DB2 10.1 DBA for Linux, UNIX, and Windows certification exam 611 prep, Part 1: Server management

DB2 certification exam 611 server management

Kshitij Kohli
Amyris Rada

December 05, 2013

This tutorial helps you learn the skills required to manage DB2 database servers, instances and databases. Furthermore, you will get introduced to DB2 autonomic computing capabilities and you will learn to use IBM data Studio to perform database administration tasks such as job scheduling and generating diagrams of access plans. This tutorial prepares you for Part 1 of the DB2 10.1 DBA for Linux, UNIX, and Windows certification exam 611.

View more content in this series

Before you start

Learn what to expect from this tutorial, and how to get the most out of it.

About this series

If you are preparing to take the DB2 10.1 for Linux, UNIX, and Windows DBA certification exam (exam 611), you have come to the right place. The "DB2 10.1 for Linux, UNIX, and Windows DBA certification exam 611" series consists of 8 tutorials that cover all the major concepts and topics that you must know for the exam. Even if you are not planning to seek certification right away, the information presented in this set of tutorials helps you learn about many of the new features and capabilities available in DB2 10.1 for Linux, UNIX, and Windows.

About this tutorial

Ten percent (10%) of the DB2 10.1 for Linux, UNIX, and Windows DBA certification exam 611 is designed to test your ability on Server Management. This tutorial covers DB2 Server management topics by providing information about how to configure and manage DB2 data servers, instances, and databases. It will also cover how to use autonomic features, schedule jobs and use Visual
Explain with IBM data Studio. For a more in depth knowledge about server management, see the Related topics section.

Objectives

After completing the tutorial, you should have a better understanding of the following topics:

- Demonstrate the ability to configure and manage DB2 data servers, instances, and databases.
- Demonstrate the ability to use autonomic computing capabilities.
- Demonstrate the ability to use IBM Data Studio.
  - Demonstrate the ability to schedule jobs.
  - Demonstrate the ability to use visual explain.

You can also find these objectives listed on IBM certification web site under the Section 1 – DB2 Server Management in the exam objectives.

Approximately 10% of the exam covers the topics listed in this section.

Prerequisites

Understanding the content in this tutorial requires knowledge of the following topics:

- DB2 data server, instances, databases
- DB2 database backup and restore
- REORG and RUNSTATS operations
- Self-tuning memory manager (STMM)
- Data compression
- Configuration Advisor
- DB2 Explain and Design Advisor
- IBM Data Studio

System requirements

To run the examples and commands presented in this tutorial, you need access to a DB2 10.1 database. If you do not have such access, you can download a trial copy of DB2 10.1 and a complimentary download of IBM data Studio for personal use.

Configuring and managing DB2 servers, instances, and databases

DB2 data servers

A DB2 data server refers to a computer running a supported operating systems where DB2 for Linux, UNIX, and Windows is installed. The DB2 data server is an RDBMS where data gets stored in pages and containers. It also includes optimized SQL support based on actual database usage and tools to help manage the data.
A DB2 data server has registry and environment variables, one or more instances, one or more databases within an instance. This section goes over each of these components shown in the following figure.

**Figure 1. DB2 data server**

![Figure 1. DB2 data server](image)

**DB2 instances**

A DB2 instance provides an independent environment where database objects can be created and applications can run. Several instances can be created on one server, and each instance can have a multiple number of databases. Because of these independent environments, one instance cannot “see” the contents of another instance. Therefore, objects of the same name can exist in two or more instances. Each instance configuration and maintenance can be set up differently based on the usage and service level agreements.

**Creating instances**

Creating instances has different steps depending on the operating system.

**Creating instances on Windows operating systems**

To create an instance, issue the `db2icrt` command from a DB2 command window as follows:

```
db2icrt instance_name
```

This command creates a subdirectory under the SQLLIB directory with the name of the instance just created. In addition, a Windows service called `DB2 – db2copy_name – instance_name` is created. DB2 always precedes any name of related Windows services.

