

## DB2 Version 8 Instances and Databases

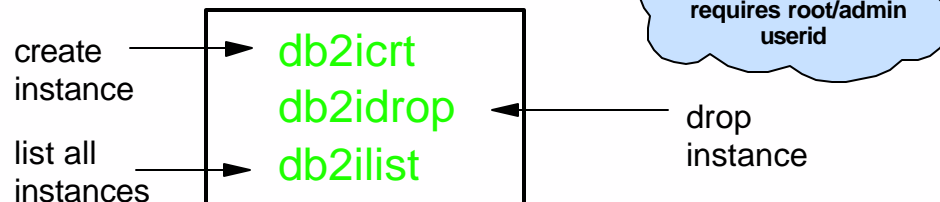
DB2 Quickstart Education

Maintained by Paul Yip (ypaul@ca.ibm.com)

February 2003

IBM Software Group

### Working with Instances



- An default instance is created during install
  - ▶ db2inst1 (UNIX)
  - ▶ DB2 (Windows)
- To create additional instances, you need:
  - ▶ A user ID to run the set of instance processes
  - ▶ UNIX only: user to run "fenced" processes (db2fenc1)

## Additional notes...

DB21085I Instance "db2inst1" uses "32" bits and DB2 code release "SQL08010" with level identifier "01010106". Informational tokens are "DB2 v8.1.1.0", "s021023", "", and FixPak "0". **Product is installed at "/usr/opt/db2\_08\_01".**

### ■ UNIX commands

```
<install_path>/instance/db2icrt -u <fenced userid> <instance name>  
<install_path>/instance/db2idrop <instance name>
```

### ■ Windows commands (commands are in system path)

```
db2icrt <instance name>  
db2idrop <instance name>
```

### ■ dropping an instance will not drop its associated databases(!)

## Switching instances on the same machine

### ■ UNIX:

- ▶ Each instance provides a profile script:
  - ─ Any local user that wants to access the instance simply includes this file in their \$HOME/.profile file or source it as needed.
  - ─ example (note the space after period):

```
$ . /home/db2inst1/sqllib/db2profile
```

### ■ Windows

- ▶ The environment variable DB2INSTANCE defines user's context
  - ─ example:

```
C:\>set DB2INSTANCE=DEVINST
```

## Working with Instances

start instance →

```
db2start
db2stop [force]
```

← stop instance

- you can stop and start instance independently by using the above commands as the instance owner
- note:  
FORCE APPLICATION ALL + db2stop is "nicer" than db2stop force

## Working with Databases

```
CREATE DATABASE MYDB
CREATE DATABASE MYDB on /databases/mydb
CREATE DATABASE MYDB
  AUTOCONFIGURE APPLY DB AND DBM
```

- Requirements:
  - ▶ Instance created and started
- Also Available in DB2 Control Center: CREATE DATABASE WIZARD
- Default CREATE DATABASE ... AUTOCONFIGURE automatically configures database to best use 25% of system resources

- AUTOCONFIGURE can be run after database has been created

**Note:** default behavior of AUTOCONFIGURE with CREATE DATABASE is different from standard AUTOCONFIGURE defaults

Keyword	Valid Values [default]	Explanation
mem_percent	1-100 [80]	Percentage of memory to dedicate. If other applications (other than the operating system) are running on this server, set this to less than 100.
workload_type	simple, mixed, complex [mixed]	Simple workloads tend to be I/O intensive and mostly transactions, whereas complex workloads tend to be CPU intensive and mostly queries.
num_stmts	1-1000000 [10]	Number of statements per unit of work
tpm	1-50000 [60]	Transactions per minute
admin_priority	performance, recovery, both [both]	Optimize for better performance (more transactions per minute) or better recovery time
is_populated	yes, no [yes]	Is the database populated with data?
num_local_apps	0-5000 [0]	Number of connected local applications
num_remote_apps	0-5000 [10]	Number of connected remote applications
isolation	RR, RS, CS, UR [RR]	Isolation level of applications connecting to this database (Repeatable Read, Read Stability, Cursor Stability, Uncommitted Read)
bp_resizeable	yes, no [yes]	Are buffer pools resizeable?

## LIST DATABASE DIRECTORY LIST DB DIRECTORY

```
DB2 CLP
Number of entries in the directory = 2
Database 1 entry:
Database alias          = PWIP
Database name           = PWIP
Node name               = TCP89EE9
Database release level  = a.00
Comment                 =
Directory entry type    = Remote
Catalog database partition number = -1
Database 2 entry:
Database alias          = TWOPART
Database name           = TWOPART
Database drive          = C:\DB2
Database release level  = a.00
Comment                 =
Directory entry type    = Indirect
Catalog database partition number = 0
C:\>
```

## Dropping Databases

### DROP DB MYDB

- **Cool Trick:**
  - ▶ You can drop an instance, but the database files will still be there. If you recreate an instance, to recover the database, just "re-catalog" it.
  - ▶ **Tip:** to expose where previously created database files exist:
    - LIST DB DIRECTORY ON C:\

## Connecting To Databases

- **CONNECT TO <dbname>**
  - ▶ Assumes current userid and password as logged into the system
- **CONNECT TO <dbname> USER <user>**
  - ▶ Will prompt for password
- **CONNECT TO <dbname> USER <user> USING <password>**
  - ▶ Connect with all information provided
- **GET CONNECTION STATE**
  - ▶ gets current connection status
- **CONNECT RESET or TERMINATE**
  - ▶ disconnects from current database

## DB2 UDB System Catalog

- **Each database has own system catalog tables/views.**
  - ▶ Meta Data about the database objects
  - ▶ Object level user privileges security
  - ▶ all reside in SYSIBM, SYSSTAT, or SYSCAT schema
- **Cannot be user created or dropped, but can be queried directly or through the system catalog views.**
  - ▶ Some values in the system catalog views can be updated. They are grouped in SYSSTAT schema.

## db2start...db2stop...connect...activate...deactivate

- on DB2START
  - ▶ instance processes started
- on ACTIVATE
  - ▶ allocates database global memory explicitly
  - ▶ prevents:
    - deallocation of resources when last connection disconnects
    - first connection from "first connect cost"
- on CONNECT
  - ▶ database global memory allocated if not already activated
  - ▶ process allocated/dispatched to service connection
- on DEACTIVATE
  - ▶ explicitly deallocate database global memory
- on DB2STOP
  - ▶ instance processes stopped

## CONNECT RESET ... TERMINATE

- CONNECT RESET
  - ▶ disconnects current connection, but not necessarily db2bp on server
- TERMINATE
  - ▶ kills the background processes for that session
  - ▶ db2bp may exist even if you did not connect

## Other Useful Commands

---

- LIST APPLICATIONS [SHOW DETAIL]
  - ▶ show all connections
- FORCE APPLICATION [ ALL | ( $h_1[,h_2,...h_n]$ ) ]
  - ▶ disconnect all connections or selectively by handle #
- LIST TABLES [ FOR ALL | FOR SCHEMA <schema> ]
  - ▶ list tables for the current user, all tables, or by schema
- DESCRIBE TABLE <tablename>
  - ▶ show the structure of <tablename>
- GET INSTANCE
  - ▶ retrieve your current instance context