

DB2 System Command Examples

Michael Dang
Certified Database Administrator for DB2 UDB

Sylvia Qi
Certified Database Administrator for DB2 UDB

The purpose of this document is to provide examples for each DB2 system command, because these examples may be lacking in the DB2 Command Reference. However, the document is not intended to replace the Command Reference. The Command Reference still contains the most complete information about syntax and available options for each command.

The DB2 system commands must be invoked from the Operating System command prompt on Linux/UNIX, or from the DB2 Command prompt on Windows. The DB2 Command prompt on Windows can be launched by the 'db2cmd' command, or simply by opening the DB2 Command Window from the Start menu. The DB2 system commands are not valid in the DB2 interactive CLP session.

The commands in this document are applicable to DB2 version 8.

dasauto (UNIX only)
Description: Enables or disables autostarting of the DB2 Administration Server (DAS)
Linux/UNIX Examples
Command Location: /usr/opt/db2_08_01/das/adm (AIX) Command Location: /opt/IBM/db2/V8.1/das/adm (Linux/UNIX) Authorization: dasadm
Example 1: To enable autostart on the DAS, issue: <code>dasauto -on</code>
Example 2: To disable autostart on the DAS, issue: <code>dasauto -off</code>

dasCRT (UNIX only)
Description: Creates a DB2 Administration Server.
Linux/UNIX Examples
Command Location: /usr/opt/db2_08_01/instance (AIX) Command Location: /opt/IBM/db2/V8.1/instance (Linux/UNIX) Authorization: Root
Example: To create a DAS user "db2as". <code>dasCRT -u db2as</code>
Notes
<ul style="list-style-type: none"> • In previous DB2 versions, this command was known as "dasicrt". • This command can create only 1 DAS per server. • The command must be issued from the command location by a root user.

dasdrop (UNIX only)
Description: Removes a DB2 Administration Server
Linux/UNIX Examples
Command Location: /usr/opt/db2_08_01/instance (AIX) Command Location: /opt/IBM/db2/V8.1/instance (Linux/UNIX) Authorization: Root
Example: To remove or drop a DAS user "db2as", issue:

dasdrop
Notes
<ul style="list-style-type: none"> • Before running “dasdrop”, make sure to log onto the DAS user “db2as” and stop the DAS as follows: <code>db2admin stop</code> • 2. See <code>db2admin</code> command for more details. • 3. The command must be issued from the command location by a root user.

dasmigr (UNIX only)
Description: Migrates the DB2 Administration Server (DAS) following version upgrade.
Linux/UNIX Examples
Command Location: <code>/usr/opt/db2_08_01/instance</code> (AIX) Command Location: <code>/opt/IBM/db2/V8.1/instance</code> (Linux/UNIX) Authorization: Root Example: To migrate DAS after version upgrade, issue: <code>dasmigr</code>
Notes
<ul style="list-style-type: none"> • Migrating the DAS requires that a tools catalog be created and activated for the DAS. • Refer to “<code>create tools catalog</code>” in SQL Reference for more details. • If tools catalog is not required, an alternative to migrating the DAS is to first drop the DAS (in older version) and recreating the DAS (in new version). • The command must be issued from the command location by a root user.

db2
Description: Command Line Processor (CLP) Invocation Command.
Windows/Linux/UNIX Examples
Command Location: <code>~\sqllib\misc</code> (Windows) Command Location: <code>/INSTHOME/sqllib/misc</code> (all UNIX) Authorization: None Example 1: To start the CLP, issue: <code>db2</code> Example 2: To issue db2 commands, invoke the CLP with the desired db2 command. <code>db2 backup db sample</code> Example 3: To issue SQL statements, invoke the CLP with the desired SQL statement on Windows. <code>db2 select * from sales</code> Example 4: Same statement as example 3, but issued on UNIX. A double quote (") is required for proper parsing of the statement with special characters. <code>db2 "select * from sales"</code> Example 5: Run a DDL script “ <code>cr8_db_sample.clp</code> ” by invoking the CLP. An example of the script would be: <code>connect to sample;</code> <code>select * from sales;</code> <code>db2 -tvf cr8_db_sample.clp</code>

<p>Example 6: Encountered SQL0805N in application. Determine SQL error and possible action plan. db2 ? sql0805n</p>
<p>Notes</p> <ul style="list-style-type: none"> • QUIT stops the command line processor. • TERMINATE also stops the command line processor, but removes the associated back-end process and frees any memory that is being used. • It is recommended that a TERMINATE be issued prior to every STOP DATABASE MANAGER (db2stop) command. • It may also be necessary for a TERMINATE to be issued after database configuration parameters have been changed, in order for these changes to take effect.

db2admin
Description: This utility is used to manage the DB2 Administration Server (DAS).
Windows Examples
<p>Command Location: /sqlib/bin Authorization: Local Administrator</p> <p>Example 1: To create DAS and associate it with existing account "db2as" and password "xyz", issue: db2admin create /USER: db2as /PASSWORD: xyz</p> <p>Example 2: To remove or drop the DAS, issue: db2admin drop</p>
Linux/UNIX Examples
<p>Command Location: /home/DASuser/das/bin Authorization: dasadm</p> <p>Example 1: To start the DAS, issue: db2admin start</p> <p>Example 2: To stop the DAS. Issue: db2admin stop</p> <p>Example 3: To set or change the user ID associated with the DAS to "db2as" with password "xyz", issue: db2admin setid db2as xyz</p>

db2adutl
Description: Utility to work with TSM Archived Images
Windows/Linux/UNIX Examples
<p>Command Location: \sqlib\misc (Windows) Command Location: ~/sqlib/misc (all UNIX) Authorization: None</p> <p>Example 1: To view all DB2 objects on TSM server for all databases, issue: db2adutl query</p> <p>Example 2: To view tablespace backups -or- full database backups only for all databases, issue db2adutl query tablespace</p>

-or-

```
db2adutl query full
```

Example 3:

To view all logs -or- view logs from S0000100.LOG to S0000150.LOG inclusive for database SAMPLE only, issue:

```
db2adutl query logs database sample
```

-or-

```
db2adutl query logs between S0000100.LOG and S0000150.LOG db sample
```

Example 4:

To retrieve a full backup image taken at timestamp 2003062811000 for database SAMPLE only to your current directory, issue:

```
db2adutl extract full taken at 2003062811000 db sample
```

Example 5:

To retrieve a loadcopy image including images that are deactivated for database SAMPLE, issue:

```
db2adutl extract loadcopy show inactive db sample
```

Example 6:

To deactivate tablespace backups older than 90 days for all databases without asking for confirmation before deactivating, issue:

```
db2adutl delete tablespace older than 90 days without prompting
```

Example 7:

To delete logs between S0000100.LOG and S0000150.LOG for database SAMPLE, issue:

```
db2adutl delete logs between S0000100.LOG and S0000150.LOG db sample
```

Example 8:

To perform consistency checking on a full backup image taken at timestamp 2003062811000 for database SAMPLE, issue:

```
db2adutl verify full taken at 2003062811000 db sample
```

Notes

“VERIFY” parameter will transfer the backup image over the network before doing the consistency check.

db2advis

Description: DB2 Index Advisor Command. Advises users on what indexes to create for one or more SQL statements. A group of related SQL statements is known as a *workload*. Users can rank the importance of each statement in a workload, and specify the frequency at which each statement in the workload is to be executed. The recommended indexes for each table, the statistics derived for them, and the DDL by which each can be created, are written to a user-created table, ADVISE_INDEX.

Windows/Linux/UNIX Examples

Authorization: Read access to the database. Read and write access to the explain tables.

Example 1:

You want an index recommendation for table employee in database SAMPLE with no constraints. Use SQL statement “select * from employee a where a.empno in ('000160','000170','000180')”.

```
db2advis -d sample -s "select * from employee a where a.empno in ('000160','000170','000180')"
```

Example 2:

You want an index recommendation for workload name of “performance” on database SAMPLE with following constraints:

- maximum of 50MB for all indexes created.
- a solution must be found within 10 minutes.
- using user ID "db2admin" with password "xyz"
- saves the script to create indexes in output file db2advis.out

```
db2advis -d sample -w performance -l 50 -t 10 -a db2admin/xyz -o db2advis.out
```

Example 3:

You want an index recommendation for SQL statements in input file db2advis.in on database SAMPLE with the following constraints

- a solution must be found within 20 minutes.
- saves the script to create indexes in output file db2advis.out

```
db2advis -d sample -i db2advis.in -t 20 -o db2advis.out
```

Notes

1. You must first create table ADVISE_INDEX that mimics table you want the index advice on.
2. You must also create the explain tables from ~/sqlib/misc/EXPLAIN.DDL.
3. You must populate table ADVISE_WORKLOAD first with list of SQL statements to optimize under the workload name of "performance".
4. The input file "db2advis.in" must contain SQL statements and a specification of frequency of execution. For example:


```
-- SET FREQUENCY 50
select count(*) from employee;
select * from employee where workdept = 'D21';
-- SET FREQUENCY 5
select * from SAMPLE ;
```

db2atld

Description: Autoloader Command. Autoloader is a tool for partitioning and loading data in an MPP (formally EEE) environment.

Windows/Linux/UNIX Examples

Please see LOAD command for details.

db2audit

Description: Audit Facility Administrator Tool Command. DB2 provides an audit facility to assist in the detection of unknown or unanticipated access to data. The DB2 audit facility generates and permits the maintenance of an audit trail for a series of predefined database events.

Windows/Linux/UNIX Examples

Authorization: None

Example 1:

Display the current audit configuration and status.

```
db2audit describe
```

Example 2:

Set the default configuration and display the settings.

```
db2audit configure reset
db2audit describe
```

Example 3:

Start and stop the **db2audit** utility

```
db2audit start
db2audit stop
```

Example 4:

Auditing user IDs attempting to connect to the database SAMPLE, and determining what objects

they are trying to access.

```
db2audit stop
db2audit configure scope checking
db2audit start
```

Example 5:

Tracking user IDs being granted or revoked authorizations or privileges on objects in the database SAMPLE. Make sure to log both successful commands and errors.

```
db2audit stop
db2audit configure scope secmaint status both
db2audit describe
db2audit start
```

Example 6: Tracking all activities, including context, on the database SAMPLE. Log only errors.

```
db2audit stop
db2audit configure scope all, context status failure
db2audit start
```

Example 7:

A new user ID has been granted DBADM on database SAMPLE. You're interested in who granted the authority and when it was granted. Dump the security maintenance information to a flat file called "audit.out"

```
db2audit extract file audit.out category secmaint database sample
```

Example 8:

Dump all **db2audit** information for all databases into individual delimited ASCII files. Flush all information that may still be in memory to disk before extracting the file.

```
db2audit flush
db2audit extract delasc
```

Example 9:

You found the "db2audit.log" file is growing too large. Prune all entries before timestamp 2003062811.

```
db2audit prune date 2003062811
```

Notes

The db2audit information is written to a binary file "db2audit.log" located in the ~/sqllib/security directory.

db2batch

Description: Benchmark tool. Reads SQL statements and dynamically prepares and describes the statements.

Windows/Linux/UNIX Examples

Command Location: \sqllib\misc (Windows)

Command Location: ~/sqllib/misc (all UNIX)

Authorization: same authority as required for SQL statement to be read

Assume:

Input file "db2batch.sql":

SQL statement: "select lastname, firstme, deptname, count(*) from employee"

Example 1:

Determine how long SQL statement will take to run (p 1). No fetched rows will be sent to output (r 0). Output sent to stdout.

```
db2batch -d sample -f db2batch.sql -o p 1 r 0
```

Example 2:

Typical db2batch requested by DB2 Service Team. Gathers all available snapshots and monitoring information (p5), optimizer level your query is running (o 5), 1 row fetched will be sent to output (r1), gathers elapsed time intervals for prepare, execute and fetch phase (-i complete). Output sent to file "output.txt".

```
db2batch -d sample -f db2batch.sql -i complete -o p 5 o 5 r 1 -r
output.txt
```

Example 3:

Same output as Example 2. Set control options in db2batch.sql:

```
--#SET PERF_DETAIL 5 ROWS_OUT 1
```

```
select lastname, firstnme, deptname, count(*)from employee;
```

```
db2batch -d sample -f db2batch.sql -i complete -o o 5 -r output.txt
```

Example 4:

Gather a summary of monitoring information (p3), fetch all rows (f -1), output all rows (r -1), determine time to open cursor, complete the fetch, and close cursor (-i short). Output sent to file "output.txt"

```
db2batch -d sample -f db2batch.sql -i short -o p 3 f -1 r -1 -r
output.txt
```

Example 5:

Gathers all available snapshots and monitoring information (p 5), 5 rows fetched will be sent to output (r 5), populate the explain tables and run query (e 2). Output sent to file "output.txt"

```
db2batch -d sample -f db2batch.sql -o p 5 r 5 e 2 -r output.txt
```

Example 6:

Gather statistics of SQL statement in CLI mode. Output sent to file "output.txt"

```
db2batch -d sample -f db2batch.sql -cli -r output.txt
```

Notes

Statements in input file must be terminated by a delimiter (default is semicolon).

db2bfd

Description: Bind File Descriptor Tool. Displays the contents of a bind file.

