

## **DB2 Command Line Processor (CLP) 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 CLP command, because these examples be lacking in the DB2 Command Reference. However, this 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.

All the commands in this document can be issued either from the CLP interactive session, from the Operating System command prompt (Linux/UNIX) or the DB2 Command prompt (Windows). When issued from the OS command prompt or the DB2 Command prompt, the commands must be prefixed by 'db2'. For example, the command 'activate database sample' becomes 'db2 activate database sample'.

When issuing the DB2 CLP commands from the OS command prompt on Linux/UNIX systems, it may be necessary to put double quotes around the CLP command. For example, when issuing db2 describe output select \* from sales, you need to put double quotes around the describe command, e.g. db2 "describe output select \* from sales". These double quotes are necessary because the '\*' is a special OS command line character. The double quotes will allow DB2 to parse the statement properly.

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

| <b>ACTIVATE DATABASE</b>  |
|---|
| <p>Description: Activates the specified database and starts up all necessary database services, so that the database is available for connection and use by any application.<br/>           Authorization: sysadm/sysctrl/sysmaint</p>  |
| <b>Windows/Linux/UNIX Examples</b>  |
| <p>Example 1:<br/>           To activate database sample to eliminate any application time spent on database initialization:<br/>           activate database sample</p> <p>Example 2:<br/>           To activate database sample with user ID db2user and password abc123:<br/>           activate database sample user db2user using abc123</p> |
| <b>Notes</b>  |
| <ul style="list-style-type: none"> <li>• Databases initialized by ACTIVATE DATABASE can be shut down using the DEACTIVATE DATABASE command.</li> <li>• The application issuing the ACTIVATE DATABASE command cannot have an active database connection to any database.</li> </ul>  |

| <b>ADD CONTACT</b>  |
|---|
| <p>Description: Adds a contact to the contact list which can be either defined locally on the system or in a global list. Contacts are users to whom processes such as the Scheduler and Health Monitor send messages. The setting of the Database Administration Server (DAS) <i>contact_host</i> configuration parameter determines whether the list is local or global.<br/>           Authorization: None</p>   |
| <b>Windows/Linux/UNIX Examples</b>  |
| <p>Example 1:<br/>           Add Michael as the contact person to receive event notifications via email at <a href="mailto:dangm@ca.ibm.com">dangm@ca.ibm.com</a>:<br/>           add contact Michael type email address <a href="mailto:dangm@ca.ibm.com">dangm@ca.ibm.com</a></p> <p>Example 2:<br/>           Add Michael as the contact person to receive event notifications via page at <a href="mailto:Michael@page.com">Michael@page.com</a>. Include a description of the contact person.<br/>           add contact Michael type page address <a href="mailto:Michael@page.com">Michael@page.com</a> description 'DB2</p> |

Support for IBM'

Example 3:

Add Michael as the contact person to receive event notifications via page at [Michael@page.com](mailto:Michael@page.com). Specify the 250 character message-limit restriction on the paging service:

```
add contact Michael type page maximum page length 250 address  
Michael@page.com
```

#### Notes

The notification system uses the SMTP protocol to send the notification to the mail server specified by the DB2 Administration Server configuration parameter *smtp\_server*. It is the responsibility of the SMTP server to send the e-mail or call the pager.

### ADD CONTACTGROUP

Description: Adds a new contact group to the list of groups defined on the local system. A contact group is a list of users and groups to whom monitoring processes such as the Scheduler and Health Monitor can send messages.

Authorization: None

#### Windows/Linux/UNIX Examples

Example 1:

Create a new contact group "instance\_monitor" and add Michael and Sylvia as contact members:  
add contactgroup instance\_monitor contact Michael, contact Sylvia

Example 2:

Create a new contact group "instance\_monitor" and add Michael and Sylvia as contact members. Include a description of the group:

```
add contactgroup instance_monitor contact Michael, contact Sylvia  
description 'Instance down monitor group'
```

Example 3:

Create a new contact group "instance\_monitor" and add the group "dba\_group" to it. Include a description of the group.

```
add contactgroup instance_monitor group dba_group description 'Instance  
down monitor group'
```

#### Notes

- Local execution only: this command cannot be used with a remote connection.
- Contact group name must be unique among the set of groups on the system.

### ADD DBPARTITIONNUM

Description: Adds a new database partition server to the partitioned database environment. This command also creates a database partition for all databases on the new database partition server. The user can specify the source database partition server for the definitions of any system temporary table spaces to be created with the new database partition, or specify that no system temporary table spaces are to be created.

Authorization: sysadm/sysctrl

#### Windows/Linux/UNIX Examples

Example 1:

Using the following db2node.cfg:

|   |         |   |
|---|---------|---|
| 0 | ServerA | 0 |
| 1 | ServerA | 1 |
| 2 | ServerA | 2 |

Example 1:

To add a fourth partition (node 3), set "export db2node=3"

```
add dbpartitionnum
```

**Example 2:**

To add a fourth partition (node 3), you want the new partition's system temporary table space to be the same as node 2. Set "export db2node=3"

```
add dbpartitionnum like dbpartitionnum 2
```

**Example 3:**

To add a fourth partition (node 3), you do not want system temporary table space containers to be created. Set "export db2node=3"

```
add dbpartitionnum without tablespaces
```

**Notes**

- This command must be issued from the database partition server that is being added.
- For compatibility with versions earlier than Version 8, use the keyword NODE instead of DBPARTITIONNUM.

**ARCHIVE LOG**

Description: Closes and truncates the active log file for a recoverable database. If user exit is enabled, an archive request is issued.

Authorization: sysadm/sysctrl/sysmaint/dbadm

**Windows/Linux/UNIX Examples**

Using the following db2node.cfg:

|   |         |   |
|---|---------|---|
| 0 | ServerA | 0 |
| 1 | ServerA | 1 |
| 2 | ServerA | 2 |

**Example 1:**

You want to copy all log files for a sample database on all partitions to a development sample database at a particular timestamp. Truncate the logs at a particular time with:

```
archive log for database sample on all dbpartitionnums
```

**Example 2:**

The following command is the same as example 1. Archive logs for database on all dbpartitionnums is the default.

```
archive log for database sample
```

**Example 3:**

To archive all log files for sample database on all partitions except partition two:

```
archive log for db sample on all dbpartitionnums except dbpartitionnum  
(2)
```

**Example 4:**

You want to archive all log files for sample database on partition 0 only.

```
archive log for db sample on dbpartitionnum (2)
```

**Example 5:**

The following command is the same as example 4.

```
archive log for db sample on all dbpartitionnums except dbpartitionnums  
(1 to 2)
```

**Example 6:**

You want to archive all logs files for sample database on partitions 0 and 2.

```
archive log for db sample on dbpartitionnums(0, 2)
```

**Notes**

- This command can only be executed when the invoking application or shell does not have a database connection to the specified database, otherwise sqlcode SQL1490N will be returned.
- This command can only be executed on a recoverable database; otherwise sqlcode SQL2417N will be returned.
- For compatibility with versions earlier than Version 8, use the keyword NODE instead of DBPARTITIONNUM.

## ATTACH

Description: Enables an application to specify the instance at which instance-level commands (CREATE DATABASE and FORCE APPLICATION, for example) are to be executed. This instance may be the current instance, another instance on the same workstation, or an instance on a remote workstation.

Authorization: None

### Windows/Linux/UNIX Examples

Catalog two remote nodes: (See CATALOG TCP/IP NODE command for information on cataloging remote nodes.)

```
catalog tcpip node node1 remote freedom server 50000
catalog tcpip node node2 remote flash server 50000
```

#### Example 1:

Attach to the first node, force all users, and then detach.

```
attach to node1
force application all
detach
```

#### Example 2:

Attach to the second node, and see who is on.

```
attach to node2
list applications
```

After the command returns agent IDs 1, 2 and 3, force 1 and 3, and then detach.

```
force application (1, 3)
detach
```

#### Example 3:

Attach to the current instance (not necessary, will be implicit), force all users, then detach (AIX only).

```
attach to $DB2INSTANCE
force application all
detach
```

#### Example 4:

Attach to instance db2inst1 using user ID user1 and password.

```
attach to db2inst1 user user1 using abc123
```

#### Example 5:

Attach to instance db2inst1 using user ID user1 and desire to change the password. You will be prompted with for the old password, a new password and a confirmation of the new password.

```
attach to db2inst1 user user1 change password
```

#### Example 6:

Attach to instance db2inst1 using user ID user1 and change the password to newpw1.

```
attach to db2inst1 user user1 using abc123 new newpw1 confirm newpw1
```

### Notes

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

| <b>AUTOCONFIGURE</b>   |
|--|
| <p>Description: Calculates and displays initial values for the buffer pool size,  database configuration and database manager configuration parameters, with the  option of applying these recommended values.</p> <p>Authorization: sysadm</p>  |
| <b>Windows/Linux/UNIX Examples</b>   |
| <p>Example 1:<br/>Display the recommended changes but do not apply them at this time. The AUTOCONFIGURE command will recommend parameter settings on the database, the database manager, and the buffer pools.</p> <pre>autoconfigure apply none</pre> <p>Example 2:<br/>Apply the recommendations given by the AUTOCONFGIURE command on the database configuration only based on the following criterion: The system handles about 1000 transactions per minute.</p> <pre>autoconfigure using tpm 1000 apply db only</pre> <p>Example 3:<br/>Apply the recommendations given by the AUTOCONFGIURE command on both the database configuration and the database manager configuration, based on the following criterion: The system handles about 50 remote connections at any time.</p> <pre>autoconfigure using num_remote_apps 50 apply db and dbm</pre> |
| <b>Notes</b>   |
| <p>Refer to the Command Reference for more criterion keywords.</p>   |

| <b>BACKUP DATABASE</b>   |
|--|
| <p>Description: Creates a backup copy of a database or a table space.</p> <p>Authorization: sysadm/sysctrl/sysmaint</p>  |
| <b>Windows/Linux/UNIX Examples</b>   |
| <p>Example 1:<br/>Backup the SAMPLE database while it is offline. The backup image will be created in your current directory.</p> <pre>backup db sample</pre> <p>Example 2:<br/>Backup the SAMPLE database to the /database1 and /datababase2 directories while the database is still accessible by other users.</p> <pre>backup db sample online to /database1, /database2</pre> <p>Example 3:<br/>Backup the SAMPLE database to TSM using 2 parallel sessions and 4 buffers.</p> <pre>backup database sample use tsm open 2 sessions with 4 buffers</pre> <p>Example 4:<br/>Backup only two table spaces of the SAMPLE database to /dev/rmt0</p> <pre>backup database sample tablespace (syscatspace, userspace1) to /dev/rmt0</pre> <p>Example 5:<br/>Unlike UNIX, where the backup image file is stored directly under directory specified in the BACKUP DATABASE command, on Windows, the BACKUP DATABASE command creates a</p> |

hierarchy of directories. For example, the following command creates the following directories:  
D:\backups\SAMPLE.0\DB2\NODE0000\CATN0000\20040521\122242.001 where 'SAMPLE' is the database name, 'DB2' is the instance name, 'NODE0000' is the partition number, '20040521' is the date when the backup is taken, and '122242.001' is the actual backup image file.

```
backup db sample to D:\backups
```

**Example 6:**

In a partitioned environment, the SAMPLE database is defined on all four partitions, numbered 0 through 3. The path /dev3/backup is accessible from all partitions. Partition 0 is the catalog partition, and needs to be backed-up separately since this is an offline backup. To perform an offline backup of all the SAMPLE database partitions to /dev3/backup, issue the following commands from one of the database partitions:

```
db2_all '<<+0< db2 backup database sample to /dev3/backup'  

db2_all '|<<-0< db2 backup database sample to /dev3/backup'
```

In the second command, the db2\_all utility will issue the same backup command to each database partition in turn (except partition 0). All four database partition backup images will be stored in the /dev3/backup directory.

**BIND**

Description: Invokes the bind utility, which prepares SQL statements stored in the bind file generated by the precompiler, and creates a package that is stored in the database.

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.

**Windows/Linux/UNIX examples**

**Example 1:**

Bind myapp.bnd (the bind file generated when the myapp.sqc program was precompiled) to the database to which a connection has been established.

```
bind myapp.bnd
```

**Example 2:**

You have just upgraded your DB2 client from fixpak 5 to fixpak 6. The next thing you should do is to bind the database utilities to the database server. Without doing this, the client may not be able to use database utilities such as import, export. Change directory to the sqllib/bnd directory (This is where all the DB2 bind files are), issue:

```
connect to proddb user db2inst1 using db2pwd (First connect to the  

remote database 'proddb'.)  

bind @db2cli.lst blocking all grant public  

bind @db2ubind.lst blocking all grant public
```

**Notes**

The character '@' indicates that the file is not a bind file, but a file that contains the names of the bind files. The BIND command will bind all the bind files contained in this file.

**CATALOG APPC NODE**

Description: Adds an APPC node entry to the node directory. The Advanced Program-to-Program Communications protocol is used to access the remote node.