**Creating instances on Linux and UNIX operating systems**

To create an instance, issue the `db2icrt` command as follows:

```
cd $DB2DIR/instance/
db2icrt -u fenced_user_ID instance_name
```

You must have root authority or have the system administrator run the `db2icrt` command for you.
The `db2icrt` command creates the subdirectory SQLLIB under the home directory of the user instance name.

DB2 on Linux and UNIX also requires a fenced user (fenc_name) to run user-defined routines such as stored procedures and user-defined functions (UDFs) in a separate address space than the one used by the DB2 database manager. This ensures that problems with these objects do not affect your database or instance. This protects the database manager from being accidentally or maliciously damaged by any user-defined routines.

**Dropping instances**

To drop an instance, issue the `db2idrop` command as follows:

```bash
cd $DB2DIR/instance/
db2idrop instance_name
```

Make sure to stop all the databases and back them up before you invoke this command.

**Listing, updating, and upgrading instances**

You can list all the instances running on your server by running the following DB2 command:

```bash
db2ilist
```

When you install fix packs or patches to the DB2 data server, you must update the existing instances to have them run on the new fix pack release or patch. Use the following DB2 command to update an instance:

```bash
db2iupdt instance_name
```

Use the `db2ckupgrade` and `db2iupgrade` commands to verify that your instance is ready to upgrade from one release to a new release and upgrade your instance to the new release. The `db2ckmigr` and `db2imigr` command are discontinued.

**Attaching to and detaching from instances**

Before you can issue certain DB2 commands, you must be attached to an instance. To attach to an instance, issue the following DB2 command:

```bash
db2 attach to instance_name
```

To detach from an instance, issue the `db2idrop` command as follows:

```bash
db2 detach
```

Additionally, you can issue the `db2start` command to start an instance and the `db2stop` command to stop an instance.

To run any of the DB2 commands that manage instances, you require special authorization such as instance owner. For more details, see the DB2 command reference.
Setting up a DB2 Environment

Proper setup of the DB2 environment is very important because it controls how a DB2 data server operates and functions. The DB2 environment is made up of:

- DB2 profile registries
- Operating system environment variables
- DB2 database manager configuration parameters
- DB2 database configuration parameters

Figure 2. DB2 data server environment

Setting registry variables

A DB2 profile registry consists of registry variables that affect the management, configuration, and performance of the DB2 data server. You usually need to stop and restart an instance so that the changes to the registry variables take effect.

To list all the supported registry variables, issue the following command:

```
db2set -lr
```

To set a registry variable, issue the following command:

```
db2set registry_variable=value
```

Note that there are no spaces between the variable name, the equals sign, and the variable value.

The following example sets the DB2COMM registry variable to a single value:

```
db2set DB2COMM=TCPIP
```

The following example shows how to set the DB2COMM registry variable to multiple values:

```
db2set DB2COMM=TCP/IP,NPIPE,LOCAL
```
To reset a registry variable to its default value, simply issue the `db2set` command without specifying a value after the `=` sign as follows:

```
db2set registry_variable=
```

To display all the registry variables currently set on the server, issue the following command:

```
db2set -all
```

The following text is an example of the output that you can get when you issue this command:

```
[e] DB2PATH=C:\Program Files\IBM\SQLLIB_01
[i] DB2ACCOUNTNAME=IBM-TP101\kkohli
[i] DB2INSTOWNER=IBM-SB2QTSR5RSN
[i] DB2PORTRANGE=60001:60004
[i] DB2INSTPROF=C:\Program Files\IBM\SQLLIB_01
[i] DB2COMM=TCPIP,NPIPE,LOCAL
[g] DB2_EXTSECURITY=YES
[g] DB2SYSTEM=IBM-TP101
[g] DB2INSTDEF=DB2V
```

Indicators surrounded by the square brackets ([ ]) represent the scope of the registry variable, as follows:

- `[e]` represents a registry variable setting for the current session or environment
- `[u]` represents a user-level registry variable setting
- `[n]` represents a node-level registry variable setting
- `[i]` represents an instance-level registry variable setting
- `[g]` represents a global-level registry variable setting

### Setting system environment variables

Most of the DB2 environment settings are controlled by registry variables in the DB2 profile registry. Those variables that are not stored in the profile registry are called the operating system environment variables. The commands for setting the system variables will vary depending on the operating systems and the UNIX shells where the DB2 data server is installed. Here are some examples of setting the `DB2INSTANCE` variable to the `PROD` instance name:

- For Windows operating systems, the command is `set DB2INSTANCE=PROD`.
- For the Korn shell in Linux and UNIX operating systems, the command is `export DB2INSTANCE=PROD`.