Windows/Linux/UNIX Examples

Authorization: None.

Example 1:

Display bind file header information for bind file "db2sampl.bnd" to determine the isolation level.

```
db2bfd -b db2sampl.bnd
```

Example 2:

Display the SQL statements within bind file "db2sampl.bnd"

```
db2bfd -s db2sampl.bnd
```

Example 3:

Display the host variable declarations used in bind file "db2sampl.bnd"

```
db2bfd -v db2sampl.bnd
```

db2cap

Description: CLI/ODBC Static Package Binding Tool command. Binds a capture file to generate one or more static packages.

Windows/Linux/UNIX Examples

Authorization: Select privileges on any database objects referenced by SQL statements in capture file

-or-

BINDADD authority if package bound for the first time; otherwise, BIND authority is required.

Example 1:

Capture file is "db2capture.sql". Bind statements in capture file to generate packages in database SAMPLE.

```
db2cap bind db2capture.sql -d sample
```

Example 2:

Same as example 1, but user ID "user1" and password "xyz" is required.

```
db2cap bind db2capture.sql -d sample -u user1 -p xyz
```

Notes

A capture file is generated from a static profiling session of a CLI/ODBC/JDBC application. It contains SQL statements that were captured during the application run.

db2cc

Description: Starts the DB2 Control Center. The Control Center is a graphical interface that is used to manage database objects (such as databases, tables, and packages) and their relationship to one another.

Windows Examples

Authorization: sysadm.

Example 1:

Start the Control Center.

```
db2cc
```

Example 2:

Start the Control Center with trace option for problem determination. Output file sent to c:\trace\trace.txt.

```
db2cc -tf c:\trace\trace.txt
```

Example 3:

Start the Command Center.

```
db2cc -ccf
```

Example 5:

Start the Health Centre

```
db2cc -hc
```

Example 5:

Start the Control Center, but close any idle connections after 15 minutes.

```
db2cc -ict 15
```

db2cfexp

Description: Connectivity Configuration Export Tool. Exports all connectivity configuration information to a configuration profile, which can be later imported using the db2cfimp tool.

Windows/Linux/UNIX Examples

Authorization: sysadm -or- syctrl.

Example 1:

Generate configuration profile "template_cfg.txt" of current environment to be used as a template on other instances of the same instance type.

```
db2cfexp template_cfg.txt template
```

Example 2:

Generate configuration profile "backup_cfg.txt" of current environment for local backup purposes.
db2cfexp backup_cfg.txt backup

Example 3:

Generate configuration profile "maintain_cfg.txt" of current environment for maintaining or updating other instances.

```
db2cfexp maintain_cfg.txt maintain
```

Notes

- The backup option is an effective way to save your connectivity information in case of an environment corruption.
- A configuration profile may contain connectivity items such as:
 - Database information
 - Node information
 - Protocol information
 - DBM CFG settings
 - UDB registry settings
 - Common ODBC/CLI settings

db2cfimp

Description: Connectivity Configuration Import Tool – Imports connectivity configuration information from a configuration profile.

Windows/Linux/UNIX Examples

Authorization: sysadm -or- syctrl.

Example 1:

Created a new instance that will have same connectivity information as another instance saved in "template_cfg.txt" configuration profile.

```
db2cfimp template_cfg.txt
```

Example 2:

The catalog information was lost due to disk errors. Can retrieve the catalog information from the backup configuration profile previously saved in "backup_cfg.txt".

```
db2cfimp backup_cfg.txt
```

db2cidmg

Description: Remote Database Migration. Supports remote unattended migration in the Configuration, Installation, and Distribution (CID) architecture environment.

Windows/Linux/UNIX Examples

Authorization: sysadm -or- dbadm.

Example 1:

Migrate local database SAMPLE. Error log "db2cidmg.log" will be created in current directory where command was performed.

```
db2cidmg sample
```

Example 2:

Migrate all databases provided in response file "db2respfl.txt" and log error in path "~/db2mig/db2cidmg.log".

```
db2cidmg /r=db2respfl.txt /l1=~ /db2mig/db2cidmg.log
```

Example 3:

Migrate all databases cataloged in system directory. Log errors to path "~/db2mig/db2cidmg.log". Rebind all packages after migration is complete.

```
db2cidmg /e /l1=~ /db2mig/db2cidmg.log /b
```

Notes

Regardless if database is local or remote, the database alias must be cataloged on the target workstation.

db2ckbkp

Description: Check Backup Command. Used to test if backup image can be restored and also to display the meta-data in the backup header.

Windows/Linux/UNIX Examples

Authorization: read permission on backup image.

Assume:

Backup image: SAMPLE.0.db2v8.NODE0000.CATN0000.20030628110000.001

Example 1:

Validate the integrity of backup image. Will return "Image Verification Complete – successful" if backup image is good. Will return "ERROR - Unable to migrate media header from image. ERROR: Failed to verify media header. Cannot continue" if backup image is corrupt.

```
db2ckbkp SAMPLE.0.db2v8.NODE0000.CATN0000.20030628110000.001
```

Example 2:

Display all available information in backup image.

```
db2ckbkp -a SAMPLE.0.db2v8.NODE0000.CATN0000.20030628110000.001
```

Example 3:

Check online backup image for logs required for rollforward.

```
db2ckbkp -a SAMPLE.0.db2v8.NODE0000.CATN0000.20030628110000.001 | grep "File Number"
```

Example 4:

Display all Log File Header information

```
db2ckbkp -l SAMPLE.0.db2v8.NODE0000.CATN0000.20030628110000.001
```

Example 5: Determine the database codepage from backup image on tape (UNIX)

```
db2ckbkp -h /dev/rmt0 | grep "DB codeset"
```

Example 6: Determine the database seed from backup image on tape (WINDOWS)

```
db2ckbkp -a \\.\tape1 | grep "Database seed"
```

Notes

Regardless if database is local or remote, the database alias must be cataloged on the target workstation.

db2ckmig

Description: Database Pre-migration Tool. Verifies that a database can be migrated.

Windows/Linux/UNIX Examples

Command Location: \db2\Windows\utilities on DB2 v8 CD-ROM (Windows)

Command Location: /usr/opt/db2_08_01/instance (AIX)

Command Location: /opt/IBM/db2/V8.1/instance (Linux/UNIX)

Authorization: sysadm.

Example 1:

Verify that database SAMPLE is not in an inconsistent state; SAMPLE not in backup pending state; SAMPLE is not in rollforward pending state; tablespaces are in normal state. Output errors to /tmp/db2ckmig.log.

```
db2ckmig sample -l /tmp/db2ckmig.log
```

<p>Example 2: Verify that all local databases cataloged is ready for migration. Output errors to /tmp/db2chmig.log. <code>db2ckmig -e -l /tmp/db2chmig.log</code></p>
<p>Notes</p> <p>Regardless if database is local or remote, the database alias must be cataloged on the target workstation.</p>

db2ckrst
<p>Description: Check Incremental Restore Image Sequence. Generates a list of backup image timestamps required for an incremental restore. Also generates a simplified syntax for a manual incremental restore.</p>
Windows/Linux/UNIX Examples
<p>Command Location: \db2\Windows\utilities on DB2 v8 CD-ROM Command Location: /usr/opt/db2_08_01/instance (AIX) Command Location: /opt/IBM/db2/V8.1/instance (Linux/UNIX) Authorization: None.</p> <p>Example 1: Assume full backup taken at 20030628110000; incremental backup taken at 200306281103045; another incremental backup taken at 20030628110838. Determine the restore order for most recent incremental backup (timestamp 20030628110838). <code>db2ckrst -d sample -t 20030628110838</code></p> <p>Assume output: Suggested restore order of images using timestamp 20030628110838 for database sample</p> <pre>===== db2 restore db sample incremental taken at 20030628110838 db2 restore db sample incremental taken at 20030628110000 db2 restore db sample incremental taken at 20030628110345 db2 restore db sample incremental taken at 20030628110838 =====</pre> <p>Example 2: Assume full backup taken at 20030628110000; incremental backup taken at 200306281103045. Determine the restore order required to restore the tablespace "tbspdef". <code>db2ckrst -d sample -t 20030628110345 -r tablespace -n tbspdef</code></p>
Notes
<p>The database history must exist in order for this utility to be used.</p>

db2cmd (Windows only)
<p>Description: Opens the DB2 Command Window.</p>
Windows Examples
<p>Command Location: \SQLLIB\bin\ Authorization: None.</p> <p>Example 1: Invoke db2cmd to call a script "E:\scripts\sample\bkupsample.bat" to perform an online backup for database SAMPLE. Make sure command window will not terminate until the task is ended. <code>db2cmd /c /w E:\scripts\sample\bkupsample.bat</code></p> <p>Example 2: Invoke db2cmd to display the Db2 database manager parameters and pipe the output to a file "myoutput".</p>

```
db2cmd -i db2 get dbm cfg > myoutput
```

db2dart

Description: DB2 Analysis and Reporting Tool. Examines databases for architectural correctness and reports any encountered errors.

Windows/Linux/UNIX Examples

Authorization: sysadm.

Example 1:

Use **db2dart** to inspect entire database SAMPLE.

```
db2dart sample
```

Example 2:

Use **db2dart** to display suggestions on lowering the high-water mark as much as possible for tablespace 6 in database SAMPLE. Required to provide DMS tablespace ID (6) and desired highwater mark (0 pages).

```
db2dart sample /LHWM /TSI 6 /NP 0
```

Example 3:

Use **db2dart** to inspect integrity of table ID 7 in the tablespace "userspace1". Required to provide the tablespace ID and table ID or name.

```
db2dart sample /TN 7
```

Example 4:

Use **db2dart** to high water mark information for analysis. Redirect the output to a filename "sample_20030628.rpt" to directory ~/db2inst1/db2dart/. Required to provide the DMS tablespace ID.

```
db2dart sample /DHWM -RPT ~/db2inst1/db2dart/ -RPTN  
sample_20030628.rpt
```

Example 5:

Table ID 7 in database SAMPLE is corrupted between page 3 and 500. Use **db2dart** to dump the non-corrupted table data into a delimited ASCII format. Required to provide the table ID or name (7), tablespace ID (2), the first page dumped (500), and total number of pages to dump (high number – 9999999999999).

```
db2dart sample /DDEL
```

Example 6:

Index corruption found. Use db2dart to mark the index as invalid. Index will be recreate after first connect to database. Required to provide index ID (4) and tablespace ID (2).

```
db2dart sample /MI /TSI 2 /OI 4
```

Example 7:

Run db2dart on entire database sample, and only output the errors into report file.

```
db2dart sample /RPTF e
```

Notes

- The db2dart tool must run with no users connected to the database.
- All reports will be named <database>.RPT and located in current directory unless otherwise specified.

db2dclgn

Description: Declaration Generator Command. Generates declarations for a specified database table, eliminating the need to look up those declarations in the documentation. The generated declarations can be modified as necessary. The supported host languages are C/C++, COBOL, JAVA, and FORTRAN.

Windows/Linux/UNIX Examples

Command Location:

Command Location:

Authorization: none

Example 1:

```
db2dc1gn -d sample
```

db2drdat

Description: DRDA trace. Captures DRDA data stream exchanged between a DRDA Application Requestor (AR) and the DB2 UDB DRDA Application Server (AS)

Windows/Linux/UNIX Examples

Authorization: none.