Authorization: sysadm/sysctrl

**Windows/Linux/UNIX examples**

**Example 1:**

To connect to a remote server using the APPC protocol, the APPC software must be installed on

both the client and the server. Prior to performing the `CATALOG APPC NODE` command, the LU to LU communication must have been setup, including the CPIC side information table. In this example, the symbolic destination name of the remote partner node is 'prodserv'. (This is defined in the CPIC side info.)

```
catalog appc node db2appc1 remote prodserv security program with
'Remote APPC node'
terminate
```

#### Notes

- The `TERMINATE` command is needed to refresh the directory cache.
- Use the `LIST NODE DIRECTORY` command to display the existing node entries.

### CATALOG DATABASE

Description: Creates an entry in the system database directory. The database can either be a remote database or a local database.

Authorization: sysadm/sysctrl

#### Windows/Linux/UNIX examples

Example 1:

You want to connect to a database on a remote server with the following characteristics:

Hostname = prodserv

DB2 port number on the server = 50000

OS = AIX

The database name on the remote server = SAMPLE

First, catalog a TCP/IP node. The name of the node is arbitrary. In this example, we are going to use 'node1'. Then you need to catalog the SAMPLE database to node1. However, a SAMPLE database already exists on the local machine; therefore, you have to catalog the remote SAMPLE database using a different name, say MYSAMPLE.

```
catalog tcpip node node1 remote prodserv server 50000 ostype aix
catalog db sample as mysample
terminate
```

Example 2:

Someone has un-cataloged the local SAMPLE database on the system. This does not mean that the database is dropped, but when you do a `LIST DB DIRECTORY` command, the SAMPLE database does not show up anymore. You need to re-catalog the SAMPLE database before anyone can use it again. You need to specify the path/drive where the SAMPLE database was created in. If not, the default path/drive is assumed. For Windows systems, it is C:\, for UNIX systems, it is the instance home directory. For this example, the SAMPLE database was created under the D:\ drive.

```
catalog db sample at D:\
```

#### Notes

1. The `TERMINATE` command is needed to refresh the directory cache.
2. Use the `LIST DB DIRECTORY` command to display the existing node entries.

### CATALOG DCS DATABASE

Description: Stores information about remote host or iSeries databases in the Database Connection Services (DCS) directory. These databases are accessed through an Application Requester (AR), such as DB2 Connect.

Authorization: sysadm/sysctrl

#### Windows/Linux/UNIX examples

Example 1:

You want to connect to a remote OS/390 database server with the following characteristics:

Hostname = prodserv

DB2 port number on the server = 446

Location Name on the server = 'prod1' (This is the database you want to connect to.)

First thing you need to do is to catalog a TCPIP node or an APPC node using the `CATALOG TCPIP NODE` or the `CATALOG APPC NODE` command respectively. Then catalog a database using the `CATALOG DATABASE` command, and a DCS database using the `CATALOG DCS DATABASE` command.

```
catalog tcpip node node1 remote prodserv server 446
catalog database hostdb at node node1
catalog dcs db hostdb as prod1
terminate
```

#### Notes

- The `TERMINATE` command is needed to refresh the directory cache.
- 2. Use the `LIST DCS DIRECTORY` command to display the existing node entries.

### CATALOG LDAP NODE/CATALOG LDAP DATABASE

Description: The `CATALOG LDAP NODE` command catalogs a new node entry in Lightweight Directory Access Protocol (LDAP). The command is used to specify a different alias name for the node that represents the DB2 server. The `CATALOG LDAP DATABASE` command catalogs a database to an ldap node.

Authorization: none

#### Windows/Linux/UNIX examples

Example 1:

Catalog a LDAP node 'ldapnode' and catalog the SAMPLE database as a LDAP database.

```
catalog ldap node ldapnode as ldapnode
catalog ldap database sample at node ldapnode
terminate
```

#### Notes

The `TERMINATE` command is needed to refresh the directory cache.

### CATALOG LOCAL NODE

Description: Creates a local alias for an instance that resides on the same machine. A local node should be cataloged when there is more than one instance on the same workstation to be accessed from the user's client. Interprocess Communications (IPC) is used to access the local node.

Authorization: sysadm/sysctrl

#### Windows/Linux/UNIX examples

Example 1:

A DB2 server has two instances, db2inst1 and db2inst2. You can catalog a local node to represent an instance. In this example, we are going to catalog a local node lnode2 to represent db2inst2.

```
catalog local node lnode2 instance db2inst2
terminate
```

```
attach to lnode2
```

<All the subsequent commands will be applicable to db2inst2>

#### Notes

The `TERMINATE` command is needed to refresh the directory cache.

### CATALOG NAMED PIPE NODE (Windows only)

Description: Adds a Named Pipe node entry to the node directory. A Named Pipe node contains connection information necessary to connect a remote server using the Named Pipe protocol. This command is performed on clients. The information in the node directory only allows the client to reach the remote server. In order to reach the database residing on the remote server, you must also run the `CATALOG DATABASE` command.

Authorization: sysadm/sysctrl

| <b>Windows/Linux/UNIX examples</b>  |
|---|
| <p>Example 1:<br/>Catalog a named pipe node 'db2np1'. The remote server's computer name is 'nphost'. The remote instance name is db2inst1.</p> <pre>catalog npipe node db2np1 remote nphost instance db2inst1</pre>   |
| <b>Notes</b>  |
| <ul style="list-style-type: none"> <li>• The <code>TERMINATE</code> command is needed to refresh the directory cache.</li> <li>• In this example, on the server, the <code>DB2COMM</code> registry variable must be set to <code>NPIPE</code>. (Check it with the <code>db2set -all</code> command.)</li> <li>• Use the <code>LIST NODE DIRECTORY</code> command to display the existing node entries.</li> </ul> |

| <b>CATALOG NETBIOS NODE (Windows only)</b>   |
|--|
| <p>Description: Adds a NetBIOS node entry to the node directory. A NETBIOS node contains connection information necessary to connect a remote server using the NetBios protocol. This command is performed on clients. The information in the node directory only allows the client to reach the remote server. In order to reach the database residing on the remote server, you must also run the <code>CATALOG DATABASE</code> command.</p> <p>Authorization: sysadm/sysctrl</p>  |
| <b>Windows/Linux/UNIX examples</b>   |
| <p>Example:<br/>Catalog a netbios node 'db2netb1'. The remote server's workstation name is 'prodserv'. The outgoing adapter number is 0.</p> <pre>catalog netbios node db2netb1 remote prodserv adapter 0 terminate</pre>  |
| <b>Notes</b>   |
| <ul style="list-style-type: none"> <li>• The <code>TERMINATE</code> command is needed to refresh the directory cache.</li> <li>• In this example, on the server, the <code>NNAME dbm cfg</code> parameter must be set to 'prodserv' (check it with the <code>GET DBM CFG</code> command.) The <code>DB2COMM</code> registry variable must be set to <code>NETBIOS</code>. (Check it with the <code>db2set -all</code> command.)</li> <li>• Use the <code>LIST NODE DIRECTORY</code> command to display the existing node entries.</li> </ul> |

| <b>CATALOG ODBC DATA SOURCE (Windows only)</b>  |
|---|
| <p>Description: Catalogs a user or system ODBC data source. A data source, in ODBC (Open Database Connectivity) terminology, is a user-defined name for a specific database or file system. That name is used to access the database or file system through ODBC APIs. Either user or system data sources can be cataloged. A user data source is only visible to the user who cataloged it, whereas a system data source is visible to and can be used by all other users. This command is available on Windows platforms only.</p> <p>Authorization: none</p> |
| <b>Windows/Linux/UNIX examples</b>  |
| <p>Example 1:<br/>Catalog the SAMPLE database as a system ODBC data source.</p> <pre>catalog system odbc data source sample</pre>   |
| <p>Example 2:<br/>Catalog the SAMPLE database as a user ODBC data source.</p> <pre>catalog user odbc data source sample</pre>   |

| <b>CATALOG TCPIP NODE</b>   |
|---|
| <p>Description: Adds a node entry in the system's node directory. A TCPIP node contains connection information necessary to connect a remote server using the TCPIP protocol. This command is performed on clients. The information in the node directory only allows the client to reach the remote server. In order to reach the database residing on the remote server, you must</p> |

also run the `CATALOG DATABASE` command. In addition, if the remote server is a host system (z/OS, OS/390 or AS/400), you must also run the `CATALOG DCS DATABASE` command.

Authorization: sysadm/sysctrl

#### Windows/Linux/UNIX examples

Example:

You want to connect to a remote server with the following characteristics:

Hostname = prodserv

DB2 port number on the server = 50000

OS = AIX

First thing you need to do is to catalog a TCPIP node. The name of the node is arbitrary. In this example, we are going to use 'node1'.

```
catalog tcpip node node1 remote prodserv server 50000 ostype aix
terminate
```

#### Notes

- The `TERMINATE` command is needed to refresh the directory cache.
- The IP address of the remote host can be used instead of the hostname.
- In this example, on the server, the `SVCENANE dbm cfg` parameter must be set to 50000 or a service name that matches to port 50000 in the services file (Check it with the `GET DBM CFG` command). The `DB2COMM` registry variable must be set to TCPIP. (Check it with the `db2set -all` command.)
- Use the `LIST NODE DIRECTORY` command to display the existing node entries.

### CHANGE DATABASE COMMENT

Description: Changes a database comment in the system database directory or the local database directory. New comment text can be substituted for text currently associated with a comment. This command only affects the database partition on which it is executed.

Authorization: sysadm/sysctrl

#### Windows/Linux/UNIX examples

Example:

Add comment "DB2 default SAMPLE database" to the SAMPLE database.

```
change database sample comment with 'DB2 default SAMPLE database'
```

#### Notes

The comments are displayed in the `LIST DB DIRECTORY` output.

### CHANGE ISOLATION LEVEL

Description: Changes the way that DB2 isolates data from other processes while a database is being accessed. The available isolation levels are: Uncommitted Read (UR), Cursor Stability (CS), Read Stability (RS) and Repeatable Read (RR). RR is the highest isolation level, allows the least amount of concurrency. The default DB2 uses is CS.

Authorization: none

#### Windows/Linux/UNIX examples

Example 1:

For type 1 connection (this is the default), change isolation level to RR. Make sure the current connection is terminated. The new connection will pick up the new isolation level.

```
terminate
```

```
change isolation to rr
```

```
connect to sample
```

Example 2:

For a type 2 connection, the isolation levels can be changed after connections are established. set client connect 2 (Indicate this is a type 2 connection)

```
connect to sample
```

```
connect to test1
```

```

change isolation to rr
set connection sample (Switch current connection to the SAMPLE database.)
<Execute SQL statements on the SAMPLE database using isolation level RR>

change isolation to rs
set connection test1 (Switch current connection to the TEST1 database.)
<Execute SQL statements on the TEST1 database using isolation level RS>

```

### CREATE DATABASE

Description: Initializes a new database with an optional user-defined collating sequence, creates the three initial table spaces, creates the system tables, and allocates the recovery log.  
 Authorization: sysadm/sysctrl

#### Windows/Linux/UNIX examples

Example 1:  
 Create a database called 'test' on /database.  

```
create db test on /database
```

Example 2:  
 Create a database called 'test' in the instance home directory, the database is going to be a Unicode database and with a territory U.S.  

```
create db test using codeset utf-8 territory US
```

Example 3:  
 Create a database called 'test' in the instance home directory. Specify the containers to be used for the three default tablespaces, instead of using the default containers. Also change the USER tablespace from the default SMS type to DMS type, with a file container of 5000 (4K) pages.  

```
create db test \
catalog tablespace managed by system using
('C:\tablespaces\sms\catalog') \
user tablespace managed by database using (file
'C:\tablespaces\dms\user1' 5000) \
temporary tablespace managed by system using
('C:\tablespaces\sms\temp1')
```

### CREATE TOOLS CATALOG

Description: Creates the DB2 tools catalog tables in a new or existing database. The database must be local. The tools catalog contains information about the administrative tasks that you configure with such tools as the Task Center and Control Center. The node from which this command is issued becomes the catalog node for the new database.  
 Multiple DB2 tools catalogs can be created in the same database and are uniquely identified by the catalog name.  
 Authorization: sysadm/sysctrl and dasadm

#### Windows/Linux/UNIX examples

Example:  
 Create the tools catalogs tables in a new TOOLSDB database. The tables will have schema name 'cc'. The default schemas are cc, systools and sysibm if the TOOLSDB was created during DB2 installation.  

```
create tools catalog cc create new database toolsdb
```

### DEACTIVATE DATABASE

Description: Stops the specified database. The database has to be started using the `ACTIVATE DATABASE` command. If the database is started by the first connection to the database, it is stopped when the last connections disconnects. In that case, `DEACTIVATE DATABASE` is not needed.

|   |
|---|
| Authorization: sysadm/sysctrl/sysmaint  |
| <b>Windows/Linux/UNIX examples</b>  |
| Example:<br>Deactivate the SAMPLE database.<br>deactivate db sample   |
| <b>Notes</b>  |
| In an MPP system, this command deactivates the specified database on all database partitions in the system. If one or more of these database partitions encounters an error, a warning is returned. The database will be successfully deactivated on some database partitions, but might continue to be active on the nodes encountering the error. |

|  |
|--|
| <b>DEREGISTER</b>  |
| Description: Deregisters the DB2 server from the network directory server.<br>Authorization: none  |
| <b>Windows/Linux/UNIX examples</b>   |
| Example:<br>Deregister the DB2 server in the LDAP node 'ldapnode'.<br>deregister db2 server in ldap node ldapnode  |
| <b>Notes</b>   |
| This command can only be issued for a remote machine when in the LDAP environment. When issued for a remote machine, the node name of the remote server must be specified. |

|   |
|---|
| <b>DESCRIBE</b>   |
| Description: This command: <ul style="list-style-type: none"> <li>• Displays the output SQLDA information about a SELECT or CALL statement</li> <li>• Displays columns of a table or a view</li> <li>• Displays indexes of a table or a view</li> </ul> Authorization: sysadm/dbadm/CONTROL privilege/SELECT privilege/EXECUTE privilege on the stored procedure. |
| <b>Windows/Linux/UNIX examples</b>  |
| Example 1:<br>Get information on the columns of the SALES table.<br>describe table sales  |
| Example 2:<br>Get information on the indexes of the SALES table.<br>describe indexes for table sales  |
| Example 3:<br>Get information on the output columns of the "select * from sales" statement.<br>describe output select * from sales  |
| Example 4:<br>Get information on the output columns of the call to a stored procedure 'proc1', which takes one integer argument.<br>describe output call proc1(1)   |

|  |
|--|
| <b>DETACH</b>  |
| Description: Removes the logical DBMS instance attachment, and terminates the physical communication connection if there are no other logical connections using this layer.<br>Authorization: none |
| <b>Windows/Linux/UNIX examples</b>   |
| Example:<br>Attach to the instance db2inst1, then detach from it.  |

```
attach to db2inst1
detach
```

#### **DROP CONTACT**

Description: Removes a contact from the list of contacts defined on the local system. A contact is a user to whom the Scheduler and Health Monitor send messages.

