files to a different, long-term storage device, assuming that the files are less likely to be lost there in the case of a system failure. Not to be confused with *backup*, the opposite of *retrieving* or *retrieval*. For more information, see Components of the Db2 Log File Management [page 79].
back end	Target to which files are archived to, such as tape, TSM, or a vendor product.
backup	Action of storing the database in a form that will allow it to be recovered (restored) later.
Db2 Database Manager	Db2 software that controls a database instance and its databases.
DB6	SAP's internal short name for the IBM Db2 for Linux, UNIX, and Windows (LUW) platform
ESE	Product name that refers to the IBM Db2 Enterprise Server Edition for Linux, UNIX, and Windows.
`ini` file `init<SAPSID>.db6`	The `init<SAPSID>.db6` file contains environment variables used by the `brdb6brt` tool for tasks such as turning on tracing.

Term	Description
log directory	Directory where Db2 stores log files, usually `/db2/<DBSID>/log_dir/NODExxxx`. This is a database parameter (`db cfg`) defined as “Path to log files”.
log file	File generated by Db2 to keep track of changes made to the database for recovery and rollback purposes.
password file	Refers to the `dscdb6.conf` file that contains encrypted passwords. Contents are set using the `dscdb6up` utility. For more information, see Managing Passwords [page 31].
restore	Action of restoring the database from a backup, for example, after a system failure or to generate a database copy. A restore often requires a database rollforward afterwards.
retrieving retrieval	Moving or copying files back to disk from long-term storage. Usually, this is only necessary after a system failure. Not to be confused with *restore*, retreiving is the opposite of *archiving*.
retrieve directory	Directory where `brrestore` stores log files, usually `/db2/<DBSID>/log_retrieve`. It is defined in the `ini` file as `DB2DB6_RETRIEVE_PATH`.
rollforward	Extracting database transaction data from log files After a restore operation, this data is added to a database to bring it up-to-date.
SAPTOOLS	Database schema for SAP extensions (UDFs, stored procedures, and tables) that are database-related and that do not depend on the SAP system.
TSM	IBM Tivoli Storage Manager (IBM storage product) As of Version 7.1.3, IBM Tivoli Storage Manager has been renamed to *IBM Spectrum Protect*.

A.3 Disclaimer

By following links to IBM Documentation you are leaving the SAP product documentation and entering a site that is not hosted by SAP. By using the link, YOU AGREE that unless expressly stated otherwise in your agreements with SAP you are about to access an external webpage which is not part of SAP's offering:

(i) the content of the linked-to site and any further external site is not product documentation and you may not infer any product documentation claims against SAP based on this information;

(ii) the fact that SAP provides links to external sites does not imply that SAP agrees or disagrees with the contents and information provided on such sites. SAP does not guarantee the correctness of the information provided.

(III) SAP DOES NOT GIVE ANY REPRESENTATION REGARDING THE QUALITY, SAFETY, SUITABILITY, ACCURACY OR RELIABILITY OF ANY EXTERNAL WEBPAGE OR ANY OF INFORMATION, CONTENT AND MATERIALS PROVIDED THEREON;

(IV) YOU VISIT THOSE EXTERNAL WEBPAGES ENTIRELY AT YOUR OWN RISK. SAP SHALL NOT BE DIRECTLY OR INDIRECTLY RESPONSIBLE OR LIABLE FOR ANY DAMAGE OR LOSS CAUSED OR ALLEGED TO BE CAUSED BY OR IN CONNECTION WITH YOUR USE OF OR RELIANCE ON ANY CONTENT, GOODS OR SERVICES AVAILABLE ON OR THROUGH ANY SUCH LINKED WEBPAGE.

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.

www.sap.com/contactsap

THE BEST RUN SAP