Example 1:

Use DRDA trace to trace errors received from DB2 Connect.

```
db2drdat -on
```

Example 2:

Use DRDA trace to trace DRDA requests from the DRDA AR. Redirect the output file to “~/db2int1/drdatrc/drda_20030628.trc”.

```
db2drdat -on -r -t ~/db2inst1/drdatrc/drda-20030628.trc.
```

Example 3:

Use DRDA trace to trace SQLCA received from the DRDA server on the host system. Specify a buffer size of 8000000 bytes.

```
db2drdat -on -c -l 8000000
```

Example 4:

DB2 Connect performance problems. Use DRDA trace to track packets sent (-s) and received (-r) to and from the DRDA AR respectively. Include the timestamps in the trace information.

```
db2drdat -on -s -r -i
```

Example 5:

Multiple connections to DRDA AS. Use DRDA trace to trace errors found with process ID (PID) 3076.

```
db2drdat -on -p=3076
```

Example 6:

Turn off DRDA trace after required information is gathered.

```
db2drdat -off
```

Notes

If the tracefile is not specified, messages are directed to db2drdat.dmp in the current directory.

db2empfa

Description: Enable Multipage File Allocation. With multipage file allocation enabled for SMS tablespaces, disk space is allocated one extent rather than one page at a time.

Windows/Linux/UNIX Examples

Command Location:

Command Location:

Authorization: sysadm.

Example 1:

Enable multipage file allocation on database SAMPLE.

```
db2empfa sample
```

Notes
<ul style="list-style-type: none"> • Utility connects to database partition in exclusive mode. Cannot run concurrently on the catalog database partition, or on any other database partition. • Changes the value of the database configuration parameter “multipage_alloc” to YES.

db2eva
Description: Event Analyzer. Used to trace performance data produced by the DB2 event monitors that have their data directed to files.
Windows/Linux/UNIX Examples
Command Location: Command Location: Authorization: sysadm -or- sysctrl -or- sysmaint -or- dbadm.
Example 1: Analyze the event monitor traces located in “c:\DB2\event_monitors”. <code>db2eva -path c:\DB2\event_monitors\</code>
Example 2: Analyze the event monitor traces by connecting to database SAMPLE and run the event monitor “check_application_deadlocks”. <code>db2eva -evm check_application_deadlocks -db sample</code>
Notes
1. See System Monitor Guide and Reference for more information on event monitors.

db2evmon
Description: Event Monitor Productivity Tool Command.
Windows/Linux/UNIX Examples
Authorization: sysadm -or- sysctrl -or- sysmaint -or- dbadm if connecting to database; otherwise none.
Example 1: Format all event monitor files in present directory and print to screen. <code>db2evmon</code>
Example 2: Format the event monitor file “event_sample” for database sample and save to “event_sample.txt” file. <code>db2evmon -db sample -evm event_sample > event_sample.txt</code>
Example 3: Format all event monitor trace files located in “~/event/traces/”. <code>db2evmon -path ~/event/traces/</code>

db2evtbl
Description: Generate Event Monitor Target Table Definitions. Generates sample CREATE EVENT MONITOR SQL statements that can be used when defining event monitors that write to SQL tables.
Windows/Linux/UNIX Examples
Authorization: none.
Example 1: Request db2evtbl to generate SQL statement to monitor deadlocks in detail. <code>db2evtbl -evm check_application_deadlocks deadlocks with details</code>

Example 2:

Create an event monitor “condlmon” to monitor connections and deadlocks. Assume “user1” will be the owner of the event monitor.

```
db2evtb1 -schema user1 -evm condlmon connections, deadlocks
```

db2exfmt

Description: Explain Table-Format Tool. Used to format the contents of the explain tables.

Windows/Linux/UNIX Examples

Command Location: \sqllib\misc (Windows)

Command Location: ~/sqllib/misc (UNIX)

Authorization: select privilege on explain tables.

Example 1:

Format explain tables for database SAMPLE. Generate a graph plan with all format information. Output file “exfmt_bad.out”.

```
db2exfmt -d sample -g -o exfmt_bad.out
```

Example 2:

Format explain tables for database SAMPLE. Generate a graph plan, include total cost and I/O cost. Obtain the latest explain request and output directed to screen.

```
db2exfmt -d sample -g TI -w -1 -t
```

Example 3:

Typical db2exfmt output requested by DB2 Support.

```
db2exfmt -d sample -g TIC -w -1 -n % -s % -# 0 -o exfmt_bad.txt
```

db2expln

Description: Generate Event Monitor Target Table Definitions. Generates sample CREATE EVENT MONITOR SQL statements that can be used when defining event monitors that write to SQL tables.

Windows/Linux/UNIX Examples

Authorization: none.

Example 1:

Request db2evtb1 to generate SQL statement to monitor deadlocks in detail.

```
db2evtb1 -evm check_application_deadlocks deadlocks with details
```

Example 2:

Create an event monitor “condlmon” to monitor connections and deadlocks. “user1” will be the owner of the event monitor.

```
db2evtb1 -schema user1 -evm condlmon connections, deadlocks
```

db2flsn

Description: Returns the name of the file that contains the log record identified by a specified log sequence number (LSN).

Windows/Linux/UNIX Examples

Authorization: none.

Example 1:

Encountered crash recovery failure. Found the following error message in db2diag.log:

```
“Crash recovery failed with rc -2045771763  
LowtranLSN 000000A41EB2000C MinbuffLSN 000000A41D81000C  
Recovery started on log file S0006632.LOG.”
```

Use **db2flsn** to determine if the log file indicated by the LSN are indeed pointing to

S0006632.LOG.

```
db2flsn -q 000000A41EB2000C
db2flsn -q 000000A41D81000C
```

Example 2:

Determine if the above LowtranLSN has any errors or warning associated with it.

```
db2flsn 000000A41EB2000C
```

Notes

- The log header control file SQLOGCTL.LFH must reside in the current directory.
- This utility can only work with logretain and/or userexit activated.

db2fm (UNIX only)

Description: DB2 Fault Monitor. Controls the DB2 fault monitor daemon. Used to configure the fault monitor.

Windows Examples

Not applicable (command unavailable in Windows platform)

Linux/UNIX Examples

Command Location: ~/sqllib/
Authorization: Instance owner.

Example 1:

To turn on fault monitoring for instance DB2INST1, issue:

```
db2fm -i db2inst1 -f yes
```

Example 2: To turn off fault monitoring for instance DB2INST1, issue:

```
db2fm -i db2inst1 -f no
```

Example 3: To update the START_TIMEOUT value to 100 seconds and the STOP_TIMEOUT value to 200 seconds for instance DB2INST1, issue:

```
db2fm -i db2inst1 -T 100/200
```

Example 4: To determine the status of the fault monitor service and fault monitor daemon respectively, issue:

```
db2fm -s -S
```

Example 5: To bring up the fault monitor service for instance DB2INST1, issue:

```
db2fm -i db2inst1 -u
```

Example 6: To update the STATUS_INTERVAL value to 60 seconds for instance DB2INST1, issue:

```
db2fm -i db2inst1 -I /60
```

Example 7: To update the NOTIFY_ADDRESS to send errors to your email address

db2_dba@ibm.com, issue:

```
db2fm -n db2\_dba@ibm.com
```

Notes

db2fm can be used to update the fm.<machine_name>.reg

db2fs

Description: First Steps. Launches the First Steps GUI which contains links to the functions users need to begin learning about and using DB2.

Windows/Linux/UNIX Examples

Command Location: DB2PATH\bin (Windows)

Command Location: ~/sqllib/bin (UNIX)

Authorization: sysadm.

Example 1:
Start the First Steps.
db2fs

db2gncol

Description: Update Generated column Values Command. Updates generated columns in tables that are in check pending mode and have limited log space. This tool is used to prepare for a SET INTEGRITY statement on a table that has columns which are generated by expressions.

Windows/Linux/UNIX Examples

Command Location: \sqllib\bin (WINDOWS)
Command Location: ~/sqllib/bin (UNIX)
Authorization: sysadm -or- dbadm.

Example 1: Use db2gncol to regenerate column values for user1.table1 in database SAMPLE. Commit every 500 rows updated.

```
db2gncol -d sample -s user1 -t table1 -c 500
```

Example 2: Execute example 1 with a valid ID (USER1) with sysadm or dbadm authority.

```
db2gncol -d sample -s user1 -t table1 -c 500 -u user1 -p xyz
```

Notes

1. Using this tool instead of the FORCE GENERATED option on the SET INTEGRITY statement may be necessary if a table is large and the following conditions exist:

- All column values must be regenerated after altering the generation expression of a generated column.
- An external UDF used in a generated column was changed, causing many column values to change.
- A generated column was added to the table.
- A large load or load append was performed that did not provide values for the generated columns.
- The log space is too small due to long-running concurrent transactions or the size of the table.

db2gov

Description: DB2 Governor. Monitors and changes the behavior of applications that run against a database.

Windows/Linux/UNIX Examples

Command Location: \sqllib\bin (WINDOWS)
Command Location: ~/sqllib/bin (UNIX)
Authorization: sysadm -or- sysctrl.

Example 1:

Start **db2gov** for database SAMPLE using the “govcfigsample” configuration file located in ~/sqllib/ and write the output to “govcfigsample.log”.

```
db2gov start sample govcfigsample govcfigsample.log
```

Example 2:

Start **db2gov** as above, but this time, only monitor the activity against the database partition at node 2.

```
db2gov start sample nodenum 2 govcfigsample govcfigsample.log
```

Example 3:

Stop **db2gov** started in example 2.

```
db2gov stop sample nodenum 2
```

Example 4:

Stop **db2gov** started in example 1.

```
db2gov stop sample
```

Example 5:

It is a good idea to specify the nodenum, even if you only have 1 partition (EE). Otherwise, you may receive the following error:

“rshd: 0826-813 Permission is denied.”

```
db2gov start sample nodenum 0 govcfgsample govcfgsample.log
```

Notes

- The default location for the configuration file is the sqllib directory.
- If your rule requirements change, you edit the configuration file without stopping the governor. Each governor daemon detects that the file has changed, and rereads it.
- The configuration file must be created in a directory that is mounted across all the database partitions so that the governor daemon on each partition can read the same configuration file.
- For more information on the configuration file setup, see the Administration Guide.
- The default location for the log file is sqllib/log directory.

db2govlg

Description: DB2 Governor Log Query. Extracts records of specified type from the governor log files.

Windows/Linux/UNIX Examples

Authorization: None.

Example 1:

Extract all record types from the db2gov log.

```
db2govlg govcfgsample.log
```

Example 2:

Extract all records relating to the action “force” from the db2gov log.

```
db2govlg govcfgsample.log rectype FORCE
```

Example 3:

Extract all records relating to any “errors” from the db2gov log on partition 1.

```
db2govlg govcfgsample.log nodenum 1 rectype ERROR
```

Notes

Can only request one record type at a time if the RECTYPE option is used.

db2hc

Description: DB2 Health Center. The Health Center is a graphical interface that is used to view the overall health of the database systems.

Windows/Linux/UNIX Examples

Authorization: No authority required to view. Appropriate authority required to take action.

Example 1:

Start the Health Centre GUI.

```
db2hc
```

Example 2:

DB2 Support requires a full **db2hc** trace for problem determination.

```
db2hc -t
```

Example 3:

DB2 Support requires to trace communication events for problem determination.

```
db2hc -tcomms
```

db2iclus (Windows only)

Description: Microsoft Cluster Server. Allows users to add, drop, migrate and unmigrate instances and DB2 Administration Servers (DAS) in a Microsoft Cluster Server (MSCS) Environment.

Windows Examples

Command Location: \sqlib\bin

Authorization: Local Administrator.

Example 1:

Manually configure the DB2 instance to run in a hot standby configuration that consists of two machines, WA26 and WA27.

To start, MSCS and DB2 UDB Enterprise Server Edition must be installed on both machines.