Authorization: none

#### **Windows/Linux/UNIX examples**

Example:

Drop the contact name user1.

```
drop contact user1
```

#### **Notes**

- The contact name is added using the `ADD CONTACT` command.
- Use the `GET CONTACTS` command to list existing contacts.

#### **DROP CONTACTGROUP**

Description: Removes a contact group from the list of contacts defined on the local system. A contact group contains a list of users to whom the Scheduler and Health Monitor send messages.

Authorization: none

#### **Windows/Linux/UNIX examples**

Example:

Drop the contact group congr1

```
drop contactgroup congr1
```

#### **Notes**

- The contact groups are created using the `ADD CONTACTGROUP` command.
- Use the `GET CONTACTGROUPS` command to list existing contacts.

#### **DROP DATABASE**

Description: Deletes the database contents and all log files for the database, uncatalogs the database, and deletes the database subdirectory. This command affects all database partitions that are listed in the db2nodes.cfg file.

Authorization: sysadm/sysctrl

#### **Windows/Linux/UNIX examples**

Example 1:

Drop the SAMPLE database.

```
drop db sample
```

Example 2:

In a partitioned database environment, drop the SAMPLE database only on the partition that issues the `DROP DATABASE` command. (Caution: Improper use of the `AT DBPARTITIONNUM` parameter can cause inconsistencies in the system, so it should only be used with caution.)

```
drop db sample at dbpartitionnum
```

#### **DROP DATALINKS MANAGER**

Description: Drops a DB2 Data Links Manager from the list of registered DB2 Data Links Managers for a specified database.

Authorization: sysadm/sysctrl/sysmaint

#### **Windows/Linux/UNIX examples**

Please refer to the Command Reference for examples.

#### **DROP DBPARTITIONNUM VERIFY**

Description: Verifies if a database partition exists in the database partition groups of any databases, and if an event monitor is defined on the database partition. This command should be

|   |
|---|
| used prior to dropping a partition from a partitioned database system.<br>Authorization: sysadm |
| <b>Windows/Linux/UNIX examples</b>  |
| Example:<br>drop dbpartitionnum verify  |
| <b>Notes</b>  |
| This command only affects the database partition on which it is issued.                         |

|  |
|--|
| <b>DROP TOOLS CATALOG</b>  |
| Description: Drops the DB2 tools catalog tables for the specified catalog in the given database.<br>Authorization: sysadm/sysctrl  |
| <b>Windows/Linux/UNIX examples</b>   |
| Example:<br>Drop the tools catalog tables in the TOOLSDB database. Use the <code>FORCE</code> option to force the DB2 administration server's scheduler to stop. If this is not specified, the tools catalog will not be dropped if the scheduler cannot be stopped.<br>drop tools catalog in database toolsdb force |
| <b>Notes</b>   |
| If you drop the active tools catalog, you can no longer schedule tasks and scheduled tasks are not executed. To activate the scheduler, you must activate a previous tools catalog or create a new one using the <code>CREATE TOOLS CATALOG</code> command.  |

|   |
|---|
| <b>EDIT</b>   |
| Description: Launches a user-specified editor with a specified command for editing. When the user finishes editing, saves the contents of the editor and exits the editor, permits the user to execute the command in CLP interactive mode.<br>Authorization: none  |
| <b>Windows/Linux/UNIX examples</b>  |
| Example 1:<br>Use the <code>HISTORY</code> command to display the command history. Then edit the first command and execute it.<br>db2 => history<br>1 select * from sales<br>2 history<br>db2 => edit 1 (Here the default text editor is launched, the "select * from sales" command is displayed in the editor. Add to the command "where region ='Quebec'" and save the changes.)<br>db2 => select * from sales where region='Quebec'<br>Do you want to execute the above command ? (y/n) y<br>(Enter 'y' to execute the command.)<br><br>Example 2:<br>You can specify a different editor to use instead of the default editor. Here we use the editor called 'pico' on UNIX.<br>db2 => edit editor pico 1 |

|   |
|---|
| <b>EXPORT</b>   |
| Description: Exports data from a database to one of several external file formats. The user specifies the data to be exported by supplying an SQL <code>SELECT</code> statement, or by providing hierarchical information for typed tables.<br>Authorization: sysadm/dbadm/CONTROL or <code>SELECT</code> on the table or view. |
| <b>Windows/Linux/UNIX examples</b>  |
| Example 1:<br>Export all rows in the <code>STAFF</code> table to the file <code>myfile.ixf</code> . Write the messages to the <code>msgs.txt</code> file.   |

```
export to myfile.ixf of ixf messages msgs.txt select * from staff
```

**Example 2:**

Export all rows in the EMP\_PHOTO table to the file mefile.del. The LOB data is to be export to separate file(s) under the /tmp/mylobs/ directory. (Note: The lob directory must end with '/')

```
export to myfile.del of del lobs to /tmp/mylobs/ lobfile modified by lobsinfile select * from emp_photo
```

**Example 3:**

Export all rows in the STAFF table to the file myfile.del. Use the character '@' as column delimiters, instead of the default character ','.

```
export to myfile.del of del modified by coldel@ select * from staff
```

**Notes**

Refer to the Command Reference for more options in the EXPORT command.

### FORCE APPLICATION

Description: Forces local or remote applications off the system.

Authorization: sysadm/sysctrl/sysmaint

**Windows/Linux/UNIX examples**

**Example 1:**

Force all applications off the system.

```
force applications all
```

**Example 2:**

Force the applications with application handles 1 and 2. (Use the LIST APPLICATIONS command to display connected applications.)

```
force application (1,2)
```

**Notes**

This command affects all database partitions that are listed in the \$HOME/sql/lib/db2nodes.cfg file. In a partitioned database environment, this command does not have to be issued from the coordinator database partition of the application being forced. It can be issued from any node (database partition server) in the partitioned database environment.

### GET ADMIN CFG

Description: Returns the values of individual DB2 Administration Server (DAS) configuration parameter values on the administration node of the system.

Authorization: none

**Windows/Linux/UNIX Examples**

**Example:**

```
get admin cfg
```

### GET ALERT CFG

Description: Returns the alert configuration settings for health indicators for a particular instance. The indicators are dbm, databases, containers, tablespaces, database, tablespace and container.

Authorization: none

**Windows/Linux/UNIX Examples**

**Example 1:**

Get the alert configuration for databases.

```
get alert cfg for databases
```

**Example 2:**

Get the alert configuration for the database manager.

```
get alert cfg for dbm
```

Example 3:  
Get the alert configuration for the USERSPACE1 table space in the SAMPLE database.  
`get alert cfg for tablespace userspacel on sample`

**GET AUTHORIZATIONS**

Description: Reports the authorities of the current user from values found in the database configuration file and the authorization system catalog view (SYSCAT.DBAUTH).  
Authorization: none

**Windows/Linux/UNIX Examples**

Example:  
`get authorizations`

**GET CLI CFG**

Description: Lists the contents of the db2cli.ini file. This command can list the entire file, or a specified section.  
Authorization: none

**Windows/Linux/UNIX Examples**

Example 1:  
Display the entire content of the db2cli.ini file.  
`get cli cfg`

Example 2:  
Display the TSTCLI1X section of the db2cli.ini file.  
`get cli cfg for section tstclilx`

**GET CONNECTION STATE**

Description: Displays the connection state.  
Authorization:

**Windows/Linux/UNIX Examples**

Example:  
`get connection state`

**GET CONTACTGROUP**

Description: Returns the contacts included in a single contact group that is defined on the local system. A contact is a user to whom the Scheduler and Health Monitor send messages. You create named groups of contacts with the `ADD CONTACTGROUP` command.  
Authorization: none

**Windows/Linux/UNIX Examples**

Example:  
Displays the contact names in the contact group `congr1`  
`get contactgroup congr1`

**GET CONTACTGROUPS**

Description: Provides a list of contact groups, which can be either defined locally on the system or in a global list.  
Authorization: none

**Windows/Linux/UNIX Examples**

Example:  
`get contactgroups`

**GET CONTACTS**

Description: Returns the list of contacts defined on the local system. Contacts are users to whom the monitoring processes such as the Scheduler and Health Monitor send notifications or

|                                    |
|------------------------------------|
| messages.<br>Authorization: none   |
| <b>Windows/Linux/UNIX Examples</b> |
| Example:<br>get contacts           |

|  |
|--|
| <b>GET DB CFG</b>  |
| Description: Returns the values of individual entries in a specific database configuration file. This command returns information only for the partition on which it is executed.<br>Authorization: none                                       |
| <b>Windows/Linux/UNIX Examples</b>   |
| Example 1:<br>Display all the db cfg parameters for the SAMPLE database.<br>get db cfg for sample  |
| Example 2:<br>Display detailed information showing the current value of db cfg parameters of the SAMPLE database as well as the value of the parameters the next time you activate the database.<br>connect to sample<br>get db cfg for sample |

|   |
|---|
| <b>GET DBM CFG</b>  |
| Description: Returns the values of individual entries in a specific database manager configuration file.<br>Authorization:  |
| <b>Windows/Linux/UNIX Examples</b>  |
| Example 1:<br>Display all the dbm cfg parameters.<br>get dbm cfg  |
| Example 2:<br>Display detailed information showing the current value of dbm cfg parameters for the DB2INST1 instance, as well as the value of the parameters the next time you start the instance.<br>attach to db2inst1<br>get dbm cfg |

|   |
|---|
| <b>GET DATABASE MANAGER MONITOR SWITCHES</b>  |
| Description: Displays the status of the database system monitor switches.<br>Authorization: sysadm/sysctrl/sysmaint |
| <b>Windows/Linux/UNIX Examples</b>  |
| Example 1:<br>Get the monitor switches at the current database partition.<br>get dbm monitor switches               |
| Example 2:<br>Get the monitor switches at database partition 1.<br>get dbm monitor switches at dbpartitionnum 1     |
| Example 3:<br>Get the monitor switches at all database partitions.<br>get dbm monitor switches global               |

|  |
|--|
| <b>GET DESCRIPTION FOR HEALTH INDICATOR</b>  |
| Description: Returns a description for the specified health indicator. A Health Indicator measures |

the healthiness of a particular state, capacity, or behavior of the database system. The state defines whether or not the database object or resource is operating normally.

Authorization: none

#### **Windows/Linux/UNIX Examples**

Example:

Get description for the db2.sort\_privmem\_util health indicator.

```
get description for health indicator db2.sort_privmem_util
```

#### **GET HEALTH NOTIFICATION CONTACT LIST**

Description: Returns the list of contacts and contact groups that are notified about the health of an instance. A contact list consists of e-mail addresses or pager Internet addresses of individuals who are to be notified when non-normal health conditions are present for an instance or any of its database objects.

Authorization: none

#### **Windows/Linux/UNIX Examples**

Example:

```
get health notification contact list
```

#### **GET HEALTH SNAPSHOT**

Description: Retrieves the health status information for the database manager and its databases. The information returned represents a snapshot of the health state at the time the command was issued.

Authorization: none

#### **Windows/Linux/UNIX Examples**

Example 1:

Get a health snapshot on the database manager.

```
get health snapshot for dbm
```

Example 2:

Get a health snapshot on all databases at database partition 1

```
get health snapshot for all database at dbpartitionnum 1
```

Example 3:

Get a health snapshot on the SAMPLE database and show detail

```
get health snapshot for database on sample show detail
```

Example 4:

Get a health snapshot on the tablespaces of the SAMPLE database on all partitions.

```
get health snapshot for tablespaces on sample global
```

#### **GET INSTANCE**

Description: Returns the value of the DB2INSTANCE environment variable.