`DB2INSTANCE` is an important system variable that you should be aware of. It specifies the default DB2 instance or the instance for the current application sessions. Once this variable is set, all subsequent DB2 commands are executed within the scope of that instance.

To find out which DB2 instance you are working with, issue the `GET INSTANCE` command. The following text shows an example output for this command:

```
The current database manager instance is: DB2V
```
Attaching to and detaching from instances

Setting database manager and database configuration parameters

You can set configuration parameters at the database manager (instance) level and at the database level. At the database manager level, you can configure the entire DB2 environment for an instance. These settings impact all the databases in that instance and all applications using databases in that instance. Configuring parameters at the database level affects the behavior of all applications accessing that particular database. You can use IBM Data Studio or the DB2 command line processor (CLP) to set the database manager (DBM) and database (DB) configuration parameters and to display their current settings.

Using IBM Data Studio to display configuration parameter settings

To view DBM configuration parameters, open the Administration Explorer and right-click on the instance name. Then select Configure as shown in Figure 3:

Figure 3. Displaying DBM configuration parameters in the Administration Explorer

To view DB configuration parameters, open the Administration Explorer, right-click on the database name, and select Set Up and Configure. Then select Configure as shown in the following figure:
Using the DB2 CLP to display configuration parameter settings

To view DBM and DB CFG configuration parameters, issue the following commands:

```
  db2 get dbm cfg
  db2 get db cfg for database_name
```

For windows operating systems, you must use a DB2 command window to issue DB2 CLP commands.

Updating DBM and DB configuration parameter settings with IBM Data Studio

To update the values of the DBM or DB configuration parameters, follow the steps shown in Using IBM Data Studio to display configuration parameter settings to run the Configure action. Then find the parameter you want to change and enter a new value in the Pending Value column. Figure 5 shows how to modify a database configuration parameter setting:
Figure 5. Updating DB configuration parameters in the Administration Explorer

Click on Preview Command to verify the DB2 command that will be issued. Then click on the Run button to issue the command. Hovering over the configuration parameter with the mouse displays what the parameter is and what is it used for. Some configuration parameters cannot change dynamically and show different values in Value (current) and Pending Value columns. The pending value is the new value that will be set the next time the instance or database is stopped and restarted. The Pending Value column also shows when the new value takes effect.

Updating DBM and DB configuration parameter settings with the DB2 CLP

To update DBM and DB CFG configuration parameters, issue the following commands:

```
db2 update dbm cfg using parameter_name new_value
db2 update db cfg for database_name using parameter_name new_value
```

You can set most configuration parameters online while an instance or a database is still running. By default, changes to these online configuration parameters take effect immediately where possible. For example, if you update the value of sortheap database configuration parameter, all new SQL statements use the new value. To explicitly specify that the change takes effect immediately, append the IMMEDIATE keyword to the UPDATE DBM or UPDATE DB command:

```
db2 update dbm cfg using parameter_name new_value immediate
db2 update db cfg for database_name using parameter_name new_value immediate
```
If you choose to defer the changes until the instance is restarted or until the database is activated, specify the `DEFERRED` keyword instead as shown in the following example:

```
db2 update dbm cfg using parameter_name new_value deferred
db2 update db cfg for database_name using parameter_name new_value deferred
```

In IBM Data Studio, clear the box in the immediate column and verify with `Preview Command` that the `DEFERRED` keyword is added to the command. See Figure 5.