1. Create a new instance called DB2 on machine WA26:

```
db2icrt DB2
```

2. From the Windows Services dialog box, ensure that the instance is configured to start manually.

3. If the DB2 instance is running, stop it with the DB2STOP command.

4. Install the DB2 resource type from WA26:

```
db2wolfi i
ok
```

5. If the db2wolfi command returns "Error : 183", then it is already installed. To confirm, the resource type can be dropped and added again. Also, the resource type will not show up in Cluster Administrator if it does not exist.

```
db2wolfi u
ok
```

```
db2wolfi i
ok
```

6. From WA26, use the db2iclus command to transform the DB2 instance into a clustered instance.

```
db2iclus migrate /i:db2 /c:mycluster /m:wa26 /p:p:\db2profs
```

```
DBI1912I The DB2 Cluster command was successful.
```

```
Explanation: The user request was successfully processed.
```

```
User Response: No action required.
```

Note:

The directory p:\db2profs should be on a clustered drive and must already exist. This drive should also be currently owned by machine WA26.

1. From WA26, use the db2iclus command to add other machines to the DB2 cluster list:

```
db2iclus add /i:db2 /c:mycluster /m:wa27
```

```
DBI1912I The DB2 Cluster command was successful.
```

```
Explanation: The user request was successfully processed.
```

```
User Response: No action required.
```

This command should be executed for each subsequent machine in the cluster.

8. From Cluster Administrator, create a new group called "DB2 Group".

9. From Cluster Administrator, move the Physical Disk resources Disk O and Disk P into DB2 Group.

10. From Cluster Administrator, create a new resource type of type "IP Address" called "mcs5" that resides on the Public Network. This resource should also belong to DB2 Group. This will be a

highly available IP address, and this address should not correspond to any machine on the network. Bring the IP Address resource type online and ensure that the address can be pinged from a remote machine.

11. From Cluster Administrator, create a new resource of type "DB2" that will belong to DB2 Group. The name of this resource must be exactly identical to the instance name, so it is called DB2 for this case. When Cluster Administrator prompts for dependencies associated with the DB2 resource, ensure it is dependent on Disk O, Disk P and mscc5.

12. Configure DB2 Group for fallback, if desired, via Cluster Administrator and using the DB2_FALLBACK profile variable.

13. Create or restore all databases putting all data on Disk O and Disk P.

14. Test the failover configuration.

Linux/UNIX Examples

Not applicable (command unavailable on UNIX platforms)

db2icrt

Description: Creates DB2 instances.

Windows Examples

Command Location: \sqllib\bin

Authorization: Local administrator authority.

Example 1:

Create a new instance DB2INST1 with default settings.

```
db2icrt db2inst1
```

Example 2:

Create a new client instance DB2INST1 with default settings.

```
db2icrt -s client db2inst1
```

Example 3:

Create a new instance DB2INST1 on a partitioned server, specify TCP/IP ports to be used from 50000 to 50050 inclusive. Create the instance on second hostname HOST1 with account name "user1" and password "xyz"

```
db2icrt -s ese -u user1, xyz -h host1 -r 50000,50050 db2inst1
```

Linux/UNIX Examples

Command Location: /usr/opt/db2_08_01/instance (AIX)

Command Location: /opt/IBM/db2/V8.1/instance (Linux/UNIX)

Authorization: Root.

Example 1:

Create a new instance DB2INST1 with default settings.

```
db2icrt db2inst1
```

Example 2:

Create a new client instance DB2INST1 with default settings.

```
db2icrt -s client db2inst1
```

Example 3:

Create a new instance DB2INST1 with fenced ID "db2fenc1". Use authentication SERVER encryption and instance type ESE.

```
db2icrt -a server_encrypt -s ese -u db2fenc1 db2inst1
```

Example 4:

Create a new 64-bit instance DB2INST1 with fenced ID "db2fenc1". Use authentication SERVER and instance type ESE.

```
db2icrt -a server -s ese -w -u db2fenc1 db2inst1
```

Notes
<ul style="list-style-type: none"> 64-bit instance option is only valid on AIX 5L, HP-UX and Solaris. On Linux/UNIX, the command must be issued from the command location by a root user.

db2idrop
Description: Remove or drop DB2 instance.
Windows Examples
Command Location: \sqllib\bin Authorization: Local Administrator. Example 1: Drop the instance DB2INST1. <pre>db2idrop db2inst1</pre> Example 2: Drop instance DB2INST1 failed. There are applications still connected to databases. Force the applications before dropping instance. <pre>db2idrop -f db2inst1</pre>
Linux/UNIX Examples
Command Location: /usr/opt/db2_08_01/instance (AIX) Command Location: /opt/IBM/db2/V8.1/instance (Linux/UNIX) Authorization: Root Example 1: Drop the instance DB2INST1. <pre>db2idrop db2inst1</pre>
Notes
On Linux/UNIX, the command must be issued from the command location by a root user.

db2ilist
Description: List DB2 instances.
Windows/Linux/UNIX Examples
Command Location: \sqllib\bin Command Location: /usr/opt/db2_08_01/instance (AIX) Command Location: /opt/IBM/db2/V8.1/instance (Linux/UNIX) Authorization: Root or instance owner on UNIX. No authority required on Windows. Example 1: List the existing DB2 instances on the server. <pre>db2ilist</pre>

db2imigr (UNIX only)
Description: Migrate Instance. Migrates an existing instance following installation of the database manager.
Windows Examples
Not applicable (instance migration implicit during migration)
Linux/UNIX Examples
Command Location: /usr/opt/db2_08_01/instance (AIX) Command Location: /opt/IBM/db2/V8.1/instance (Linux/UNIX) Authorization: Root. Example 1: Migrate instance DB2INST1 with defaults. <pre>db2imigr db2inst1</pre>

Example 2:

Migrate instance DB2INST1 changing it to client authentication.

```
db2imigr -a client db2inst1
```

Example 3:

Migrate instance DB2INST1 using server authentication, and fenced id "db2fenc1".

```
db2imigr -a server -u db2fenc1 db2inst1
```

Example 4:

Migration of instance failed in example 3. Turn on debug mode for problem determination.

```
db2imigr -d -a server -u db2fenc1 db2inst1
```

Example 5:

Migrate Data Links File Manager instance DB2INST1 with defaults. Assume dlfxgrp ID is "dlfm".

```
db2imigr -g dlfm db2inst1
```

Notes

- Use debug mode only when instructed by DB2 Support.
- The command must be issued from the command location by a root user.

db2inidb

Description: Initialize a Mirrored Database. Initializes a mirrored database in a split mirror environment. The mirrored database can be initialized as a clone of the primary database, placed in roll forward pending state, or used as a backup image to restore the primary database.

Windows/Linux/UNIX Examples

Authorization: sysadm -or- sysctrl -or- sysmaint

Example 1:

Create a mirrored database SAMPLE2 that will be initialized as a clone of the primary database SAMPLE.

```
db2inidb sample2 as snapshot
```

Example 2:

Create a standby database SAMPLE2 that will be placed in roll forward pending state. The standby database can be used if the primary database goes down, by moving the new logs from the primary to the standby database, and rolling forward.

```
db2inidb sample2 as standby
```

Example 3:

Create a mirrored database SAMPLE2 that will be used as a backup image which can be used to restore the primary database SAMPLE.

```
db2inidb sample2 as mirror
```

Example 4:

Run example 3 and use a configuration file "relocatefile" with valid information to relocate the database.

```
db2inidb sample2 as mirror relocate using relocatefile
```

Notes

In a partitioned database environment, **db2inidb** must be run on every partition before the split mirror from any of the partitions can be used. **db2inidb** can be run on all partitions simultaneously using the **db2_all** command.

db2inspf

Description: This utility formats the data from INSPECT CHECK results into ASCII format. Use this utility to see details of the inspection. The formatting by the db2inspf utility can be format for

a table only, or a table space only, or for errors only, or summary only.
Windows/Linux/UNIX Examples
Authorization: must have read permission on datafile.
<p>Example 1: Format the data from "inspf.data" into a flat file "inspf.out" for all tables in tablespace "2". db2inspf inspf.data inspf.out -tsi 2</p> <p>Example 2: Format the data from "inspf.data" into a flat file "inspf.out" for table "8" in tablespace "2". db2inspf inspf.data inspf.out -tsi 2 -ti 8</p> <p>Example 3: Format the data from "inspf.data" into a flat file "inspf.out" for any errors. db2inspf inspf.data inspf.out -e</p> <p>Example 4: Format the data from "inspf.data" into a flat file "inspf.out" for a summary only. db2inspf inspf.data inspf.out -s</p>

db2isetup (UNIX only)
Description: Start Instance Creation Interface Command. Starts the DB2 Instance Setup wizard, a graphical tool for creating instances and for configuring new functionality on existing instances.
Windows Examples
Not applicable (instance migration implicit during migration)
Linux/UNIX Examples
<p>Command Location: /usr/opt/db2_08_01/instance (AIX) Command Location: /opt/IBM/db2/V8.1/instance (Linux/UNIX) Authorization: Root</p> <p>Example 1: Start the DB2 Instance Setup Wizard with defaults. The log file is located in default location "/tmp/db2isetup.log". db2isetup</p> <p>Example 2: Start the DB2 Instance Setup Wizard and log in new path "~/db2isetup.log". db2isetup -l ~/db2isetup.log</p> <p>Example 3: Error encountered starting db2isetup. Turn on tracing for problem determination. Log to "~/db2isetup.log". db2isetup -t -l ~/db2isetup.log</p>
Notes
This instance setup wizard provides a subset of the functionality provided by the DB2 Setup wizard. The DB2 Setup wizard (which runs from the installation media) allows you to install DB2 components, do system setup tasks such as DAS creation/configuration, and set up instances. The DB2 Instance Setup wizard only provides the functionality pertaining to instance setup.

db2iupdt
Description: Update DB2 instances. On UNIX, updates a specified DB2 instance to enable acquisition of a new system configuration or access to function associated with the installation or removal of certain product options. On Windows, updates single-partition instances for use in a partitioned database system.
Windows Examples

Command Location: \sqlib\bin
Authorization: Local Administrator.

Example 1:

Instance DB2INST1 was used as a single partition. Desired to move to MPP environment. Add 3 more partition on the server . Use TCP/IP port ranges from 50000 to 50003.

```
db2iupdt db2inst1 /r:50000,50003
```

Example 2:

Desired to add 4 more partitions to the server in example 1. The new partitions will be running on a second hostname HOST2 on the same server.

```
db2iupdt db2inst1 /r:50004,50007 /h:host2
```

Linux/UNIX Examples

Command Location: /usr/opt/db2_08_01/instance (AIX)

Command Location: /opt/IBM/db2/V8.1/instance (Linux/UNIX)

Authorization: Root.

Example 1:

Completed applying a DB2 Fixpak. Run **db2iupdt** to update instance DB2INST1 with fenced ID "db2fenc1".

```
db2iupdt -u db2fenc1 db2inst1
```

Example 2:

Run **db2iupdt** to migrate instance DB2INST1 and fenced ID "db2fenc1" to 64-bit.

```
db2iupdt -w -u db2fenc1 db2inst1
```

Example 3:

Run **db2iupdt** to change the authentication type from SERVER to SERVER_ENCRYPT.

```
db2iupdt -a server_encrypt db2inst1
```

Example 4:

Client instances will automatically be converted to a server instance after **db2iupdt**. To maintain the client instance type, issue:

```
db2iupdt -k db2inst1
```

Example 5:

Turn on debug mode if errors running **db2iupdt**.

```
db2iupdt -d -a server -u db2fenc1 db2inst1
```

Notes

- Use debug mode only when instructed by DB2 Support.
- On Linux/UNIX, the command must be issued from the command location by a root user.

db2jdbcbind

Description: DB2 JDBC Package Binder Command. This utility is used to bind or rebind the JDBC packages to a DB2 database.

Windows/Linux/UNIX Examples

Command Location: /usr/opt/db2_08_01/instance (AIX)

Command Location: /opt/IBM/db2/V8.1/instance (Linux/UNIX)

Authorization: sysadm -or- dbadm -or-

BINDADD -and- IMPLICIT_SCHEMA -and- CREATEIN -or-

ALTERIN -or- BIND.