Authorization: none

#### **Windows/Linux/UNIX Examples**

Example:

```
get instance
```

#### **GET MONITOR SWITCHES**

Description: Displays the status of the database system monitor switches for the current session.

Authorization: sysadm/sysctrl/sysmaint

#### **Windows/Linux/UNIX Examples**

Example 1:

Get the monitor switches at the current database partition.

```
get monitor switches
```

Example 2:

Get the monitor switches at database partition 1.

```
get monitor switches at dbpartitionnum 1
```

Example 3:

Get the monitor switches at all database partitions.

```
get monitor switches global
```

### GET RECOMMENDATIONS

Description: Returns descriptions of recommendations for improving the health of the aspect of the database system that is monitored by the specified health indicator. Recommendations can be returned for a health indicator that is in an alert state on a specific object, or the full set of recommendations for a given health indicator can be queried.

Authorization: none

#### Windows/Linux/UNIX Examples

Example 1:

Get recommendations for the health indicator db.db\_heap\_util at current partition.

```
get recommendations for health indicator db.db_heap_util
```

Example 2:

Get recommendations for the health indicator db.db\_heap\_util at partition 1.

```
get recommendations for health indicator db.db_heap_util dbpartitionnum 1
```

### GET ROUTINE

Description: Retrieves a routine SQL Archive (SAR) file for a specified SQL routine.

Authorization: dbadm

#### Windows/Linux/UNIX Examples

Example:

Retrieves the SQL archive of the SQL procedure MYAPPL.PROC1 into the file proc1.sar.

```
get routine into procl.sar from procedure myappl.procl
```

### GET SNAPSHOT

Description: Collects status information and formats the output for the user. The information returned represents a snapshot of the database manager operational status at the time the command was issued.

Authorization: sysadm/sysctrl/sysmaint

#### Windows/Linux/UNIX Examples

Example 1:

Get a complete snapshot on the SAMPLE database.

```
get snapshot for all on sample
```

Example 2:

Get a database manager snapshot on database partition 1.

```
get snapshot for dbm at dbpartitionnum 1
```

Example 3:

Get a snapshot for the application with application handle 189. (Use the LIST APPLICATION command to display currently connected applications.)

```
get snapshot for application agentid 189
```

Example 4:

Get a snapshot on the buffer pools of the SAMPLE database.

```
get snapshot for bufferpools on sample
```

Example 5:

Get a lock snapshot on the SAMPLE database on all database partitions.

```
get snapshot for locks on sample global
```

#### Notes

- In a partitioned database environment, this command can be invoked from any database partition defined in the db2nodes.cfg file. It acts only on that partition.
- Besides the BUFFERPOOLS and the LOCKS snapshots, other available snapshots are TABLES, TABLESPACES, REMOTE\_DATABASES, REMOTE\_APPLICATIONS and DYNAMIC SQL snapshots.

## HISTORY

Description: Displays the history of commands run within a CLP interactive mode session.

Authorization: none

#### Windows/Linux/UNIX examples

Example 1:

List the command history. The most recent command is listed last.

```
db2 => history
1 db2 connect to sample
2 connect to sample
3 select * from sales
4 history
```

Example 2:

List the command history in reverse order. The most recent command is displayed first.

```
db2 => history reverse
5 history reverse
4 history
3 select * from sales
2 connect to sample
1 db2 connect to sample
```

Example 3:

List the 3 most recent commands.

```
db2 => history 3
```

#### Notes

This command can only be run within CLP interactive mode. Specifically, it cannot be run from the CLP command mode or the CLP batch mode.

## IMPORT

Description: Inserts data from an external file with a supported file format into a table, hierarchy, or view. LOAD is a faster alternative, but the load utility does not support loading data at the hierarchy level.

- Authorization: IMPORT using the INSERT option requires one of the following:
  - *sysadm*
  - *dbadm*
  - CONTROL privilege on each participating table or view
  - INSERT and SELECT privilege on each participating table or view
- IMPORT to an existing table using the INSERT\_UPDATE option, requires one of the following:
  - *sysadm*
  - *dbadm*
  - CONTROL privilege on the table or view
  - INSERT, SELECT, UPDATE and DELETE privilege on each participating table or view

- IMPORT to an existing table using the REPLACE or REPLACE\_CREATE option, requires one of the following:
  - *sysadm*
  - *dbadm*
  - CONTROL privilege on the table or view
  - INSERT, SELECT, and DELETE privilege on the table or view
- IMPORT to a new table using the CREATE or REPLACE\_CREATE option, requires one of the following:
  - *sysadm*
  - *dbadm*
  - CREATETAB authority on the database and USE privilege on the table space, as well as one of:
    - IMPLICIT\_SCHEMA authority on the database, if the implicit or explicit schema name of the table does not exist
    - CREATIN privilege on the schema, if the schema name of the table refers to an existing schema
- IMPORT to a hierarchy that does not exist using the CREATE, or the REPLACE\_CREATE option, requires one of the following:
  - *sysadm*
  - *dbadm*
  - CREATETAB authority on the database and USE privilege on the table space and one of:
    - IMPLICIT\_SCHEMA authority on the database, if the schema name of the table does not exist
    - CREATEIN privilege on the schema, if the schema of the table exists
    - CONTROL privilege on every sub-table in the hierarchy, if the REPLACE\_CREATE option on the entire hierarchy is used
- IMPORT to an existing hierarchy using the REPLACE option requires one of the following:
  - *sysadm*
  - *dbadm*
  - CONTROL privilege on every sub-table in the hierarchy

### Windows/Linux/UNIX examples

#### Example 1:

Import data from file myfile.ixf and insert into the STAFF table. Write the messages to file msg.txt.  
`import from myfile.ixf of ixf messages msg.txt insert into staff`

#### Example 2:

Import data from file myfile.del. The columns in the file are delimited by the '@' character instead of the default character ','. The data is going to be inserted into the STAFF table, or updates existing rows (of the target table) with matching primary keys.

`import from myfile.del of del modified by coldel@ insert_update into staff`

#### Example 3:

Import data from a ASC file, myfile.asc, replace into table T1. The data.asc file has the following contents:

1.....5....10....15...20...25...30. (This row is not in the file. It is here to show the positions of the characters in the data.asc file.)

```
1234 mary   Texas
3242 tom    Washington
0352 sandra Michigan
```

`import from myfile.asc of asc method L (1 4, 5 13, 15 30) replace into t1`

Note: The pairs in the method L clause specify the begin and end of a column. For example, column 1 starts at position 1 and ends at position 4. Column 2 starts at position 5 and ends at

position 13.

Example 4:

Import data from myfile.ixf and use the data to create a new table T2. Stop the import process if the number of warnings reaches 20. (Note: The REPLACE\_CREATE option is only available when the input file is of type IXF. Because only IXF files contain information to create a table and indexes.)

```
import from myfile.ixf of ixf warningcount 20 replace_create into t2
```

#### Notes

Please refer to the Command Reference for more IMPORT options.

### INITIALIZE TAPE (Windows only)

Description: When running on Windows NT-based operating systems, DB2 supports backup and restore operations to streaming tape devices. Use this command for tape initialization.

Authorization: none

#### Windows/Linux/UNIX examples

Example 1:

Initialize [\\.\TAPE0](#), use the default block size.

```
initialize tape on \\.\TAPE0
```

Example 2:

Initialize [\\.\TAPE0](#) and specify to use a variable length block size. (Note: If the device does not support variable length block mode, an error will be returned.)

```
initialize tape on \\.\TAPE0 using 0
```

### INSPECT

Description: Inspects database for architectural integrity, checking the pages of the database for page consistency. The inspection checks whether the structures of table objects and structures of table spaces are valid. This command is used while the database is online. A connection to the database is required.

Authorization: sysadm/dbadm/sysctrl/sysmaint/CONTROL privilege if single table

#### Windows/Linux/UNIX examples

Example 1:

Inspect the whole SAMPLE database. Write the results to file inspect.out. The results file will be generated in the DB2 diagnostic data directory (e.g. where db2diag.log is). It is a binary file - you need to format it with the DB2INSPF command. If no errors are found, the results file will be erased after the inspect operation is complete. (To keep the results file regardless, use the KEEP option. See example 2.)

```
connect to sample
inspect check database results inspect.out
```

Example 2:

Inspect the tablespace with tablespace ID 2. Keep the result and write it to file inspect.out.

```
connect to sample
inspect check tablespace tbspaceID 2 results keep inspect.out
```

Example 3:

Inspect the SAMPLE database only on database partitions 0 and 1. If there are errors, write them to file inspect.out.

```
connect to sample
inspect check database results inspect.out on dbpartitionnums (0,1)
```

Example 4:

Inspect the SAMPLE database only on all database partitions except 0. If there are errors, write them to file inspect.out.

```
connect to sample
inspect check database results inspect.out on all dbpartitionnums
except dbpartitionnums(0)
```

**Example 5:**

Inspect the SALES table. If there are errors, write them to file inspect.out.

```
connect to sample
inspect check table name sales results inspect.out
```

**Notes**

In a single-partition system, the scope is that single partition only. In a partitioned database system, it is the collection of all logical partitions defined in db2nodes.cfg.

### LIST ACTIVE DATABASES

Description: Displays a subset of the information listed by the GET SNAPSHOT FOR ALL DATABASES command. An active database is available for connection and use by any application. For each active database, this command displays the following:

- Database name
- Number of applications currently connected to the database
- Database path.

Authorization: sysadm/sysctrl/sysmain

**Windows/Linux/UNIX examples**

**Example 1:**

List the active databases on the current database partition.

```
list active databases
```

**Example 2:**

List the active databases on the database partition 1.

```
list active databases at dbpartitionnum 1
```

**Example 3:**

List the active databases on all database partitions.

```
list active databases global
```

### LIST APPLICATIONS

Description: Displays to standard output the application program name, authorization ID (user name), application handle, application ID, and database name of all active database applications. This command can also optionally display an application's sequence number, status, status change time, and database path.

Authorization: sysadm/sysctrl/sysmaint

**Windows/Linux/UNIX examples**

**Example 1:**

List all applications that are currently connected to any databases at the current database partition.

```
list applications
```

**Example 2:**

List all applications that are currently connect to the SAMPLE database at the current database partition.

```
list applications for db sample
```

**Example 3:**

List all applications that are currently connect to the SAMPLE database at the current database partition and show detailed info.

```
list applications for db sample show detail
```

Example 4:  
List all applications that are currently connected to any databases on all database partitions.  
`list applications global`

Example 5:  
List all applications that are currently connect to the SAMPLE database at the database partition 1.  
`list applications at dbpartitionnum 1`

#### DB2 LIST COMMAND OPTIONS

Description: List the current setting for the environment variables: DB2BQTIME, DB2DQTRY, DB2RQTIME, DB2IQTIME, DB2OPTIONS.  
Authorization: none

#### Windows/Linux/UNIX examples

Example:  
`list command options`

#### Notes

The settings can be changed using the `UPDATE COMMAND OPTIONS` command.

#### LIST DATABASE DIRECTORY

Description: List the contents of the system database directory containing both local and remote databases. If a path is specified, the contents of the local database directory are listed.  
Authorization: none

#### Windows/Linux/UNIX examples

Example 1:  
List both local and remote databases.  
`list db directory`

Example 2:  
List only the local databases residing on /home/db2inst1 (UNIX)  
`list db directory on /home/db2inst1`

Example 3:  
List only the local databases residing on C:\. (Windows)  
`list db directory on C:\`

#### LIST DATABASE PARTITION GROUPS

Description: List all database partition groups associated with the current database.  
Authorization: sysadm/dbadm/CONTROL privilege/SELECT privilege

#### Windows/Linux/UNIX examples

Example 1  
`list database partition groups`

Example 2:  
`list database partition groups show detail`

#### LIST DATALINKS MANAGERS

Description: Lists the DB2 Data Links Managers that are registered to a specified database.  
Authorization: none

#### Windows/Linux/UNIX examples

Example:  
List the data links managers registered to the SAMPLE database.  
`list datalinks managers for db sample`

| <b>LISA DBPARTITIONENUMS</b>   |
|--|
| Description: List all database partitions associated with the current database.<br>Authorization: none |
| <b>Windows/Linux/UNIX examples</b>   |
| Example:<br><code>list dbpartitionnums</code>  |

| <b>LIST DCS APPLICATIONS</b>   |
|--|
| Description: List the applications that are connected to host databases.<br>Authorization: sysadm/sysctrl/sysmaint |
| <b>Windows/Linux/UNIX examples</b>   |
| Example 1:<br><code>list dcs applications</code>   |
| Example 2:<br>List dcs applications and show detail<br><code>list dcs applications show detail</code>              |
| Example 3:<br>Get an extensive report on the dcs applications.<br><code>list dcs applications extended</code>      |

| <b>LIST DCS DIRECTORY</b>  |
|--|
| Description: List the contents of the Database Connections Services (DCS) directory. This directory is only present when host databases are cataloged on the machine.<br>Authorization: none |
| <b>Windows/Linux/UNIX examples</b>   |
| Example:<br><code>list dcs directory</code>  |
| <b>Notes</b>   |
| A dcs entry is created when a CATALOG DCS DB DIRECTORY command is entered.   |

| <b>LIST DRDA INDOUBT TRANSACTIONS</b>   |
|---|
| Description: Provides a list of transactions that are indoubt between DRDA requesters and DRDA servers. If APPC commit protocols are being used, lists indoubt transactions between partner LUs. If DRDA commit protocols are being used, lists indoubt transactions between DRDA sync point managers.<br>Authorization: sysadm |
| <b>Windows/Linux/UNIX examples</b>  |
| Example 1:<br>List the DRDA indoubt transactions.<br><code>list drda indoubt transactions</code>  |
| Example 2:<br>List the DRDA indoubt transactions and initiate an interactive dialog mode, permitting user to commit or roll back<br><code>list drda indoubt transactions with prompting</code>  |
| <b>Notes</b>  |
| A forget option is not supported. Once the indoubt transaction is committed or rolled back, the transaction is automatically forgotten.   |

| <b>LIST HISTORY</b>   |
|---|
| Description: Lists entries in the history file. The history file contains a record of recovery and administrative events. The following events can be used in the LIST HISTORY command: |

BACKUP, ROLLFORWARD, DROPPED TABLE, LOAD, CREATE TABLESPACE, ALTER TABLESPACE, RENAME TABLESPACE, REORG, ARCHIVE LOG.