To list current and deferred (delayed) values of the `dbm` configuration parameters, issue the following commands:

```
db2 attach to instance_name
`db2 get dbm cfg show detail`
```

Figure 6 shows an example that updates the `numdb` database manager configuration parameter and uses the `db2 get dbm cfg show detail` command to check the current value and the deferred value of the configuration parameter in the command output:

**Figure 6. List of DBM configuration parameters with details**

To list the current and deferred values of the database configuration parameters, issue the following commands:

```
db2 connect to database_name
`db2 get db cfg for database_name show detail`
```
Figure 7 shows an example that updates the `dft_queryopt` database configuration parameter and uses the `db cfg for database_name show detail` command to check the current value and the deferred value of the parameter in the command output:

**Figure 7. List of DB configuration parameters with details**

If the DBM or DB configuration parameter cannot be configured online, the change to the DBM or DB configuration parameter cannot take effect immediately and the following warning message is returned:

```
SQL1362W One or more of the parameters submitted for immediate modification were not changed dynamically. Client changes will not be effective until the next time the application is started or the TERMINATE command has been issued. Server changes will not be effective until the next DB2START command.
```

If you want the change to take effect immediately, then you must restart the instance or the database. If there are applications connected to the database, you must terminate all the connections to the database before you can restart the database or the instance by using the following DB2 commands:

```
db2 force applications all
db2stop force
db2start
```

To terminate database connections for specific applications, first determine their Application Id by issuing the following command:

```
db2 list applications
```
Then issue the `db2 force applications` command specifying the Application Id that you found in the previous step.

Listing 1 shows an example of the `db2 list applications` command and its output:

**Listing 1. List of applications connected to DB2 databases**

<table>
<thead>
<tr>
<th>Auth Id</th>
<th>Application Name</th>
<th>Appl. Handle</th>
<th>Application Id</th>
<th>DB Name</th>
<th># of Agents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2ADMIN</td>
<td>db2bp.exe</td>
<td>7688</td>
<td>*LOCAL.DB2.131122179320</td>
<td>SAMPLE</td>
<td>1</td>
</tr>
<tr>
<td>DB2ADMIN</td>
<td>db2jcc_applica</td>
<td>7628</td>
<td>9.26.41.88.51940.131122161832</td>
<td>SAMPLE</td>
<td>1</td>
</tr>
<tr>
<td>DB2ADMIN</td>
<td>db2jcc_applica</td>
<td>7627</td>
<td>9.26.41.88.51941.131122161831</td>
<td>SAMPLE</td>
<td>1</td>
</tr>
</tbody>
</table>

To terminate database connections for the last two applications shown in Listing 1, issue the following command:

`db2 force applications (12635, 12619)`

**DB2 autonomic computing capabilities**

The DB2 autonomic computing environment is self-configuring, self-healing, self-optimizing, and self-protecting. By automatically sensing and responding to situations, these autonomic computing capabilities reduce considerable the overhead of managing a DB2 environment. Some of the following capabilities are enabled by default but others you must enable manually.

- Self-tuning memory manager (STMM)
- Automatic storage
- Data compression
- Automatic maintenance
  - Automatic database backup
  - Automatic reorganizations
  - Automatic statistics collection
- Configuration advisor
- Utility throttling

**Self-tuning memory manager (STMM)**

A memory-tuning capability that simplifies the task of memory configuration by automatically setting values for several memory configuration parameters. When enabled, the memory tuner dynamically distributes available memory resources among the memory consumers such as buffer pools, locking memory, package cache, and sort memory. Best of all, STMM does not require any DBA intervention in tuning the memory parameters based on workload change.

You can enable memory areas or consumers by using the following memory-related database configuration parameters that are automatically tuned:

- `database_memory` - Database shared memory size
To enable buffer pools as a memory area, specify size using the AUTOMATIC value with the ALTER BUFFERPOOL or the CREATE BUFFERPOOL statement.

**Enabling STMM**

You can enable STMM through the `self_tuning_mem` database configuration parameter. Then you enable the self tuning of several memory areas.

First, set the `self_tuning_mem` database configuration parameter to ON as follows:

```
db2 "connect to KKOHLI"
db2 "update db cfg for KKOHLI using self_tuning_mem ON"
```

Alternatively, change the Pending Value in Modifying Database Configuration database parameters window of the Administration Explorer of IBM Data studio as shown in Figure 8.