Example 1:

Bind JDBC/ODBC packages to database SAMPLE; hostname "test.ibm.com"; port 50000; user ID "user1"; password "xyz".

```
db2jdbcbind -url jdbc:db2://test.ibm.com:50000/sample -user user1 -
password xyz
```

Example 2:

Bind example 1 with 20 internal packages.

```
db2jdbcbind -url jdbc:db2://test.ibm.com:50000/sample -user user1 -
password xyz size 20
```

Example 3:

Error binding example 1. Turn on tracing for all for problem determination.

```
db2jdbcbind -url jdbc:db2://test.ibm.com:50000/sample -user user1 -
password xyz -tracelevel trace_all
```

Notes

- DB2 Version 8 databases already have the JDBC packages preinstalled, therefore, this command is usually necessary only for downlevel servers.
- JDBC and CLI share the same packages. If the CLI packages have already been bound to a database, then it is not necessary to run this utility and vice versa.
- 3. The DB2 JDBC type 4 driver is used to establish the connection.

db2ldcfg

Description: Configure LDAP Environment. Configures the Lightweight Directory Access Protocol (LDAP) user distinguished name (DN) and password for the current logon in an LDAP environment using an IBM LDAP client.

Windows/Linux/UNIX Examples

Authorization: None.

Example 1:

Setup the IBM LDAP client to save the current LDAP user's DN and password. This will eliminate the need to specify the LDAP user's DN and password when issuing the LDAP command or API. The distinguished name consists of user ID USER1, and domain "mydomain.ibm.com"; password "xyz"

```
db2ldcfg -u "uid=user1,dc=mydomain,dc=ibm dc=com" -w xyz
```

Notes

Command can only be run when using an IBM LDAP client.

db2level

Description: Show DB2 Service Level. Shows the current version and service level of the installed DB2 product. Output from this command goes to the console by default.

Windows/Linux/UNIX Examples

Command Location: \sqllib\bin (Windows)

Command Location: /usr/opt/db2_08_01/instance (AIX)

Command Location: /opt/IBM/db2/V8.1/instance (Linux/UNIX)

Authorization: None.

Example 1:

Use db2level to determine the current DB2 Fixpak level you are running.

```
db2level
```

db2licm

Description: License Management Tool. Performs basic license functions in the absence of the Control Center. Adds, removes, lists, and modifies licenses and policies installed on the local system.

Windows/Linux/UNIX Examples

Command Location: \sqllib\bin (Windows)

Command Location: /usr/opt/db2_08_01/instance (AIX)

Command Location: /opt/IBM/db2/V8.1/instance (Linux/UNIX)
Authorization: Root required on UNIX. No authority required on Windows.

Example 1:

Display all DB2 products with available licenses installed.

```
db2licm -l
```

Example 2:

Just installed DB2 Enterprise Server Edition (ESE). Install the license file "db2ese.lic" located at ~/product/DB2/license/.

```
db2licm -a ~/product/DB2/license/db2ese.lic
```

Example 3:

Update the DB2 Workgroup Server Edition (WSE) license policy to reflect registered and concurrent users.

```
db2licm -p db2wse registered concurrent
```

Example 4:

Remove the license for DB2 ESE.

```
db2licm -r db2ese
```

Example 5:

Purchased additional 100 user licenses for DB2 WSE. Update the license policy accordingly.

```
db2licm -u db2wse 100
```

Example 6:

Encountered the following error: "SQL8017W The number of processors on this machine exceeds the defined entitlement of "1" for the product "DB2 Enterprise Server Edition". The number of processors on this machine is 2." To rectify, issue:

```
db2licm -n db2ese 2
```

db2logsforrwd

Description: List Logs Required for Rollforward Recovery Command. Parses the DB2TSCHG.HIS file. This utility allows a user to find out which log files are required for a table space rollforward operation.

Windows Examples

Command Location: \sqllib\bin (Windows)

Authorization: None.

Example 1:

There are over 100 logs required for a full database rollforward. You restored a tablespace backup and wish to only rollforward logs that contain the tablespace information. Use **db2logsforrwd** to determine the required logs for a tablespace rollforward.

```
db2logsforrwd c:/DB2/NODE0000/SQL00001/DB2TSCHG.HIS
```

Example 2:

You are in the database directory. You wish to display the full details of the logs required for a tablespace rollforward.

```
db2logsforrwd DB2TSCHG.HIS -all
```

Linux/UNIX Examples

Command Location: ~/sqllib/bin (UNIX)

Authorization: None.

Example 1:

There are over 100 logs required for a full database rollforward. You restored a tablespace backup and wish to only rollforward logs that contain the tablespace information. Use

db2logsforrwd to determine the required logs for a tablespace rollforward.

```
db2logsForRfwd /home/user1/user1/NODE0000/SQL00001/DB2TSCHG.HIS
```

Example 2:

You are in the database directory. You wish to display the full details of the logs required for tablespace rollforward.

```
db2logsForRfwd DB2TSCHG.HIS -all
```

Notes

- Use debug mode only when instructed by DB2 Support.
- 2. The db2logsForRfwd command is case sensitive on UNIX, but not on Windows.

db2look

Description: DB2 Statistics and DDL Extraction Tool. Extracts the required DDL statements to reproduce the database objects of a production database on a test database. This tool can also generate the required UPDATE statements used to replicate the statistics on the objects in a test database, as well as the update database configuration and update database manager configuration parameters and the db2set statements so that the registry variables and configuration parameter settings on the test database match those of the production database.

Windows/Linux/UNIX Examples

Command Location: \sqllib\bin (Windows)

Command Location: ~/sqllib/bin (UNIX)

Authorization: SELECT privilege on the system catalogs.

Example 1:

Generate the DDL statements for objects created by user "user1" in database SAMPLE. The db2look output is sent to file db2look.sql:

```
db2look -d sample -u user1 -e -o db2look.sql
```

Example 2:

Generate the DDL statements for objects that have schema name "ianhe", created by user "user1" and in database SAMPLE. The db2look output is sent to file db2look.sql:

```
db2look -d sample -u user1 -z ianhe -e -o db2look.sql
```

Example 3:

Generate the UPDATE statements to replicate the statistics for the tables and indexes created by user "user1" in database SAMPLE. The output is sent to file db2look.sql:

```
db2look -d sample -u user1 -m -o db2look.sql
```

Example 4:

Generate both the DDL statements for the objects created by user "user1" and the UPDATE statements to replicate the statistics on the tables and indexes created by the same user. The db2look output is sent to file db2look.sql:

```
db2look -d sample -u user1 -e -m -o db2look.sql
```

Example 5:

Generate the DDL statements for objects created by all users in the database SAMPLE. The db2look output is sent to file db2look.sql:

```
db2look -d sample -a -e -o db2look.sql
```

Example 6:

Generate the DDL statements for all user defined database partition groups, buffer pools and table spaces. The db2look output is sent to file db2look.sql:

```
db2look -d sample -l -o db2look.sql
```

Example 7:

Generate the UPDATE statements for the database and database manager configuration parameters, as well as the db2set statements for the registry variables in database SAMPLE. The db2look output is sent to file db2look.sql:

```
db2look -d sample -f -o db2look.sql
```

Example 8:

Generate the DDL for all objects in database SAMPLE, the UPDATE statements to replicate the statistics on all tables and indexes in database SAMPLE, the GRANT authorization statements, the UPDATE statements for the database and database manager configuration parameters, the db2set statements for the registry variables, and the DDL for all user defined database partition groups, buffer pools and table spaces in database SAMPLE. The output is sent to file db2look.sql.

```
db2look -d sample -a -e -m -l -x -f -o db2look.sql
```

Example 9:

Generate all authorization DDL for all objects in database SAMPLE, including the objects created by original creator. (In this case, the authorizations were granted by SYSIBM at object creation time.) The db2look output is sent to file db2look.sql:

```
db2look -d sample -xd -o db2look.sql
```

Example 10:

Generate the DDL statements for objects created by all users in the database SAMPLE. The db2look output is sent to file db2look.sql:

```
db2look -d sample -a -e -td % -o db2look.sql  
db2 -td% -f db2look.sql
```

Example 11:

Generate the DDL statements for objects in database SAMPLE, excluding the CREATE VIEW statements. The db2look output is sent to file db2look.sql:

```
db2look -d sample -e -noview -o db2look.sql
```

Example 12:

Generate the DDL statements for objects in database SAMPLE related to specified tables. The db2look output is sent to file db2look.sql:

```
db2look -d SAMPLE -e -t tabl \"My Table2\" -o db2look.sql
```

Example 13:

Generate the DDL statements for all objects (federated and nonfederated) in the federated database FEDDEPART. For federated DDL statements, only those that apply to the specified wrapper, FEDWRAP, are generated. The db2look output is sent to standard output:

```
db2look -d feddepart -e -wrapper fedwrap
```

Example 14:

Generate a script file that includes only non-federated DDLs. The following system command can be run against a federated database (FEDDEPART) and yet only produce output like that found when run against a database which is not federated. The db2look output is sent to a file out.sql:

```
db2look -d feddepart -e -nofed -o out
```

Notes

On Windows systems, **db2look** must be run from a DB2 Command Window.

db2move

Description: Database Movement Tool. This tool facilitates the movement of large numbers of tables between DB2 databases located on workstations. The tool queries the system catalog tables for a particular database and compiles a list of all user tables. It then exports these tables

in PC/IXF format. The PC/IXF files can be imported or loaded to another local DB2 database on the same system, or can be transferred to another workstation platform and imported or loaded to a DB2 database on that platform.

Windows/Linux/UNIX Examples

Command Location: \sqlib\bin (Windows)

Command Location: ~/sqlib/bin (UNIX)

Authorization: This tool calls the DB2 export, import, and load APIs, depending on the action requested by the user. Therefore, the requesting user ID must have the correct authorization required by those APIs, or the request will fail.

Example 1:

Export all tables in the database SAMPLE with default values.

```
db2move sample export
```

Example 2:

Export all tables created by "user1" or user IDs like "us%r2", and with the name "tablename1" or table names like "%tablename2".

```
db2move sample export -tc user1,us*r2 -tn tablename1,*tablename2
```

Example 3: Windows only.

Import all tables in the database SAMPLE; LOB paths "D:\LOBPATH1" and "C:\LOBPATH2" are to be searched for LOB files.

```
db2move sample import -l D:\LOBPATH1,C:\LOBPATH2
```

Example 4: UNIX only.

Load all tables in database SAMPLE; both the /home/db2inst1/lobpath subdirectory and "/tmp" subdirectory are to be searched for LOB files.

```
db2move sample load -l /home/db2inst1/lobpath,/tmp
```

Example 5:

Import all tables in database SAMPLE in replace mode; use user ID "user1" and password "xyz".

```
db2move sample import -io replace -u user1 -p xyz
```

Notes

- Tables with structured type columns are not moved when this tool is used.
- It is still necessary to move all other objects associated with the tables, such as: aliases, views, triggers, user-defined functions, and so on.
- The LOAD action must be run locally on the machine where the database and the data file reside.
- 4. 'db2move import' performance may be improved by altering default buffer pool, IBMDEFAULTBP; and by updating the configuration parameters *sortheap*, *util_heap_sz*, *logfilsz*, and *logprimary*.

db2mqIsn

Description: MQ Listener Command. Invokes the asynchronous MQListener to monitor a set of WebSphere MQ message queues, passing messages that arrive on them to configured DB2 stored procedures.

Windows/Linux/UNIX Examples

Command Location: \sqlib\bin (Windows)

Command Location: /usr/opt/db2_08_01/instance (AIX)

Command Location: /opt/IBM/db2/V8.1/instance (Linux/UNIX)

Authorization: EXECUTE privilege on package mqlConfi –or-

EXECUTE privilege on specified stored procedures –and- EXECUTE privilege on the package mqlRun.

Example 1:

List the configuration information for database SAMPLE using the configuration tasks specified in “nightlies”

```
db2mqlsn show -configDB sample -config nightlies
```

Example 2:

Add to the configuration task “nightlies”; input queue “app3”, stored procedure “user1.proc3”, stored procedure database PROCDB.

```
db2mqlsn add -configDB sample -config nightlies -inputQueue app3 -  
procSchema user1 -procName proc3 -dbName procdB -dbUser user1 -dbPwd  
xyz
```

Example 3:

Enable queue manager to listen for the tasks defined in “nightlies”

```
db2mqlsn run -configDB sample -config nightlies
```

Example 4:

Remove the input queue “app3” from configuration task “nightlies” created in example 2.

```
db2mqlsn remove -configDB sample -config nightlies -inputQueue app3  
-procSchema user1 -procName proc3 -dbName procdB -dbUser user1 -dbPwd  
xyz
```

Notes

For more information about controlling access to WebSphere MQ objects, refer to the *WebSphere MQ System Administration Guide*

db2mcs

Description: Set up Windows Failover Utility Command. Creates the infrastructure for DB2 failover support on Windows using Microsoft Cluster Server (MSCS). This utility can be used to enable failover in both single-partition and partitioned database environments.