Authorization: none

#### **Windows/Linux/UNIX examples**

Example 1:

List all the backup operations performed on the SAMPLE database.

```
list history backup all for sample
```

Example 2:

List all the rollforward operations performed on the SAMPLE database since 2004-01-01-00:00:00.

```
list history rollforward since 20040101000000 for sample
```

Example 3:

List all the create tablespace operations on tablespace TS1 in the SAMPLE database.

```
list history create tablespace containing ts1 for sample
```

Example 4:

List all the load operations on the table sylviaq.sales in the SAMPLE database.

```
list history load containing sylviaq.sales for sample
```

#### **LIST INDOUBT TRANSACTIONS**

Description: Provides a list of transactions that are indoubt. The user can interactively commit, roll back, or forget the indoubt transactions.

Authorization: dbadm

#### **Windows/Linux/UNIX examples**

Example 1:

List the indoubt transactions.

```
list indoubt transactions
```

Example 2:

List the indoubt transactions and initiate an interactive dialog mode, permitting user to commit, roll back or forget indoubt transactions.

```
list indoubt transactions with prompting
```

#### **LIST NODE DIRECTORY**

Description: Lists the contents of the node directory.

Authorization: none

#### **Windows/Linux/UNIX examples**

Example 1:

List node directory and show detailed info.

```
list node directory show detail
```

Example 2:

List the admin node directory

```
list admin node directory
```

#### **LIST ODBC DATA SOURCES (Windows only)**

Description: Lists all available system or user ODBC data sources. A data source, in ODBC (Open Database Connectivity) terminology, is a user-defined name for a specific database. That name is used to access the database or file system through ODBC APIs. On Windows, either user or system data sources can be cataloged. A user data source is only visible to the user who cataloged it, whereas a system data source is visible to and can be used by all other users.

|   |
|---|
| Authorization: none   |
| <b>Windows/Linux/UNIX examples</b>  |
| <p>Example 1:<br/>List user ODBC data sources.<br/><code>list user odbc data sources</code></p> <p>Example 2:<br/>List system ODBC data sources.<br/><code>list system odbc data sources</code></p> |

|   |
|---|
| <b>LIST PACKAGES/TABLES</b>   |
| <p>Description: Lists packages or tables associated with the current database.<br/>Authorization: For the system catalog SYSCAT.PACKAGES (LIST PACKAGES) and SYSCAT.TABLES (LIST TABLES), one of the following is required:</p> <ul style="list-style-type: none"> <li>• <i>sysadm</i> or <i>dbadm</i> authority</li> <li>• CONTROL privilege</li> <li>• SELECT privilege.</li> </ul>   |
| <b>Windows/Linux/UNIX examples</b>  |
| <p>Example 1:<br/>List all system catalog tables and details.<br/><code>list tables for system show detail</code></p> <p>Example 2:<br/>List all user tables under schema 'sylviaq'.<br/><code>list tables for schema sylviaq</code></p> <p>Example 3:<br/>List all tables in the database.<br/><code>list tables for all</code></p> <p>Example 4:<br/>List all user tables for the current user.<br/><code>list tables for user</code></p> |
| <b>Notes</b>  |
| Replace the keyword 'table' by 'packages' in the above examples to get package info.  |

|  |
|--|
| <b>LIST TABLESPACE CONTAINERS</b>  |
| <p>Description: List container information for the specified table space.<br/>Authorization: <i>sysadm/sysctrl/sysmaint/dbadm</i></p>  |
| <b>Windows/Linux/UNIX examples</b>   |
| <p>Example 1:<br/>Get container information for table space 1.<br/><code>list tablespace containers for 1</code></p> <p>Example 2:<br/>Get detailed container information for table space 1.<br/><code>list tablespace containers for 1 show detail</code></p> |
| <b>Notes</b>   |
| <ul style="list-style-type: none"> <li>• Use the LIST TABLESPACES command to get the table space ID.</li> <li>• This command returns information only for the node on which it is executed.</li> </ul>   |

|   |
|---|
| <b>LIST TABLESPACES</b>   |
| Description: List table space information for the current database. |

|   |
|---|
| Authorization: sysadm/sysctrl/sysmaint/dbadm/load   |
| <b>Windows/Linux/UNIX examples</b>  |
| <p>Example 1:<br/>list tablespaces</p> <p>Example 2:<br/>List table spaces and their detailed information.<br/>list tablespaces show detail</p> |
| <b>Notes</b>  |
| This command returns information only for the node on which it is executed.   |

|   |
|---|
| <b>LIST UTILITIES</b>   |
| <p>Description: Displays to standard output the list of active utilities on the instance. The description of each utility can include attributes such as start time, description, throttling priority (if applicable), as well as progress monitoring information (if applicable).<br/>Authorization: sysadm/sysctrl/sysmaint</p> |
| <b>Windows/Linux/UNIX examples</b>  |
| <p>Example:<br/>list utilities show detail</p>  |
| <b>Notes</b>  |
| This command only returns information for the database partition on which it is issued.   |

|   |
|---|
| <b>LOAD</b>   |
| <p>Description: Loads data into a DB2 table. Data residing on the server can be in the form of a file, tape, or named pipe. Data residing on a remotely connected client can be in the form of a fully qualified file or named pipe. Data can also be loaded from a user-defined cursor. The available operations a load can perform on a table are: INSERT, REPLACE, RESTART, TERMINATE.<br/>Authorization: One of the following:</p> <ul style="list-style-type: none"> <li>• sysadm</li> <li>• dbadm</li> <li>• load authority on the database and <ul style="list-style-type: none"> <li>○ INSERT privilege on the table when the load utility is invoked in INSERT mode, TERMINATE mode (to terminate a previous load insert operation), or RESTART mode (to restart a previous load insert operation)</li> <li>○ INSERT and DELETE privilege on the table when the load utility is invoked in REPLACE mode, TERMINATE mode (to terminate a previous load replace operation), or RESTART mode (to restart a previous load replace operation)</li> <li>○ INSERT privilege on the exception table, if such a table is used as part of the load operation.</li> </ul> </li> </ul> |
| <b>Windows/Linux/UNIX examples</b>  |
| <p>Example 1:<br/>Load data from a DEL file, data.del, replace into table T1. Establish consistency point every 5000 rows. (This means should the load fails, when it is restarted, it will pick up from the last consistency point.) Warnings and errors will be written to file /tmp/loadt1.msg.<br/>load from data.del of del savecount 5000 messages loadt1.msg replace into t1</p> <p>Example 2:<br/>Load data from an IXF file, data.ixf, insert into table T1. T1 has one LOB column. The LOB data that is to be loaded is stored in the files under the directory /tmp/t1lobs/. (Note: The path to the LOB data files must end with '.').<br/>load from data.ixf of ixf lobs from /tmp/t1lobs/ modified by lobsinfile insert into t1</p> <p>Example 3:</p>  |

Load data from a ASC file, data.asc, insert into table T1. The data.asc file has the following contents:

1.....5.....10.....15.....20.....25.....30. (This row is not in the file. It is here to show the positions of the characters in the data.asc file.)

```
1234 mary   Texas
3242 tom    Washington
0352 sandra Michigan
```

```
load from data.asc of asc method L (1 4, 5 13, 15 30) insert into t1
```

Note: The pairs in the method L clause specify the begin and end of a column. For example, column 1 starts at position 1 and ends at position 4. Column 2 starts at position 5 and ends at position 13.

Example 4:

Load all data from table T1 and insert into table T2, using a cursor.

```
declare mycurs cursor for select * from t1
load from mycurs of cursor insert into t2
```

#### Notes

- See the DB2 Command reference for more load examples.
- If a load operation failed, you can restart or terminate the load using the RESTART or the TERMINATE keyword. Just run the same load command again, but replace the INSERT or REPLACE keywords with RESTART or TERMINATE.

#### LOAD QUERY

Description: Checks the status of a load operation during processing and returns the table state. If a load is not processing, then the table state alone is returned. A connection to the same database, and a separate CLP session are also required to successfully invoke this command. It can be used either by local or remote users.

Authorization: none

#### Windows/Linux/UNIX examples

Example 1:

Check the status of a load operation on the STAFF table. Write the warning and errors to a message file '/tmp/loadstaff.msg'

```
load query table staff to /tmp/loadstaff.msg
```

Example 2:

Check the status of a load operation on the STAFF table. Write the warning and errors to a message file '/tmp/loadstaff.msg'. Only new information (pertaining to load events that have occurred since the last invocation of the LOAD QUERY command) is to be reported.

```
load query table staff to /tmp/loadstaff.msg
```

#### MIGRATE DATABASE

Description: Converts previous versions of DB2 databases to current version.

Authorization: sysadm

#### Windows/Linux/UNIX examples

Example:

Migrate the SAMPLE database after the DB2 product has been upgraded from v7 to v8 and the instance has been migrated using the db2imigr command. (On Windows systems, the instance does not need to be upgraded.)

```
migrate db sample
```

#### Notes

- Use the db2ckmig prior to running the MIGRATE DATABASE command to verify that the database is ready for migration.

- If failed to migrate a database using the `MIGRATE DATABASE` command, as a workaround, you may try taking a backup image of the old version of the database and restore it into the new version. Database migration will be performed automatically during the restore process.

### PING

Description: Tests the network response time of the underlying connectivity between a client and a connected database server.

Authorization: none

#### Windows/Linux/UNIX examples

Example 1:

Ping the SAMPLE database once.

```
ping sample
```

```
Elapsed time: 26088 microseconds
```

Example 2: Ping the SAMPLE database 3 times.

```
ping sample 3
```

```
Elapsed time: 82 microseconds
```

```
Elapsed time: 31 microseconds
```

```
Elapsed time: 22 microseconds
```

### PRECOMPILE

Description: Processes an application program source file containing embedded SQL statements. Produces a modified source file containing host language calls for the SQL statements and, by default, a package is created in the database.

Authorization: sysadm/dbadm/BINDADD privilege if the package does not exist and one of (IMPLICIT\_SCHEMA authority on the database if the schema name of the package does not exist; or 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.

#### Windows/Linux/UNIX examples

Example 1:

Precompile the sample.sqc program and generate a bind file. The bind file will be named 'sample.bnd'.

```
prep sample.sqc bindfile
```

Example 2:

Precompile the sample.sqc program and generate a bind file. The bind file will be named 's.bnd'.

```
prep sample.sqc bindfile using s.bnd
```

Example 3:

Precompile the sampe.sqc program using isolation level Uncommitted Read.

```
prep dbauth.sqc isolation UR
```

Example 4:

Precompile the sampe.sqc program using the IBM standard U.S. data and time format. The owner of the package is 'sylviaq'.

```
prep dbauth.sqc datetime usa owner sylviaq
```

### PRUNE HISTORY/LOGFILE

Description: Deletes entries from the recovery history file, or to delete log files from the active log file path. Deleting entries from the recovery history file might be necessary if the file becomes excessively large and the retention period is high.

Authorization: sysadm/sysctrl/sysmaint/dbadm

#### Windows/Linux/UNIX examples

Example 1:

Remove the entries for all restores, loads, table space backups, and full database backups taken

before and including December 1, 1999 from the recovery history file.  
prune history 199912

Example 2:

Remove all log files prior to (but not including) log file S0000245.LOG.  
prune logfile prior to S0000245.LOG

#### Notes

Be very careful when pruning your log files. Make sure they are not needed anymore.

### PUT ROUTINE

Description: Uses the specified routine SQL Archive (SAR) file to define a routine in the database.  
Authorization: dbadm

#### Windows/Linux/UNIX examples

Example:

Use the inventory.sar file to define a routine. The user 'db2inst1' is used for authorization checking of the routine. The CURRENT SCHEMA and CURRENT PATH special registers are used to define the routine. (If the USE REGISTERS clause is not specified, the settings for the default schema and SQL path are the settings used when the routine is defined.)  
put routine from inventory.sar owner db2inst1 use registers

### QUERY CLIENT

Description: Returns current connection settings for an application process.  
Authorization: none

#### Windows/Linux/UNIX examples

Example:

Query the connection settings of current database connection.

```
query client
```

The current connection settings of the application process are:

```
CONNECT = 1
DISCONNECT = EXPLICIT
MAX_NETBIOS_CONNECTIONS = 1
SQLRULES = DB2
SYNCPOINT = ONEPHASE
CONNECT_DBPARTITIONNUM = -1
ATTACH_DBPARTITIONNUM = -1
```

#### Notes

You can use the SET CLIENT command to change these settings.