**Figure 8. Setting the `self_tuning_mem` database configuration parameter**

- locklist - Maximum storage for lock list
- maxlocks - Maximum percent of lock list before escalation
- pckcachesz - Package cache size
- sheapthres_shr - Sort heap threshold for shared sorts
- sortheap - Sort heap size
Click on Preview Command to verify the DB2 command that will be issued. Then click on the Run button to issue the command.

Enable memory areas for STMM that are controlled by memory configuration parameters by setting the relevant configuration parameters to AUTOMATIC as follows:

```sql
db2 "connect to KKOHLI"
db2 "update db cfg for KKOHLI using PCKCACHESZ AUTOMATIC"
db2 "update db cfg for KKOHLI using LOCKLIST AUTOMATIC"
db2 "update db cfg for KKOHLI using MAXLOCKS AUTOMATIC"
db2 "update db cfg for KKOHLI using SORTHEAP AUTOMATIC"
db2 "update db cfg for KKOHLI using SHEAPTHRES_SHR AUTOMATIC"
db2 "update db cfg for KKOHLI using DATABASE_MEMORY AUTOMATIC"
```

Enable self-tuning of memory for buffer pools by creating or modifying existing buffer pools as follows:

- Create buffer pools specifying size AUTOMATIC as shown in the following example:
  ```sql
db2 "connect to KKOHLI"
db2 "create bufferpool bpool8k size AUTOMATIC pagesize 8 k"
  ```
- Modify existing buffer pools from AUTOMATIC to a specific size as shown in the following example:
  ```sql
db2 "connect to KKOHLI"
db2 "alter bufferpool bpool4k size AUTOMATIC"
  ```

Disabling STMM

To disable STMM, simply set the `self_tuning_mem` database configuration parameter to OFF as follows:

```sql
db2 "connect to KKOHLI"
db2 "update db cfg for KKOHLI using self_tuning_mem OFF"
```

Alternatively, enable the memory areas in the `Modifying Database Configuration database parameters` window of the Administration Explorer of IBM Data studio as shown in Figure 8.

Disabling memory areas in STMM

To disable memory areas for STMM that are controlled by memory configuration parameters, set the relevant configuration parameters to MANUAL as follows:

```sql
db2 "connect to KKOHLI"
db2 "update db cfg for KKOHLI using PCKCACHESZ MANUAL"
db2 "update db cfg for KKOHLI using LOCKLIST MANUAL"
db2 "update db cfg for KKOHLI using MAXLOCKS MANUAL"
db2 "update db cfg for KKOHLI using SORTHEAP MANUAL"
db2 "update db cfg for KKOHLI using SHEAPTHRES_SHR MANUAL"
db2 "update db cfg for KKOHLI using DATABASE_MEMORY MANUAL"
```

Alternatively, disable the memory areas in the `Modifying Database Configuration database parameters` window of the Administration Explorer of IBM Data studio as shown in Figure 8.

To disable self-tuning of memory for buffer pools:
• Create buffer pools specifying size and do not use the AUTOMATIC value as shown in the following example:
  ```
  db2 "connect to KKOHLI"
  db2 "create bufferpool bpool8k size 1000 pagesize 8 k"
  ```
• Modify existing buffer pools from AUTOMATIC to a specific size as shown in the following example:
  ```
  db2 "connect to KKOHLI"
  db2 "alter bufferpool bpool4k size 1000"
  ```

**Automatic storage**

Automatic storage is the second area covered under autonomic computing because it handles storage of databases for you. It also simplifies storage management for table spaces.

You can create storage groups consisting of paths on which the database manager places your data. Then, the database manager manages the container and space allocation for the table spaces as you create tables and populate them. You can specify the paths of the default storage group when creating the database.

By default, all databases are created with automatic storage and a default storage group. However, if the database is created specifying the AUTOMATIC STORAGE NO clause, you cannot use automatic storage table spaces. However, this clause is deprecated and might not be available in a future release.