Windows Examples

Command Location:

Authorization: The user must be logged on to a domain user account that belongs to the Administrators group of each machine in the MSCS cluster.

Example 1: Copy the sample DB2MSCS.EE file in \SQLLIB\CFG directory to DB2MSCS.CFG. Edit the DB2MSCS.CFG file to fit your environment. Assume the instance name is DB2. Test the configuration by enabling failover to use the DB2MSCS.CFG file.

```
db2mcs -f:~\SQLLIB\CFG\DB2MSCS.CFG
```

Example 2: Back out of the db2mcs operation and revert the instance DB2 back to non-MSCS.

```
db2mcs -u:db2
```

Linux/UNIX Examples

- If the -f: parameter is not specified, the utility will read the DB2MSCS.CFG file that is in the current directory.
- Two example configuration files can be found in the \SQLLIB\CFG directory; DB2MSCS.EE and DB2MSCS.EEE.
- 3. For more information on the Microsoft Cluster Server (MSCS) functionality and setup, refer to the Admin Guide.

db2mtrk

Description: Provide complete report of memory status, for instances, databases and agents. This command outputs the following memory pool allocation information:

- Current size
- Maximum size (hard limit)
- Largest size (high water mark)

- Type (identifier indicating function for which memory will be used)
- Agent who allocated pool (only if the pool is private)

Windows/Linux/UNIX examples

Authorization: sysadm/sysctrl/sysmaint

Example 1:

Display memory usage in details (-v option) on both instance (-i option) and database (-d option) levels.

```
db2mtrk -i -d -v
```

Sample output:

Tracking Memory on: 2004/03/05 at 13:35:49

Memory for instance

Database Monitor Heap is of size 180224 bytes

Other Memory is of size 1081344 bytes

Total: 1261568 bytes

Memory for database: SAMPLE

Backup/Restore/Util Heap is of size 16384 bytes

Package Cache is of size 163840 bytes

Catalog Cache Heap is of size 81920 bytes

Buffer Pool Heap is of size 4276224 bytes

Buffer Pool Heap is of size 606208 bytes

Buffer Pool Heap is of size 344064 bytes

Buffer Pool Heap is of size 212992 bytes

Buffer Pool Heap is of size 147456 bytes

Lock Manager Heap is of size 491520 bytes

Database Heap is of size 3260416 bytes

Other Memory is of size 0 bytes

Total: 9601024 bytes

Example 2:

Display high watermarks of each pool (-w option) in details. (-v option). Repeat every 30 seconds. (-r option)

```
db2mtrk -i -d -w -r 30 -v
```

Sample output:

Tracking Memory on: 2004/03/05 at 13:41:26

Memory for instance

Database Monitor Heap has watermark of 180224 bytes

Other Memory has watermark of 1097728 bytes

Total: 1277952 bytes

Memory for database: SAMPLE

Backup/Restore/Util Heap has watermark of 16384 bytes

Package Cache has watermark of 163840 bytes

Catalog Cache Heap has watermark of 81920 bytes

Buffer Pool Heap has watermark of 4276224 bytes

Buffer Pool Heap has watermark of 606208 bytes

Buffer Pool Heap has watermark of 344064 bytes

Buffer Pool Heap has watermark of 212992 bytes

Buffer Pool Heap has watermark of 147456 bytes
Lock Manager Heap has watermark of 491520 bytes
Database Heap has watermark of 3260416 bytes
Other Memory has watermark of 0 bytes
Total: 9601024 bytes

Example 3:

Display the maximum values for each pool. (-m option)
`db2mtrk -i -d -m -v`

Sample output:

Tracking Memory on: 2004/03/05 at 13:51:49

Memory for instance

Database Monitor Heap has max size of 540672 bytes
Other Memory has max size of 9076736 bytes
Total: 9617408 bytes

Memory for database: SAMPLE

Backup/Restore/Util Heap has max size of 20660224 bytes
Package Cache has max size of 4294950912 bytes
Catalog Cache Heap has max size of 4294950912 bytes
Buffer Pool Heap has max size of 4294950912 bytes
Buffer Pool Heap has max size of 4294950912 bytes
Buffer Pool Heap has max size of 4294950912 bytes
Buffer Pool Heap has max size of 4294950912 bytes
Buffer Pool Heap has max size of 4294950912 bytes
Lock Manager Heap has max size of 638976 bytes
Database Heap has max size of 8962048 bytes
Other Memory has max size of 12517376 bytes
Total: 30107435008 bytes

Example 4:

Display memory usage by agents. (-p option)
`db2mtrk -i -d -p -v`

Sample output:

Tracking Memory on: 2004/03/05 at 14:01:09

Memory for instance

Database Monitor Heap is of size 180224 bytes
Other Memory is of size 1081344 bytes
Total: 1261568 bytes

Memory for database: SAMPLE

Backup/Restore/Util Heap is of size 16384 bytes
Package Cache is of size 163840 bytes
Catalog Cache Heap is of size 81920 bytes
Buffer Pool Heap is of size 4276224 bytes
Buffer Pool Heap is of size 606208 bytes
Buffer Pool Heap is of size 344064 bytes
Buffer Pool Heap is of size 212992 bytes

Buffer Pool Heap is of size 147456 bytes
Lock Manager Heap is of size 491520 bytes
Database Heap is of size 3260416 bytes
Other Memory is of size 0 bytes
Total: 9601024 bytes

Memory for agent 4129012

Application Heap is of size 65536 bytes
Application Control Heap is of size 16384 bytes
Total: 81920 bytes

Memory for agent 35610770

Application Heap is of size 131072 bytes
Application Control Heap is of size 16384 bytes
Total: 147456 bytes

Notes

- On Windows platforms, use the same commands as in above examples. However, disregard the `-d` option.
- There has to be at least one connection to the database to show some data. Otherwise the command will return "No active agents".
- The "Other Memory" reported is the memory associated with the overhead of operating the database management system.
- In some cases (such as the package cache) the maximum size displayed will be larger than the value assigned to the configuration parameter. In such cases, the value assigned to the configuration parameter is used as a 'soft limit', and the pool's actual memory usage may grow beyond the configured size.

db2nchg (Windows Only)

Description: Modifies database partition server configuration. This includes moving the database partition from one machine to another; changing the TCP/IP host name of the machine; and selecting a different logical port number or a different network name for the database partition. This command can only be used if the database partition is stopped.

Windows examples

Authorization: local administrator

Example 1: Instance INST1 has two partitions, 0 and 1. Its db2nodes.cfg looks like this:

```
0 sylviaq SYLVIAQ 0
1 sylviaq SYLVIAQ 1
```

Now we want to change partition 1 to use port 2 instead of port 1.

```
db2nchg /i:inst1 /n:1 /p:2
```

Upon completion of this command, db2nodes.cfg will be changed to:

```
0 sylviaq SYLVIAQ 0
1 sylviaq SYLVIAQ 2
```

db2ncrt (Windows Only)

Description: Adds a database partition to an existing instance. The instance must be created as a MPP instance to be able to use this command. Use this command only when there is no database exists on this instance. Otherwise, db2start with ADD PARTITIONNUM command must be used to add a new partition.

Windows examples

Authorization: local Windows administrator

Example 1:

Instance INST1 has the following properties:

- It is created as a MPP instance. (See example in **db2icrt** command.)
- It resides on machine 'sylviaq'.
- It has only one partition. Its db2nodes.cfg contains the following line: '0 sylviaq SYLVIAQ 0'
- No database has been created under INST1.

Now we want to add one more partition to INST1. The new partition has the following properties:

- It resides on the same machine. (Use /m option to specify the machine name where the new partition resides.)
- Its instance-owning machine is 'sylviaq'. This is the machine where partition 0 resides. (/o option)
- The new partition will have partition number 1. (/n option)
- The port the partition will use is 1. (/p option)
- The user is a domain user. The domain name is 'TOROLAB'. The user id and password are 'sylvia' and 'abc', respectively. (/u option). If user is not a domain user, then no domain name needs to be specified.

```
db2nrcrt /i:inst1 /m:sylviaq /o:sylviaq /n:1 /p:1 /u:TOROLAB\sylvia,abc
```

Following the completion of this command, upon db2start, db2nodes.cfg for INST1 will be updated to contain 2 lines:

```
0 sylviaq SYLVIAQ 0  
1 sylviaq SYLVIAQ 1
```

Notes

Please do NOT try editing the db2nodes.cfg manually. Doing so may cause inconsistencies in the partitioned database system.

db2ndrop (Windows Only)

Description: Drops a database partition from an existing instance. This command can only be used when there is no databases exist on the instance. Otherwise, db2stop with DROP PARTITIONNUM command should be used.

Windows examples

Authorization: local Windows administrator

Example 1:

Instance INST1 has two database partitions. Its db2nodes.cfg contains 2 lines:

```
0 sylviaq SYLVIAQ 0  
1 sylviaq SYLVIAQ 1
```

Now we want to drop partition number 1 from INST1.

```
db2ndrop /i:inst1 /n:1
```

Notes

- Please do NOT try editing the db2nodes.cfg manually. Doing so may cause inconsistencies in the partitioned database system.
- To drop a database partition server that is assigned to the logical port 0 from a machine that is running multiple logical database partition servers, all other database partition servers assigned to the other logical ports must be dropped first. Each database partition server must have a database partition server assigned to logical port 0.
- If the instance-owning database partition (partition 0) is dropped from the instance, the instance becomes unusable. To drop the instance, use the **db2idrop** command.

db2osconf (HP-UX 64 bit and Solaris Only)

Description: Makes recommendations for kernel parameter values based on the size of a system. The recommended values are high enough for a given system that they can accommodate most reasonable workloads.

Examples

Command location: /opt/IBM/db2/V8.1/bin

Authorization: None on HP-UX, root or member of sys group on Solaris.

Example 1:

To get the current kernel configuration:

```
db2osconf -l
```

Example 2:

To get recommendations on kernel configuration based on a RAM of 8GB.

```
db2osconf -m 8
```

Example 3:

To get recommendations on kernel configuration based on 3 CPUs.

```
db2osconf -n 3
```

db2perfc (Windows Only)

Description: Resets the performance values for one or more databases. It is used with the Performance Monitor on Windows operating systems.

Windows examples

Authorization: local Windows administrator

Example 1:

Reset performance values for all active databases.

```
db2perfc
```

Example 2:

Reset performance values for SAMPLE and TESTDB.

```
db2perfc sample testdb
```

Example 3:

Reset performance values for all DCS databases.

```
db2perfc -d
```

Notes

Please also see **db2perfr** and **db2perfi**.

db2perfi (Windows Only)

Description: Adds the DB2 Performance Counters to the Windows operating system. This must be done to make DB2 and DB2 Connect performance information accessible to the Windows Performance Monitor.

Windows examples

Authorization: local Windows administrator.

Example 1: Register DB2 performance counters.

```
db2perfi -i
```

Example 2: Deregister DB2 performance counters.

```
db2perfi -u
```

Notes

Please also see **db2perfr**.

db2perfr (Windows Only)

Description: Registers a DB2 user id to Windows Performance Monitor. Windows Performance Monitor will use this id to gather DB2 information. Make sure the user id has sysadm authority to avoid any problem.

Windows examples

Authorization: local Windows administrator

Example 1:

Register user id 'sylvia' with the Windows Performance Monitor. 'sylvia' should have sysadm authority in the DB2 instance. 'abc' is the password.

```
db2perfr -r sylvia abc
```

Example 2:

Deregister user id 'sylvia'

```
db2perfr -u sylvia
```

Notes

Please also see **db2perfi**.

db2rbind

Description: Rebinds packages in a database.