### QUIESCE

Description: Forces all users off the specified instance and database and puts it into a quiesced mode. In quiesced mode, users cannot connect from outside of the database engine. While the database instance or database is in quiesced mode, you can perform administrative tasks on it. After administrative tasks are complete, use the UNQUIESCE command to activate the instance and database and allow other users to connect to the database but avoid having to shut down and perform another database start.

In this mode only users with authority in this restricted mode are allowed to attach or connect to the instance/database. Users with sysadm, sysmaint, and sysctrl authority always have access to an instance while it is quiesced, and users with sysadm and dbadm authority always have access to a database while it is quiesced.

Authorization: sysadm/dbadm (for database level quiesce)  
sysadm/sysctrl (for instance level quiesce)

#### Windows/Linux/UNIX examples

**Example 1:**

Quiesce the db2inst1 instance. All connections will be forced off immediately. Allow user FRANK access to the instance while it is quiesced.

```
quiesce instance db2inst1 user frank immediate
```

**Example 2:**

Quiesce the db2inst1 instance, but wait for applications until they commit the current unit of work. Allow users in the db2users1 group access to the instance while it is quiesced.

```
quiesce instance db2inst1 group db2users1 defer
```

**Example 3:**

Quiesce the SAMPLE database, but wait for applications until they commit the current unit of work, with a timeout of 15 minutes. If applications still have not committed after the timeout period, they will be forced off. (A connection to the SAMPLE database is needed.)

```
quiesce db defer with timeout 15
```

**Notes**

- After `QUIESCE INSTANCE`, only users with `sysadm`, `sysmaint`, or `sysctrl` authority or a user name and group name provided as parameters to the command can connect to the instance.
- After `QUIESCE DATABASE`, users with `sysadm`, `sysmaint`, `sysctrl`, or `dbadm` authority, and `GRANT/REVOKE` privileges can designate who will be able to connect. This information will be stored permanently in the database catalog tables. For example, `grant quiesce_connect on database to <username/groupname>`  
`revoke quiesce_connect on database from <username/groupname>`

**QUIESCE TABLESPACES FOR TABLE**

Description: Quiesces table spaces for a table. There are three valid quiesce modes:

- Share
- Intent to update
- Exclusive

Authorization: `sysadm/sysctrl/sysmaint/dbadm/load`

**Windows/Linux/UNIX examples****Example 1:**

Quiesce the tablespaces for table 'sylviaq.sales' in exclusive mode.

```
quiesce tablespaces for table sylviaq.sales exclusive
```

**Example 2:**

Reset the tablespaces for table 'sylviaq.sales' to normal state.

```
quiesce tablespaces for table sylviaq.sales reset
```

**Notes**

In a single-partition environment, this command quiesces all table spaces involved in a load operation in exclusive mode for the duration of the load operation. In a partitioned database environment, this command acts locally on a node. It quiesces only that portion of table spaces belonging to the node on which the load operation is performed.

**QUIT**

Description: Exits the command line processor interactive input mode and returns to the operating system command prompt. If a batch file is being used to input commands to the command line processor, commands are processed until `QUIT`, `TERMINATE`, or the end-of-file is encountered.

Authorization: none

**Windows/Linux/UNIX examples****Example:**

Enter the CLP interactive mode by typing "db2" from the command window. Then exit it using `QUIT`. You will be returned to command prompt.

```
db2 => quit
DB20000I The QUIT command completed successfully.
D:\temp>
```

**Notes**

QUIT does not terminate the command line processor back-end process or break a database connection. CONNECT RESET breaks a connection, but does not terminate the back-end process. The TERMINATE command does both.

**REBIND**

Description: Allows the user to recreate a package stored in the database without the need for a bind file.

Authorization: sysadm/dbadm/ALTERIN privilege on the schema/BIND privilege on the package.

**Windows/Linux/UNIX examples**

Example 1:

```
Rebind the package sylviaq.P0059800
rebind package sylviaq.P0059800
```

Example 2:

Rebind the package sylviaq.P0059800 using conservative binding semantics. (The default is not to use it.)

```
rebind package sylviaq.P0059800 resolve conservative
```

**RECONCILE**

Description: Validates the references to files for the DATALINK data of a table. The rows for which the references to files cannot be established are copied to the exception table (if specified), and modified in the input table.

Authorization: sysadm/sysctrl/sysmaint/dbadm/CONTROL privilege on the table.

**Windows/Linux/UNIX examples**

Example:

Reconcile the table DEPT, and writes exceptions to the exception table EXCPTAB, which was created by the user. Information about files that were unlinked during reconciliation is written into the file report.ulk, which is created in the directory /u/johnh. If FOR EXCEPTION excptab had not been specified, the exception information would have been written to the file report.exp, created in the /u/johnh directory.

```
reconcile dept dlreport /u/johnh/report for exception excptab
```

**Notes**

Reconcile produces a message file (reconcil.msg) in the instance path on UNIX based systems, and in the install path on Windows platforms. This file will contain warning and error messages that are generated during validation of the exception table.

**REDISTRIBUTE DATABASE PARTITION GROUP**

Description: Redistributes data across the database partitions in a database partition group. The current data distribution, whether it is uniform or skewed, can be specified. The redistribution algorithm selects the partitions to be moved based on the current data distribution.

This command can only be issued from the catalog database partition. Use the LIST DATABASE DIRECTORY command to determine which database partition is the catalog database partition for each database.

Authorization: sysadm/sysctrl/dbadm

**Windows/Linux/UNIX examples**

Example 1:

```
Redistribute the data in partition group gr1 uniformly.
redistribute database partition group gr1 uniform
```

**Example 2:**

Redistribute the data in partition group gr1 according to the distribution file, distfile1. The distribution file indicates the current distribution of data across the 4 096 hash partitions.

```
redistribute database partition group gr1 using distfile
/tmp/distfile1
```

**Example 3:**

Redistribute the data in partition group gr1 according to the order specified in the target map file, map1.

```
redistribute database partition group gr1 using targetmap /tmp/map1
```

**Example 4:**

Continue a previously failed redistribute database partition group operation.

```
redistribute database partition group gr1 continue
```

**Example 5:**

Roll back a previously failed redistribute database partition group operation.

```
redistribute database partition group gr1 rollback
```

**Notes**

When a redistribution operation is done, a message file is written to:

- The /sqlib/redist directory on UNIX based systems, using the following format for subdirectories and file name: database-name.database-partition-group-name.timestamp.
- The \sqlib\redist\ directory on Windows operating systems, using the following format for subdirectories and file name: database-name\first-eight-characters-of-the-database-partition-group-name\date\time.

**REFRESH LDAP**

Description: Refreshes the cache on a local machine with updated information when the information in Lightweight Directory Access Protocol (LDAP) has been changed.

Authorization: none

**Windows/Linux/UNIX examples**

**Example 1:**

Refresh the LDAP database directory.

```
refresh ldap db dir
```

**Example 2:**

Refresh the ldap node directory.

```
refresh ldap node dir
```

**REGISTER**

Description: Registers the DB2 server in the network directory server.

Authorization: none

**Windows/Linux/UNIX examples**

**Example:**

Register the DB2 server in the LDAP directory. The node name 'tcpnode' represents the DB2 server in LDAP. Clients can attach to the server using this node name. The protocol used is TCPIP.

```
register db2 server in ldap as tcpnode protocol tcpip
```

**REORG INDEXES/TABLE**

Description: Reorganizes an index or a table. The index option reorganizes all indexes defined on a table by rebuilding the index data into un-fragmented, physically contiguous pages. If you specify the CLEANUP ONLY option of the index option, cleanup is performed without rebuilding the indexes. This command cannot be used against indexes on declared temporary tables. The table option reorganizes a table by reconstructing the rows to eliminate fragmented data, and by

compacting information.

Authorization: sysadm/sysctrl/sysmaint/dbadm/CONTROL privilege on the table

#### **Windows/Linux/UNIX examples**

Example 1:

Reorganize the table 'sylviaq.sales' to reclaim space. Use the temporary tablespace tempsp1 for this operation. (If no temporary tablespace is provided, the REORG operation will use the tablespace where the table exists.)

```
reorg table sylviaq.sales use tempsp1
```

Example 2:

Reorganize the table 'employee' and index 'empid' in a partitioned environment on partitions 1 and 2 only.

```
reorg table employee index empid on dbpartitionnum (1,2)
```

Example 3:

To clean up the pseudo deleted keys and pseudo empty pages in all the indexes on the table 'sylviaq.employee' while allowing other transactions to read and update the table.

```
reorg indexes all for table sylviaq.employee allow write access cleanup only
```

Example 4:

Reorganize table T1 which contains LOB data.

```
reorg table t1 allow read access longlobdata
```

#### **REORGCHK**

Description: Calculates statistics on the database to determine if tables or indexes, or both, need to be reorganized or cleaned up.

Authorization: sysadm/dbadm/CONTROL privilege on the table

#### **Windows/Linux/UNIX examples**

Example 1:

Call the RUNSTATS routine to update statistics on the system catalog tables, and then use the updated statistics to determine if table reorganization is required.

```
reorgchk update statistics on table system
```

Example 2:

Call the RUNSTATS routine to update statistics on all the user tables, and then use the updated statistics to determine if table reorganization is required.

```
reorgchk update statistics on table user
```

Example 3:

Use the current statistics to check if a table reorganization is required on the table 'sylviaq.sales'.

```
reorgchk current statistics on table sylviaq.sales
```

Example 4:

Use the current statistics to check if a table reorganization is required on all tables (both system and user tables).

```
reorgchk current statistics on table all
```

Example 5:

Use the current statistics to check if a table reorganization is required on all tables under schema 'sylviaq'.

```
reorgchk current statistics on schema sylviaq
```

#### **RESET ADMIN CFG**

Description: Resets the parameters in the DAS configuration.

|  |
|--|
| Authorization: dasadm                              |
| <b>Windows/Linux/UNIX examples</b>                 |
| Example:<br>reset admin cfg                        |
| <b>Notes</b>                                       |
| Use GET ADMIN CFG to display the current settings. |

|  |
|--|
| <b>RESET ALTER CONFIGURATION</b>   |
| Description: Resets the health indicator settings for specific objects to the current defaults for that object type or resets the current default health indicator settings for an object type to the install defaults. The available object types are: dbm, databases, tablespaces, containers, database, tablespace <name>, container <name> for <tablespace-id><br>Authorization: sysadm/sysctrl/sysmaint |
| <b>Windows/Linux/UNIX examples</b>   |
| Example 1:<br>Reset the all health indicators for all databases in the instance.<br>reset alert cfg for databases  |
| Example 2:<br>Reset the health indicator tsc.tscont_util on all containers of all databases in this instance.<br>reset alert cfg for containers using tsc.tscont_util  |
| Example 3:<br>Reset the tsc.tscont_util health indicator for the container 'C:\DB2\NODE0000\SQL00002\SQLT0002.0' in tablespace 2 in the SAMPLE database.<br>reset alert cfg for container C:\DB2\NODE0000\SQL00002\SQLT0002.0 for 2 on sample using tsc.tscont_util  |

|  |
|--|
| <b>RESET DB CFG</b>  |
| Description: Resets the configuration of a specific database to the system defaults.<br>Authorization: sysadm/sysctrl/sysmaint |
| <b>Windows/Linux/UNIX examples</b>   |
| Example:<br>Reset of the configuration parameters for the SAMPLE database.<br>reset db cfg for sample                          |
| <b>Notes</b>   |
| In a partitioned environment, this command only affects the partition on which it is executed.                                 |

|   |
|---|
| <b>RESET DB CFG</b>   |
| Description: Resets the parameters in the database manager configuration to the system defaults.<br>Authorization: sysadm   |
| <b>Windows/Linux/UNIX examples</b>  |
| Example:<br>reset dbm cfg   |
| <b>Notes</b>  |
| Before running this command, save the output from the GET DBM CFG command to a file so that you can refer to the existing settings. Individual settings can then be updated using the UPDATE DBM CFG command. |

|   |
|---|
| <b>RESET MONITOR</b>  |
| Description: Resets the internal database system monitor data areas of a specified database, or |

of all active databases, to zero. The internal database system monitor data areas include the data areas for all applications connected to the database, as well as the data areas for the database itself.

Authorization: sysadm/sysctrl/sysmaint

#### **Windows/Linux/UNIX examples**

Example 1:

Reset the internal counters for all databases in the instance.

```
reset monitor all
```

Example 2:

Reset the internal counters for only the SAMPLE database.

```
reset monitor for db sample
```

Example 3:

In a partitioned environment, reset the internal counters for all internal counters at database partition 1.

```
reset monitor all at dbpartitionnum 1
```

Example 4:

In a partitioned environment, reset the internal counters for the SAMPLE database at database partition 0.

```
reset monitor for db sample at dbpartitionnum 0
```

Example 5:

In a partitioned environment, reset the internal counter for all database on all partitions.

```
reset monitor all global
```

#### **Notes**

In a partitioned environment, if not partition is specified, the command applies to the current partition you are connected to.