Windows/Linux/UNIX examples

Authorization: sysadm

Example 1:

Rebind all packages in the TEST database (option all). The file that is going to be used to log errors is rbind.log. (option -l)

```
db2rbind test -l rbind.log all
```

Example 2:

Rebind all functions and types in the SQL path. Conservative binding semantics are not used.

```
db2rbind test -l rbind.log -r any
```

Example 3:

Rebind only functions and types in the SQL path that were defined before the last explicit bind time stamp. Conservative binding semantics are used. This is the default. This option is not supported for an inoperative package.

```
db2rbind test -l rbind.log -r conservative
```

db2_recon_aid

Description: Provides a mechanism for checking and running RECONCILE on tables of a database that are potentially inconsistent with the DATALINK file data on the file server. It must be run on a DB2 server containing tables with DATALINK columns to be reconciled.

Windows/Linux/UNIX examples

Command Location: *INSTHOME*/sqlib/adm (UNIX)

Command Location: x:\sqlib\bin directory, where x: is the drive where DB2 Data Links Manager installed. (Window)

Example 1: List the tables that might need reconciliation in the TESTDB database.

```
db2_recon_aid -db TESTDB -check
```

Example 2:

Perform the reconcile operation on TESTDB database. Place the report files under /home/db2inst1.

```
db2_recon_aid -db TESTDB -reportdir /home/db2inst1
```

Example 3:

Perform the reconcile operation on TESTDB database.

- Place report files under /home/db2inst1.
- The operation is only going to be performed on the dlmserver.services.com data links server.
- The operation is only going to be performed on those Data Links File System (DLFS) whose prefixes are 'dlfsdir1/db2inst1' and 'dlfsdir2/db2inst1'.

```
db2_recon_aid -db TESTDB -reportdir /home/db2inst1 -selective -server  
dlmserver.services.com -prefixes/dlfsdir1/db2inst1/:/dlfsdir2/db2inst1/
```

Notes

The `-server` and `-prefixes` options must be specified when `-selective` option is present.

db2relocatedb

Description: Renames a database, or relocates a database or part of a database (for example, the container and the log directory) as specified in the configuration file provided by the user. This tool makes the necessary changes to the DB2 instance and database support files. The following properties of a database can be altered using `db2relocatedb`:

- Database name
- Instance it belongs
- Database directory
- Log directory if it does not reside in the database directory
- Location of tablespace containers if they do not reside in the database directory

The command syntax is '`db2relocatedb -f <configuration_file_name>`'.

Windows/Linux/UNIX examples

Authorization: None.

Example 1:

SAMPLE database belongs to instance `sylviaq`. It is created under `/data`. To rename the SAMPLE database to SAMPLE1, edit a configuration file as follows and run the `db2relocatedb` command using this file:

```
DB_NAME=SAMPLE, SAMPLE1  
DB_PATH=/data  
INSTANCE=sylviaq
```

Example 2:

Move the database SAMPLE from the instance `sylviaq` on path `/data` to instance `db2inst1`, on the same path. The following are what needed to be done:

- Move all the files in `/data/sylviaq` to `/data/db2inst`
- Edit a configuration file as follows and run `db2relocatedb` using this file:

```
DB_NAME=SAMPLE  
DB_PATH=/data  
INSTANCE=sylviaq,db2inst1
```

Example 3:

SAMPLE database belongs to instance `sylviaq`. It is created under `/data`. It has a SMS tablespace container `/home/sylviaq/ts1`. Now this directory must be moved to `/home/sylviaq/ts/ts1`. We need to do the following to complete this move:

- Copy all the files in `/home/sylviaq/ts1` to `/home/sylviaq/ts/ts1`.
- Edit a configuration file and run `db2relocatedb` using this file:

```
DB_NAME=SAMPLE  
DB_PATH=/data  
INSTANCE=sylviaq  
CONT_PATH=/home/sylviaq/ts1,/home/sylviaq/ts/ts1
```

Example 4:

The SAMPLE database belongs to instance sylviaq. It is created under /data. It has the following tablespace containers:

- /data/sylviaq/NODE0000/SQL00001/SQLT0000.0
- /home/sylviaq/ts1
- /dev/sylviaq/ts2
- /dev/sylviaq/ts3
- /dev/sylviaq/ts4

Now we want to move the SAMPLE database to a different system. The new instance will be db2inst1. The new database path will be /db2. The new container paths will be:

- /db2/db2inst1/NODE0000/SQL00001/SQLT0000.0
- /home/db2inst1/ts1
- /dev/db2inst1/ts2
- /dev/db2inst1/ts3
- /dev/db2inst1/ts4

This is what we need to do to complete the move:

- Copy all the files that exist in the /data/sylviaq directory to /db2/db2inst directory. Here, the first container will be copied to /db2/db2inst1/NODE0000/SQL00001/SQLT0000.0
- Now we need to copy the rest of the containers to their new locations. Because this has not been done the previous step.
- Edit a configuration file as follows and run db2relocatedb using this file. (Note we do not need to add the first container in the configuration file because it is located in the database directory.)

```
DB_NAME=SAMPLE
DB_PATH=/data, /db2
INSTANCE=sylviaq, db2inst1
CONT_PATH=/home/sylviaq/ts1, /home/db2inst1/ts1
CONT_PATH=/dev/sylviaq/*, /dev/db2inst1/*
```

Tip: An alternative to complete this move is to use the RESTORE command with the REDIRECT option.

Example 5:

The database TESTDB is created on two partition servers, 1 and 2. The instance is servinst and the database path is /home/servinst on both database partition servers. The name of the database is being changed to SERVDB and the database path is being changed to /databases on both database partition servers. In addition, the log directory is being changed on database partition server 2 from /testdb_logdir to /servdb_logdir.

Since changes are being made to both database partitions, a configuration file must be created for each database partition and db2relocatedb must be run on each database partition server with the corresponding configuration file.

On database partition server 1, the following configuration file will be used:

```
DB_NAME=TESTDB, SERVDB
DB_PATH=/home/servinst, /databases
INSTANCE=servinst
NODE_NUM=1
```

On database partition server 2, the following configuration file will be used:

```
DB_NAME=TESTDB,SERVDB
DB_PATH=/home/servinst,/databases
INSTANCE=servinst
NODE_NUM=2
LOG_DIR=/testdb_logdir,/servdb_logdir
```

db2rspgn

Description: Generates a response file for use of silent installation. It tool is available on Windows only.

Windows Examples

Authorization: None.

Example 1:

Generate a response file for all instances on the system except the Data Links File Manager instance. The file will be created in D:\temp directory.

```
db2rspgn -d D:\temp -nodlfm
```

Example 2:

Generate a response file for the DB2 instance only. The file will be created in D:\temp directory.

```
db2rspgn -d D:\temp -i db2
```

db2sampl

Description: Creates a database called SAMPLE.

Windows/Linux/UNIX examples

Authorization: sysadm/sysctrl

Example 1:

On UNIX, the following command creates the SAMPLE database in the instance home directory. (e.g. /home/db2inst1)

On Windows, the following command creates the SAMPLE database in the root directory where DB2 is installed. (e.g. D:\)

```
db2sampl
```

Example 2:

On UNIX, create the SAMPLE database in /data/db2 directory.

```
db2sampl /data/db2
```

Example 3:

On Windows, create the SAMPLE database in D:\databases

```
db2sampl D:\databases
```

Notes

If SAMPLE already exists, **db2sampl** creates the tables for the user ID issuing the command, and grants the appropriate privileges.

db2set

Description: Displays, sets, or removes DB2 profile variables.

Windows/Linux/UNIX examples

Authorization: sysadm

Example 1:

Display all the registry variables that are currently set.

```
db2set -all
```

Example 2:

Display all the registry variables that are supported.

```
db2set -lr
```

Example 3:

Set DB2COMM to TCPIP at instance level (-i option). If -i is not specified, then -g option will be used by default, which means the variable will be applied on all instances on this system.

```
db2set -i DB2COMM=TCPIP
```

Example 3:

Unset the DB2COMM variable.

```
db2set DB2COMM=
```

Example 4:

Reset the variables for instance db2inst1. (Please use this command with caution. It will erase all previous settings.)

```
db2set -r db2inst1
```

Notes

Instance needs to be restarted to make changes take effect.

db2setup (UNIX)/setup (Windows)

Description: Installs DB2 products. It is located on the DB2 installation media (The installation CD).

Windows/Linux/UNIX examples

Authorization: Root authority (On UNIX)

Administrator authority (On Windows)

Example 1:

Install DB2. Generate a file called db2setup.trace to capture trace information, another file called db2setup.log to capture log information. (This is a binary file, usually requested by DB2 service when investigating installation problems.)

```
db2setup -t db2/tmp/setup.trace -l /tmp/db2setup.log (UNIX)
Z:\db2udbesrvedtv81\db2udbesrvedtv81\011_ESE_WIN_32_NLV>setup -l
d:\db2setup.log -t d:\db2setup.trace (Windows)
```

Example 2:

Install DB2 in silent mode using a response file.

```
db2setup -t /tmp/db2setup.trace -r /tmp/db2ese.rsp (UNIX)
Z:\db2udbesrvedtv81\db2udbesrvedtv81\011_ESE_WIN_32_NLV>setup -t
d:\db2setup.trace -u d:\db2ese.rsp (Windows)
```

Notes

To generate a response file, go to /db2/<platform>/samples directory on the DB2 installation CD. The following sample response files are available:

- DB2 Application Development Client - db2adcl.rsp
- DB2 Administration Client - db2admcl.rsp
- DB2 Connect Enterprise Edition - db2conee.rsp
- DB2 Connect Personal Edition - db2conpe.rsp

- DB2 Cube Views - db2cube.rsp
- DB2 Data Links Manager - db2dlm.rsp
- DB2 Enterprise Server Edition - db2ese.rsp
- DB2 Personal Edition - db2pe.rsp
- DB2 Query Patroller - db2qp.rsp
- DB2 Run-Time Client - db2rtcl.rsp
- DB2 Spatial Extender Server - db2gse.rsp
- DB2 UDB Express Edition - db2exp.rsp
- DB2 Warehouse Manager - db2wm.rsp
- DB2 Warehouse Manager Connectors - db2wmc.rsp
- DB2 Workgroup Server Edition - db2wse.rsp
- Information Integrator Relational Wrappers - 4db2relc.rsp (Windows) or db2iirw.rsp (UNIX)
- Information Integrator Non-Relational Wrappers - 4db2lsdc.rsp (Windows) or and db2iinw.rsp (UNIX)

You need to edit the sample response file so it is customized to your environment. Then use this response file to start silent installation.

On Windows, there is a utility which generates a response file. Please refer to db2rspgn.

db2sql92

Description: Reads SQL statements from either a flat file or standard input, dynamically describes and prepares the statements, and returns an answer set. Supports concurrent connections to multiple databases.

Windows/Linux/UNIX examples

Authorization: sysadm

Example 1:

Run db2sql92 as follows:

- Save the SQL statements in a file called sql.txt. (-f option)
- These statements run against the SAMPLE database. (-d option)
- Measure the elapsed time (-i option)
- Save the output to a file called out.txt (-r option)
- The user id used is 'sylviaq', password is 'abc'. (-a option)

```
db2sql92 -d sample -i complete -a sylviaq/abc -f sql.txt -r out.txt
```

Example 2:

Run db2sql92 as follows:

- Use the interactive mode of db2sql92 to input SQL statement (no -f option)
- Use the SAMPLE database (-d option)
- Get a summary table in the end (-s option)

```
db2sql92 -d sample -a sylviaq/abc -s on
```

Output:

SQL authorization ID = SYLVIAQ

Local database alias = SAMPLE

Running in Embedded Dynamic mode.

```
-----
DB2SQL92> select * from org;
```

```
Statement # 1 :
```

```
select * from org
```

```
DEPTNUMB DEPTNAME      MANAGER DIVISION  LOCATION
```

```

-----
10 Head Office      160 Corporate New York
15 New England      50 Eastern   Boston

Number of rows retrieved is:    8
Number of rows sent to output is: 8

Elapsed Time is:    0.075    seconds
-----
DB2SQL92> quite

Summary of Results
=====
Statement #    Elapsed Time (s)
1              0.075

TOTAL ELAPSED TIME: 0.075    seconds
Notes
-----


- All SQL statements must be terminated by a semicolon!
- db2sql92 issues its own connect and connect reset.
- All statements executed with isolation level RR.
- The maximum SQL statement size is 65535 characters.
- 5. PAUSE & SLEEP are timed when in short timing mode.