### **RESTART DATABASE**

Description: Performs a crash recovery on a database that has been abnormally terminated and left in an inconsistent state.

Authorization: none

#### **Windows/Linux/UNIX examples**

Example 1:

Restart the SAMPLE database after it had been terminated abnormally by a power failure.

```
restart db sample
```

Example 2:

If one of the table spaces is inaccessible (e.g. damaged containers) during the restart database, the restart database command will fail. To make it successful, you need to use the `DROP PENDING TABLESPACES` option. After the database is successfully restarted, you must drop the table space specified by this option. (The `db2diag.log` will indicate which table space is inaccessible.)

```
restart db sample drop pending tablespaces (ts1)
```

Example 3:

If the database was in a `WRITE-SUSPEND` state when it got terminated abnormally, again, the restart database will fail. To correct this problem, use the `WRITE RESUME` option. This option can also be used in the case where the connection used to suspend I/O writes is currently hung and all subsequent connection attempts are also hanging. When used in this circumstance, `RESTART DATABASE` will resume I/O writes to the database without performing crash recovery. `RESTART DATABASE` with the `WRITE RESUME` option will only perform crash recovery when you use it after a database crash.

```
restart db sample write resume
```

**Notes**

If the `AUTORESTART` database configuration parameter is set to `ON`, the database will be auto-restarted during the first connect; the `RESTART DATABASE` command is not needed. It is needed if `AUTORESTART` is set to `OFF`.

**RESTORE DATABASE**

Description: Rebuilds a damaged or corrupted database that has been backed up using the DB2 backup utility. The restored database is in the same state it was in when the backup copy was made. This utility can also overwrite a database with a different image, or restore to a new database.

The restore utility can also be used to restore backup images that were produced on a previous version of DB2 (up to two versions earlier) as long as the word size (32-bit or 64-bit) is the same. Cross-platform restore operations from a backup image created with a previous version of DB2 are not supported. If a migration is required, it will be invoked automatically at the end of the restore operation.

This utility can also restore a table space level backup.

Authorization: `sysadm/sysctrl/sysmaint`

**Windows/Linux/UNIX examples**

Example 1:

On UNIX, under the `/dev3/backup` directory, there are two backup images of the `SAMPLE` database:

```
SAMPLE.0.db2inst1.NODE0000.CATN0000.20040331234149.001
```

```
SAMPLE.0.db2inst1.NODE0001.CATN0000.20040331234427.001
```

To restore the earlier copy, issue:

```
Restore db sample from /dev3/backup taken at 20040331234149
```

Example 2:

On Windows, the backup image is stored in a hierarchy of directories. For example, you have the following backup image for the `SAMPLE` database:

```
D:\backups\SAMPLE.0\DB2\NODE0000\CATN0000\20040521\122242.001, to restore this backup image, issue:
```

```
Restore db sample from D:\backups
```

**Notes**

See an example of a redirected restore in the `SET TABLESPACE CONTAINERS` command.

**REWIND TAPE (Windows only)**

Description: When running on Windows NT-based operating systems, DB2 supports backup and restore operations to streaming tape devices. Use this command for tape rewinding.

Authorization: none

**Windows/Linux/UNIX examples**

Example:

```
rewind tape on \\.\TAPE0
```

**ROLLFORWARD DATABASE**

Description: Recovers a database by applying transactions recorded in the database log files. Invoked after a database or a table space backup image has been restored, or if any table spaces have been taken offline by the database due to a media error. The database must be recoverable before the database can be recovered with rollforward recovery.

Authorization: `sysadm/sysctrl/sysmaint`

**Windows/Linux/UNIX examples**

Example 1:

Rollforward the SAMPLE database to the end of logs and complete.

```
rollforward db sample to end of logs  
rollforward db sample complete
```

The above commands can be combined in one:

```
rollforward db sample to end of logs and complete
```

Example 2:

Rollforward the SAMPLE database to a point in time, using OVERFLOW LOG PATH to specify the directory where the user exit saves archived logs.

```
rollforward db sample to 2004-04-03-14.21.56.245378 and complete  
overflow log path (/logs)
```

Example 3:

After three table spaces have been restored, roll one forward to the end of the logs, and the other two to a point in time, both to be done online:

```
rollforward db sample to end of logs tablespace(TBS1) online
```

```
rollforward db sample to 2004-04-03-14.21.56.245378 and stop  
tablespace(TBS2, TBS3) online
```

#### Notes

In a partitioned database environment, this command can only be invoked from the catalog partition. A database or table space rollforward operation to a specified point in time affects all partitions that are listed in the db2nodes.cfg file. A database or table space rollforward operation to the end of logs affects the partitions that are specified. If no partitions are specified, it affects all partitions that are listed in the db2nodes.cfg file; if rollforward recovery is not needed on a particular partition, that partition is ignored.

### RUNCMD

Description: Executes a specified command from the CLP interactive node command history.

Authorization: none

#### Windows/Linux/UNIX examples

Example:

The following example first displays the command history, then re-runs the second command.

```
db2 => history  
1 connect to sample  
2 select * from sales  
3 select * from department  
4 history  
db2 => runcmd 2
```

#### Notes

This command can only be run within CLP interactive mode. Specifically, it cannot be run from the CLP command mode or the CLP batch mode.

### RUNSTATS

Description: Updates statistics about the physical characteristics of a table and the associated indexes. These characteristics include number of records, number of pages, and average record length. The optimizer uses these statistics when determining access paths to the data. This utility should be called when a table has had many updates, or after reorganizing a table.

Authorization: sysadm/sysctrl/sysmaint/dbadm/CONTROL privilege on the table/LOAD authority

#### Windows/Linux/UNIX examples

Example 1:

Collect statistics on the table only, on all columns without distribution statistics:

```
runstats on table db2user.employee
```

**Example 2:**

Collect statistics on the table only, on columns empid and empname with distribution statistics:  
runstats on table db2user.employee with distribution on columns  
(empid, empname)

**Example 3:**

Collect statistics on the table only, on all columns with distribution statistics using a specified number of frequency limit for the table while picking the num\_quantiles from the configuration setting.

```
runstats on table db2user.employee with distribution default  
num_freqvalues 50
```

**Example 4:**

Collect statistics on a set of indexes.

```
runstats on table db2user.employee for indexes db2user.empl1,  
db2user.empl2
```

**Example 5:**

Collect basic statistics on all indexes only.

```
runstats on table db2user.employee for indexes all
```

**Example 6:**

Collect basic statistics on the table and all indexes using sampling for the detailed index statistics collection.

```
runstats on table db2user.employee and sampled detailed indexes all
```

**Notes**

In partitioned database environment, this command collects statistics for a table on the database partition from which it is invoked. If the table does not exist on that database partition, the first database partition in the database partition group is selected.

**SET CLIENT**

Description: Specifies connection settings for the back-end process.

Authorization: none

**Windows/Linux/UNIX examples**

**Example 1:**

In a partitioned database environment, connect to partition #3 (as specified in db2nodes.cfg). The SET CLIENT command overrides the value of the DB2NODE environment variable.

```
set client connect_partitionnum 3
```

**Example 2:**

Set the connect type to 2 and specify that a Transaction Manager (TM) is required to coordinate two-phase commit. A type 2 connect means you can connect to more than one databases within a unit of work. (The default is that you can only connect to one database within a UOW.) The following example first connects to database TEST1, inserts a row, then connects to database TEST2 and inserts a row. The commit in the end commits both inserts.

```
set client connect 2 syncpoint twophase  
update command options using c off  
connect to test1  
insert into testjp1 (col1, col2) values (1002, 'michel')  
connect to test2  
insert into testjp2 (col1, col2) values (1002, 'michel')  
commit  
terminate
```

**Example 3:**

```
Change the SQLRULES to db2.  
set client sqlrules db2
```

### SET RUNTIME DEGREE

Description: Sets the maximum run time degree of intra-partition parallelism for SQL statements for specified active applications. The value specified using this command can override the value that was determined at SQL statement compilation time. The actual run time degree will be the lowest of:

- The *max\_querydegree* configuration parameter
- The application run time degree
- The sql statement compilation degree.

Authorization: sysadm/sysctrl

#### Windows/Linux/UNIX examples

Example:

Set the run time degree for applications with application handles 1 and 34 to 4.  
set runtime degree for (1,34) to 4

#### Notes

Use the LIST APPLICATIONS command to display the active applications.

### SET TABLESPACE CONTAINERS

Description: Used during a redirected restore process, it permits the addition, change, or removal of table space containers for a database that is being restored.

Authorization: sysadm/sysctrl

#### Windows/Linux/UNIX examples

Example 1:

Restore the SAMPLE database to a different machine (with the same type of OS). Here are the steps:

1. On the source server, get the information on all table space containers. For example, get a list of table spaces using "db2 list tablespaces". Then for each tablespace ID, get the container information using "db2 list tablespace containers for <tablespace ID>".
2. On the target server, perform the restore operation. For each table space container you would like to change, use the SET TABLESPACE CONTAINERS command. Otherwise, the containers you got from step #1 are going to be used. If this is what you want, then you need to make sure the container paths exist on the target machine. The following example makes change to only one container, for table space with an ID of 1.

```
restore db sample replace existing redirect  
set tablespace containers for 1 using ('D:\sample\cont1')  
restore db sample continue
```

Example 2:

If the backup image was taken on-line, then the database is put in ROLLFORWARD pending state following the restore operation. In this case, the logs that occurred since the database was backup up need to be applied. This example specifies that any ALTER TABLESPACE operation issued against this table space since the database was backed up is to be ignored during a subsequent roll forward of the database.

```
set tablespace containers for 1 ignore rollforward container operations  
using ('D:\sample\cont1')
```

### SET TAPE POSITION (Windows only)

Description: When running on Windows NT-based operating systems, DB2 supports backup and restore operations to streaming tape devices. Use this command for tape positioning.

Authorization: none

| <b>Windows/Linux/UNIX examples</b>   |
|--|
| <p>Example:<br/>Set the tape device <a href="#">\\.\TAPE0 to position 1</a>.</p> <pre>set tape position on \\.\TAPE0 to 1</pre>  |
| <b>Notes</b>   |
| <p>The POSITION specifies the mark at which the tape is to be positioned. DB2 for Windows writes a tape mark after every backup image. A value of 1 specifies the first position, 2 specifies the second position, and so on. If the tape is positioned at tape mark 1, for example, archive 2 is positioned to be restored.</p> |

| <b>SET UTIL_IMPACT_PRIORITY</b>   |
|---|
| <p>Description: Changes the impact setting for a running utility. Using this command, you can:</p> <ul style="list-style-type: none"> <li>• Throttle a utility that was invoked in unthrottled mode</li> <li>• Unthrottle a throttled utility (disable throttling)</li> <li>• Reprioritize a throttled utility (useful if running multiple simultaneous throttled utilities)</li> </ul> <p>Authorization:</p>     |
| <b>Windows/Linux/UNIX examples</b>  |
| <p>Example 1:<br/>Unthrottle the utility with ID 3.</p> <pre>set util_impact_priority for 3 to 0</pre> <p>Example 2:<br/>Throttle the utility with ID 3 to 10.</p> <pre>set util_impact_priority for 3 to 10</pre>  |
| <b>Notes</b>  |
| <ol style="list-style-type: none"> <li>1. A value of 100 represents the highest priority and 1 represents the lowest priority. Setting priority to 0 will force a throttled utility to continue unthrottled. Setting priority to a non-zero value will force an unthrottled utility to continue in throttled mode.</li> <li>2. Use the LIST UTILITIES command to display the IDs of running utilities.</li> </ol> |

| <b>SET WRITE</b>  |
|---|
| <p>Description: Allows a user to suspend or resume I/O writes for a database. Typical use of this command is for splitting a mirrored database. This type of mirroring is achieved through a disk storage system.</p> <p>Authorization: sysadm/sysctrl/sysmaint</p> |
| <b>Windows/Linux/UNIX examples</b>  |
| <p>Example 1:<br/>To suspend I/O writes for the SAMPLE database.</p> <pre>connect to sample set write suspend for sample</pre> <p>Example 2:<br/>To resume I/O writes for the SAMPLE database.</p> <pre>connect to sample set write resume for sample</pre>         |
| <b>Notes</b>  |
| <p>A database connection is required prior to executing the SET WRITE command.</p>  |

| <b>TERMINATE</b>   |
|--|
| <p>Description: Explicitly terminates a database connection and the clp's back-end process.</p> <p>Authorization: none</p> |
| <b>Windows/Linux/UNIX examples</b>   |
| <p>Example:</p>  |

|   |
|---|
| <p>Terminate the current database connection.</p> <pre>terminate</pre>  |
| <b>Notes</b>  |
| <p>Although TERMINATE and CONNECT RESET both break the connection to a database, only TERMINATE results in termination of the back-end process.</p> |

| <b>UNCATALOG DATABASE</b>  |
|--|
| <p>Description: Deletes a database entry from the system database directory.</p> <p>Authorization: sysadm/sysctrl</p>  |
| <b>Windows/Linux/UNIX examples</b>   |
| <p>Example:</p> <p>Uncatalog the SAMPLE database.</p> <pre>uncatalog db sample terminate</pre>   |
| <b>Notes</b>   |
| <ol style="list-style-type: none"> <li>1. Use the LIST DCS DIRECTORY command to display the dcs databases.</li> <li>2. The TERMINATE command is needed to refresh the cache.</li> <li>3. If the SAMPLE database is a local database, the uncatalog database command does not drop the database. It simply removes it from the system database directory. To get the database back, run "db2 catalog db sample".</li> </ol> |

| <b>UNCATALOG DCS DATABASE</b>   |
|---|
| <p>Description: Deletes an entry from the DCS directory. The entry contains connection information to a database residing on z/OS, OS/390 or AS/400.</p> <p>Authorization: sysadm/sysctrl</p> |
| <b>Windows/Linux/UNIX examples</b>  |
| <p>Example:</p> <p>Uncatalog the dcs database called 'hostdb'.</p> <pre>uncatalog dcs db hostdb terminate</pre>   |
| <b>Notes</b>  |
| <ol style="list-style-type: none"> <li>1. Use the LIST DCS DIRECTORY command to display the dcs databases.</li> <li>2. The TERMINATE command is needed to refresh the cache.</li> </ol>       |

| <b>UNCATALOG LDAP DATABASE</b>  |
|---|
| <p>Description: Deregisters a database from LDAP.</p> <p>Authorization: none</p>  |
| <b>Windows/Linux/UNIX examples</b>  |
| <p>Example:</p> <p>Deregister the SAMPLE database from LDAP.</p> <pre>uncatalog ldap db sample</pre>  |
| <b>Notes</b>  |
| <p>When a database is dropped, the database object is removed from LDAP. The database is also automatically deregistered from LDAP when the database server that manages the database is deregistered from LDAP. It might, however, be necessary to manually uncatalog the database from LDAP if:</p> <ul style="list-style-type: none"> <li>• The database server does not support LDAP. The administrator must manually uncatalog each database from LDAP after the database is dropped.</li> <li>• During DROP DATABASE the database object cannot be removed from LDAP (because LDAP cannot be accessed). In this case, the database is still removed from the local machine, but the existing entry in LDAP is not deleted.</li> </ul> |

| <b>UNCATALOG LDAP NODE</b>  |
|---|
| Description: Deletes a node entry in LDAP.<br>Authorization: none                         |
| <b>Windows/Linux/UNIX examples</b>  |
| Example:<br>Uncatalog the LDAP node called 'ldapnode'.<br>uncatalog ldap node ldapnode    |
| <b>Notes</b>  |
| The LDAP node is automatically uncataloged when the DB2 server is deregistered from LDAP. |

| <b>UNCATALOG NODE</b>   |
|---|
| Description: Deletes a node entry from the node directory. Each node contains connection information to a host server.<br>Authorization: sysadm/sysctrl   |
| <b>Windows/Linux/UNIX examples</b>  |
| Example:<br>Deletes the TCP/IP node called 'node1' from the node directory. After this node is deleted, the host server defined in this node is no longer reachable from this system.<br>uncatalog tcpip node node1 |
| <b>Notes</b>  |
| Use the LIST NODE DIRECTORY command to display the nodes cataloged on the system.   |

| <b>UNCATALOG ODBC DATA SOURCE</b>  |
|--|
| Description: Deletes a user or system ODBC data source.<br>Authorization: none                   |
| <b>Windows/Linux/UNIX examples</b>   |
| Example:<br>Uncatalog the odbc data source sample.<br>uncatalog odbc data source sample          |
| <b>Notes</b>   |
| Use the LIST ODBC DATA SOURCES command to display the ODBC data sources cataloged on the system. |

| <b>UNQUIESCE</b>   |
|--|
| Description: When a database or an instance is quiesced (for maintenance or other reasons), no other users can access them. To restore user access, run the UNQUIESCE command.<br>Authorization: sysadm/dbadm (for database level unquiesce)<br>Sysadm/sysctrl (from instance level unquiesce) |
| <b>Windows/Linux/UNIX examples</b>   |
| Example 1:<br>Unquiesce the SAMPLE database.<br>connect to sample<br>unquiesce db  |
| Example 2:<br>Unquiesce the instance 'db2inst1'<br>unquiesce instance db2inst1   |
| <b>Notes</b>   |
| A database connection is required to unquiesce a database.   |

| <b>UPDATE ADMIN CFG</b>   |
|---|
| Description: Updates an admin cfg parameter for the DAS.<br>Authorization: dasadm |
| <b>Windows/Linux/UNIX examples</b>  |

Example:

Update the admin cfg parameter SMTP\_SERVER to specify the hostname of the email server. This email server is responsible for sending out email notifications when a health alert is triggered.

```
update admin cfg using smtp_server smtpserv
```

#### Notes

Some admin cfg parameters are configurable online, which means the DAS does not need to be restarted for the changes to take effect. Others need DAS be restarted.

### UPDATE ALERT CFG

Description: Updates the alter configuration settings for health indicators. The indicators are applicable to dbm, databases, containers, tablespaces, database, tablespace and container.

Authorization: sysadm/sysmaint/sysctrl

#### Windows/Linux/UNIX examples

Example 1:

Update a threshold-based health indicator configuration for the tsc.tscont\_util health indicator on the SAMPLE database. The update changes the warning threshold to 15, alarm threshold to 40, to enable actions, and to add two script actions.

```
update alert cfg for container
  c:\DB2\NODE0000\SQL00002\SQLT0002.0
  for 2 on sample
  using tsc.tscont_util
  set warning 15, alarm 40,
  thresholdschecked yes, actionsenabled yes
```

```
update alert cfg for container
  c:\db2\node0000\sql00002\sqlt0002.0
  for 2 on sample
  using tsc.tscont_util
  add action script c:\myscript1
    type os working directory c:\
    on warning user db2inst1 using db2inst1,
  script c:\myscript1
    type os working directory c:\
    on alarm user db2inst1 using db2inst1
```

### UPDATE CLI CFG

Description: Updates the contents of a specified section in the db2cli.ini file.

Authorization: sysadm

#### Windows/Linux/UNIX examples

Example 1:

Update the db2cli.ini file to add the EXPLAIN keyword into the section called 'tstcli2x'. We will use a setting of 3 for EXPLAIN to collect snapshot and explain data for SQLs statements submitted through CLI. (In order to collect the explain data, the explain tables must have been created by running the sqllib/misc/EXPLAIN.DDL script: db2 -tvf EXPLAIN.DDL)

```
update cli cfg for section tstcli2x using explain 3
```

Example 2:

If a keyword value is a string containing single quotation marks or imbedded blanks, the entire string must be delimited by double quotation marks.

```
update cli cfg for section tstclilx using TableType
"'TABLE', 'VIEW', 'SYSTEM TABLE'"
```

#### Notes

1. Use the GET CLI CFG command to display the contents of the db2cli.ini file.
2. You can update the contents of the db2cli.ini file directly using a text editor. The effect is the

- same as modifying it using the UPDATE CLI CFG command.
- The section name and the keywords specified on this command are not case sensitive. However, the keyword values *are* case sensitive.

#### UPDATE COMMAND OPTIONS

Description: Sets a command option during an interactive session, or from a batch input file. The settings only apply to the interactive session or the input file. The options are:

- a: Display SQLCA
- c: Auto-commit SQL statements
- e: Display SQLCODE/SQLSTATE
- l: Log commands in a history file
- n: Remove new line character
- o: Display to standard output
- p: Display DB2 interactive prompt
- r: Save output report to a file
- s: Stop execution on command error
- v: Echo current command
- w: Show SQL statement warning messages
- z: Redirect all output to a file.

Authorization: none

#### Windows/Linux/UNIX examples

Example:

Turn off the auto-commit option. This means in order to commit or rollback a SQL statement, you must execute the `commit` or `rollback` statements explicitly.

```
update command options using c off
insert into t1 values (1)
commit
```

Same effect could be achieved by adding the option in the SQL statement. For example:

```
db2 +c "insert into t1 values (1)"
db2 commit
```

#### Notes

Use the `LIST COMMAND OPTIONS` command to list the current command option settings.

#### UPDATE CONTACT

Description: Updates the attributes of a contact name that is defined on the local system.

Authorization: none

#### Windows/Linux/UNIX examples

Example:

Update the email address of user1. Notification will be sent to user's new email address

[user1@ibm.com](mailto:user1@ibm.com).

```
update contact user1 using address user1@ibm.com
```

#### UPDATE CONTACTGROUP

Description: Adds or removes users from a contact group that is defined on the local system. A contact group is a list of users who should be notified by the DB2 Scheduler and the Health Monitor.

Authorization: none

#### Windows/Linux/UNIX examples

Example 1:

Add user 'user1' to contactgroup 'grp1'.

```
update contactgroup grp1 add contact user1
```

Example 2:

```
Removes group 'db2users' from the contact group grp1
update contactgroup grp1 drop group grp1
```

#### UPDATE DB CFG

Description: Updates a database configuration parameter.  
Authorization: sysadm/sysctrl/sysmaint

#### Windows/Linux/UNIX examples

Example 1:

Update the database 'sample' to increase the database heap 'dbheap' to 1000 pages. The change will be made immediately.

```
connect to sample
update db cfg for sample using dbheap 1000
```

Example 2:

Update the database 'sample' to increase the database heap 'dbheap' to 1000 pages. However, the change is delayed until the database is restarted again (by disconnecting all connections and re-connect.)

```
update db cfg for sample using dbheap 1000
```

Example 3:

In multi-partitioned database environment, the commands in example 1 and 2 apply to only the database partition where the commands were submitted from. To update the parameter on all database partitions, use the following command:

```
db2_all ";db2 update db cfg for sample using dbheap 1000"
```

#### Notes

- 1. Use "GET DB CFG FOR <database\_alias>" command to list all the available parameters.
- Most of time, the database needs to be restarted (by disconnecting all applications and re-connect, or by deactivating the database using the DEACTIVATE DATABASE command if the database was activated by ACTIVATE DATABASE command) to make the changes effective. However, some of the parameters are configurable online, which means a database restart is not needed. However, to make their changes take effect immediately, you must connect to the database first before updating the parameter.
- To see whether a parameter is configurable online, search for the parameter in the DB2 InfoCenter (<http://publib.boulder.ibm.com/infocenter/db2help/index.jsp>).

#### UPDATE DBM CFG

Description: Updates a database manager configuration (dbm cfg) parameter.  
Authorization: sysadm

#### Windows/Linux/UNIX examples

Example 1:

Update the diagnostic level notification level 'diaglevel' to 4 on instance db2inst1. The change will take effect immediately without having to restart the instance.

```
attach to db2inst1
update dbm cfg diaglevel 4
```

Example 2:

Update the diagnostic level notification level to 4 on instance db2inst1. The change is not going to be effective until the instance is restarted.

```
update dbm cfg diaglevel 4
```

#### Notes

- Use "GET DBM CFG" command to list all the available parameters.
- Most of time, you need to restart the instance to make the changes effective. However, some of the parameters are configurable online, which means an instance restart is not needed. However, to make their changes take effect immediately, you must attach to the

- instance first before updating the parameter.
- To see whether a parameter is configurable online, search for the parameter in the DB2 InfoCenter (<http://publib.boulder.ibm.com/infocenter/db2help/index.jsp>).

#### UPDATE HEALTH NOTIFICATION CONTACT LIST

Description: Updates the contact list for notification about health alerts issued by an instance.

Authorization: sysadm/sysctrl/sysmaint

##### Windows/Linux/UNIX examples

Example 1:

Add 'user1' to the contact list.

```
update notification list add contact user1
```

Example 2:

Add group 'grp1' to the contact list.

```
update notification list add group grp1
```

Example3:

Removes 'grp1' from the contact list.

```
update notification list drop group grp1
```

#### UPDATE HISTORY FILE

Description: Updates the location, device type, or comment in a history file entry.

Authorization: sysadm/sysctrl,sysmaint/dbadm

##### Windows/Linux/UNIX examples

Example:

Update the history file entry for a full database backup taken on April 13, 1997 at 10:00 a.m. The new location for the backup image is 'backup/dbbackup.1', the storage media is disk.

```
update history for 19970413100000001 with location /backup/dbbackup.1
device type d
```

#### UPDATE LDAP NODE

Description: Updates connection information in a LDAP node.

Authorization: none

##### Windows/Linux/UNIX examples

Example:

The host name of the LDAP server has been changed to 'LDAPSERV', update the LDAP node 'LDAPNODE' to reflect this change.

```
update ldap node ldapnode hostname ldapserv
```

#### UPDATE MONITOR SWITCHES

Description: Turns monitor switches ON or OFF. The monitor switches are BUFFERPOOL, LOCK, SORT, STATEMENT, TABLE, TIMESTAMP and UOW.

Authorization: sysadm/sysctrl/sysmaint

##### Windows/Linux/UNIX examples

Example 1:

Turn on the STATEMENT monitor switch to collect information about dynamic SQLs.

```
update monitor switches using statement on
```

Example 2:

In a multi-partitioned database environment, turn on the LOCK monitor switches for partition 2.

```
update monitor switches using lock on at dbpartitionnum 2
```

Example 3:

In a multi-partitioned database environment, turn on the SORT monitor switch on all partitions.

update monitor switches using sort global