```

```

db2sqljbind
Description: Binds a previously customized SQLJ profile to a database. By default, four
packages are created, one for each isolation level. If the -singlepkgname option is used when
customizing, only a single package is created and the ISOLATION option must be used. This
utility should be run after the SQLJ application has been customized.
Windows/Linux/UNIX examples
Authorization: One of the following:



- sysadm or dbadm authority
- BINDADD privilege if a package does not exist, and one of:
  - IMPLICIT_SCHEMA authority on the database if the schema name of the
package does not exist
  - CREATEIN privilege on the schema if the schema name of the package exists
- ALTERIN privilege on the schema if the package exists
- BIND privilege on the package if it exists.


The user also needs all privileges required to compile any static SQL statements in the
application. Privileges granted to groups are not used for authorization checking of static
statements. If the user has sysadm authority, but not explicit privileges to complete the bind, the
database manager grants explicit dbadm authority automatically.

Example 1:
db2sqljbind -user db2v8 -password abc -url
jdbc:db2://server:50000/sample -bindoptions "EXPLAIN YES"
salary_SJProfile0.ser

```

```

db2sqljcustomize
Description: Processes an SQLJ profile containing embedded SQL statements. By default, four
DB2 packages are created in the database: one for each isolation level. This utility augments the
profile with DB2-specific information for use at run time, and should be run after the SQLJ
application has been translated, but before the application is run.
Windows/Linux/UNIX examples

```

Authorization: Same as db2sqljbind.

Example 1:

```
db2sqljcustomize -user db2v8 -password abc
                -url jdbc:db2:/server:50000/sample -collection user1
                -bindoptions "EXPLAIN YES" salary_SJProfile0.ser
```

db2sqljprint

Description: Prints the contents of a DB2 customized version of a profile in plain [text](#).

Windows/Linux/UNIX examples

Authorization: None.

Example 1: salary_SJProfile0.ser is an SQLJ profile file. When an SQLJ file is translated into a Java file, information about the SQL operations it contains is stored in SQLJ-generated resource files called profiles. Profiles are identified by the suffix _SJProfileN (where N is an integer) following the name of the original input file. They have a .ser extension. Profile names can be specified with or without the .ser extension.

```
db2sqljprint salary_SJProfile0.ser
```

db2start/start database manager/start dbm

Description: Starts database manager.

Windows/Linux/UNIX examples

Authorization: sysadm/sysctrl/sysmaint

The ADD DBPARTITIONNUM start option requires either *sysadm* or *sysctrl* authority.

Example 1:

Start the instance. In multi-node environment, this will start all nodes in db2nodes.cfg.

```
db2start
```

Example 2:

Starts the instance in quiesced mode for administration purposes. This is equivalent to the QUIESCE INSTANCE command except in this case the instance is not already "up. The user id that is used to execute the command is 'sylvia'.

```
db2start admin mode user sylvia
```

The following examples apply to multi-node environments only.

Example 3:

Restart database partition 1 after a failure (using the RESTART option). Partition 1 will reconnect to other operational partitions upon the successful completion of this command.

```
db2start dbpartitionnum 1 restart
```

Example 4:

The machine which partition 1 resides on has lost power. We want to restart partition 1 on a different machine with the following properties:

- The hostname of the machine is 'panipuri'.
- The logical port number to use is 0
- The netname is 'panipuri'.

```
db2start nodenum 1 restart hostname panipuri port 0 netname panipuri
```

Upon successful completion of this command, the db2nodes.cfg will be updated to reflect this change.

Example 5:

Instance DB2INST1 resides on machine 'panipuri'. It currently has 3 partitions. 0, 1 and 2. Now we want to add a 4th partition to INST1. When adding a new partition to an existing instance, a new container has to be created and added to the existing temporary tablespace, to accommodate the newly added partition.

The 4th partition has the following properties:

- It will have a partition number 3 and logical port number 3
- It will reside on the same machine, 'panipuri'.
- It will use the default temporary tablespace container assigned by DB2, `/database_path/NODE0003/SQL00001/SQLT0001.0`

```
db2start dbpartitionnum 3 add dbpartitionnum hostname panipuri port 3
```

Example 6:

Same as Example 5. Except this time partition 3 will use tablespace container definition of partition 1 to create its own container (using the LIKE option). For example, if the container for partition 1 is `/data/tmp1`, then partition 3 will also use `/data/tmp` for its container. Therefore, for this to work, partition 1 and 3 must reside on different machines.

Otherwise, we will be attempting to use the same container for two partitions, which is not allowed.

```
db2start dbpartitionnum 3 add dbpartitionnum hostname panipuri port 3  
like dbpartitionnum 1
```

Example 7:

Same as Example 5. Except this time there will be no tablespace containers created for partition 3 at db2start time. It will be created later using ALTER TABLESPACE statement.

```
db2start dbpartitionnum 3 add dbpartitionnum hostname panipuri port 3  
without tablespaces
```

An example of the ALTER TABLESPACE statement would be:

```
db2 "ALTER TABLESPACE temporary_tablespace_name ADD  
( '/home/sylviaq/sms1/cont3' ) ON DBPARTITIONNUM (3)"
```

Notes

- After adding a new partition to an instance, you need to redistribute data across all partitions using the **redistribute database partition group** command.
- For compatibility with versions earlier than Version 8:
 - The keywords LIKE NODE can be substituted for LIKE DBPARTITIONNUM.
 - The keyword ADDNODE can be substituted for ADD DBPARTITIONNUM.
 - The keyword NODENUM can be substituted for DBPARTITIONNUM.

db2stop/stop database manager/stop dbm

Description: Stops database manager.

Windows/Linux/UNIX examples

Authorization: sysadm/sysctrl/sysmaint

Example 1:

Stop the instance.
`db2stop`

Example 2:

Use FORCE APPLICATION ALL when stopping the instance.

```
db2stop force
```

Example 3:

Stops partition 2 only.

```
db2stop dbpartitionnum 2
```

Example 4:

Drop partition 3 from existing instance.

```
db2stop drop dbpartitionnum 3
```

Notes

Before dropping a partition from an instance, all data should have been moved to other partitions already using the **redistribute database partition group** command.

db2support

Description: Collects environment data about either a client or server machine and places the files containing system data into a compressed file archive.

Windows/Linux/UNIX examples

Authorization: For the most complete output, this utility should be invoked by the instance owner. Users with more limited privileges on the system can run this tool. However some of the data collection actions will result in reduced reporting and reduced output.

Example 1:

Get the following info on SAMPLE database:

- detailed hardware and operating system info (-s)
- all files in the DB2DIAGPATH directory excluding core(-g)
- all core files (-c)
- Output the info in the current working directory. The default file name is db2support.zip.

```
db2support . -d sample -s -g -c
```

Example 2:

Run the db2support tool on SAMPLE database in interactive mode. Save the output into /home/db2v8/temp directory. Collect only the active log files. (-l option)

```
db2support /home/db2v8/temp -d sample -l
```

db2sync

Description: Facilitates the initial configuration of a satellite as well as changes to the configuration. This command can also be used to start, stop and monitor the progress of a synchronization session and to upload a satellite's configuration information (for example, communications parameters) to its control server.

Windows/Linux/UNIX examples

Authorization: None

Example 1:

Displays a graphical user interface that allows an administrator to change either the application version or synchronization credentials for a satellite.

```
db2sync -t
```

Example 2:

Displays the application version currently set on the satellite.

```
db2sync -g
```

Example 3:

Sets the application version on the satellite.

```
db2sync -s
```

db2tbst
Description: Accepts a hexadecimal table space state value, and returns the state. The state value is part of the output from LIST TABLESPACES.
Windows/Linux/UNIX examples
Authorization: None.
Example 1: An error shows up in db2diag.log when performing an insert into a table. The message says that the insert was not possible because the tablespace state was '0x0004'. Use db2tbst to find out what '0x0004' means. db2tbst 0004
Output: State = Quiesced Exclusive

db2trc
Description: Trace facility.
Windows/Linux/UNIX examples
Authorization: To trace a DB2 instance on a UNIX-based system, one of the following: <ul style="list-style-type: none"> • <i>sysadm</i> • <i>sysctrl</i> • <i>sysmaint</i> <p>DASADM to trace the DB2 Administration Server on a UNIX-based system.</p> <p>On the Windows operating system, no authorization is required.</p> <p>Example 1: Trace a connect command. Set the facility to retain the last 128M of the trace buffer if trace wraps. Note the trace is turned off only after the data in trace buffer has been dumped into a file. Format the trace at last so they are readable. db2trc on -l 128M db2 connect to sample db2trc dump connect.dmp db2trc off db2trc flw connect.dmp connect.flw db2trc fmt connect.dmp connect.fmt</p> <p>Example 2: Trace PIDs 73000 and 54502. Retain the first 64M of trace buffer if trace wraps. db2trc on -i 64M -p 73000,54502</p> <p>Example 3: Trace PID 3425, thread ID 1200. Retain the last 4M of trace buffer. db2trc on -p 3425.1200 -l 4M</p>
Notes
<ul style="list-style-type: none"> • When using the -p option, a maximum of 5 processes can be specified. • On UNIX platforms other than AIX, you need to allocate the trace buffer before starting the instance. Otherwise, only a very small buffer is allocated to trace facility.

db2undgp
Description: Revokes the execute privilege on external stored procedures.

During the database migration, EXECUTE for all existing functions, methods, and External stored procedure is granted to PUBLIC. This will cause a security exposure for External Stored procedures that contain SQL data access. To prevent users from accessing SQL objects which the user might not have privilege for, use the db2undgp command.

Windows/Linux/UNIX examples

Authorization:

Example 1:

Revoke EXECUTE privileges on all functions, methods, stored procedures from PUBLIC, in SAMPLE databases.

```
db2undgp -d sample -r
```

db2untag

Description: If a tablespace/database is destroyed thru unnatural means, then the tag can be left behind preventing future DB2 use of the resource. db2untag removes the DB2 tag on a table space container.

Windows/Linux/UNIX examples

Authorization: Read/write access to the container for a table space that is owned by the ID that created the database.

Example 1:

A DMS tablespace TS1 was created using the following statement:

```
db2 "create tablespace ts1 managed by database using (FILE  
' /home/db2v8/ts1/cont1' 500)"
```

To remove the tag on container cont1:

```
db2untag /home/db2v8/ts1/cont1
```

Sample output:

db2untag: A service tool to remove the DB2 tag on a tablespace container.

The tag is used to prevent DB2 from reusing a container in more than one tablespace.

If a tablespace/database is destroyed thru unnatural means, then the tag can be left behind preventing future DB2 use of the resource.

WARNING: This tool should only be used by informed sysadmins.

Using file </home/db2v8/ts1/cont1>

```
version = 213  
db seed = 77886153  
poolID = 4  
contID = 0  
created = 1  
used = 1  
poolLSN = 0000 05DC 20A4  
Csum = 6AFA2C3F
```

```
Instance = db2v8  
Database = SAMPLE
```

If you are sure that this container is no longer needed by the identified DB2 database, answer 'Yes'.

Do you want to untag the container /home/db2v8/ts1/cont1?

--->

Example 2:

Remove the tag from the same container in example 1. But this time do it without prompting (-f option)

```
db2untag -f /home/db2v8/ts1/cont1
```

Notes

To remove tags on SMS tablespace containers, just delete the SQLTAG.NAM file from the container directory. No need to run db2untag on SMS tablespace containers.

db2updv8

Description: Updates the system catalogs in a database to support the current level by enabling the following built-in routines:

- HEALTH_DBM_INFO
- HEALTH_DBM_HI
- HEALTH_DBM_HI_HIS
- HEALTH_DB_INFO
- HEALTH_DB_HI
- HEALTH_DB_HI_HIS
- HEALTH_TBS_INFO
- HEALTH_TBS_HI
- HEALTH_TBS_HI_HIS
- HEALTH_CONT_INFO
- HEALTH_CONT_HI
- HEALTH_CONT_HI_HIS
- GET_WRAP_CFG_C
- LIST_FILES_C
- ODBC_DISCOVERY

Windows/Linux/UNIX examples

Authorization: sysadm

Example 1: Update the SAMPLE database.

```
db2updv8 -d sample
```

Example 2:

Display the help menu for this command.

```
db2updv8 -h
```