To manage automatic storage in DB2 databases:

• Use the AUTOMATIC STORAGE YES clause to create databases with automatic storage enabled. The following example shows how to create such database:
  ```
  db2 CREATE DATABASE "kkohli" AUTOMATIC STORAGE YES ON '/data' DBPATH ON '/data' ;
  ```
• Use the ON clause to create a database with multiple storage paths. As the database grows, the database manager creates containers across those storage paths, and automatically extends them or creates new ones as needed. The following example shows how to create a database with three storage paths:
  ```
  db2 CREATE DATABASE "kkohli" AUTOMATIC STORAGE YES ON '/data', '/logs', '/bkp'
  DBPATH ON '/data' ;
  ```
• Use the CREATE STOGROUP command to add new storage groups or to enable automatic storage. The following example shows how to create storage groups where operational_sg is the name of the storage group and '/data', '/data', '/logs', '/bkp' are the storage paths to be added:
  ```
  db2 create stogroup operational_sg ON '/data', '/data', '/logs', '/bkp'
  ```
• Use the AUTOMATIC STORAGE NO clause to create databases without automatic storage enabled. The following example shows how to create such database:
  ```
  db2 CREATE DATABASE "kkohli" AUTOMATIC STORAGE NO ON '/data' ;
  ```
• Use the ADMIN_GET_STORAGE_PATHS administrative view to list the storage paths for a database. The following examples shows how to use this view to list all the storage paths for the currently connected database and results for the SAMPLE database:
SELECT VARCHAR(STORAGE_GROUP_NAME, 30) AS STOGROUP, VARCHAR(DB_STORAGE_PATH, 40) AS STORAGE_PATH FROM TABLE(ADMIN_GET_STORAGE_PATHS('',-1)) AS T

<table>
<thead>
<tr>
<th>STOGROUP</th>
<th>STORAGE_PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBMSTOGROUP</td>
<td>C:</td>
</tr>
</tbody>
</table>

### Automatic storage table spaces

Any table spaces that you create are managed as automatic storage table spaces unless you specify otherwise or the database was created using the AUTOMATIC STORAGE NO clause. With automatic storage table spaces, you are not required to provide container definitions. The database manager looks after creating and extending containers to make use of the storage allocated to the database.

The following examples show how to create automatic storage table spaces:

- CREATE TABLESPACE TS1
- CREATE TABLESPACE TS2 MANAGED BY AUTOMATIC STORAGE
- CREATE TEMPORARY TABLESPACE TEMPTS
- CREATE USER TEMPORARY TABLESPACE USRTMP MANAGED BY AUTOMATIC STORAGE
- CREATE LARGE TABLESPACE LONGT5
- CREATE TABLESPACE TS3 INITIALSIZE 8K INCREASESIZE 20 PERCENT MANAGED BY AUTOMATIC STORAGE
- CREATE TABLESPACE TS4 MAXSIZE 2G
- CREATE TABLESPACE TS5 USING STOGROUP SG_HOT

Each of these examples assumes that the table spaces are created in a database that has one or more storage groups defined. When you create a table space in a database that has no storage groups defined, you cannot use the MANAGED BY AUTOMATIC STORAGE clause. You must create a storage group, and then create automatic storage table spaces.

### Data compression

You can reduce storage needed for your data by using the compression capabilities built into DB2 for Linux, UNIX, and Windows to reduce the size of tables, indexes, and even your backup images. The types of compression available in DB2 are:

- Table compression
- Index compression
- Backup compression

#### Table compression

You can enable data row compression in DB2 databases when you create or modify tables by using the COMPRESS YES clause.

- The following example shows how to create a new table that is enabled for row compression:

  ```
  db2 CREATE TABLE Sales COMPRESS YES
  ```
• The following example shows how to enable an existing table for row compression:
  
  ```sql
  db2 ALTER TABLE Sales COMPRESS YES
  ```

  The compression takes effect only once the table dictionary is built which is usually during a table reorganization. The following example shows how to built the compression dictionary by using the `REORG TABLE` command:

  ```sql
  db2 REORG TABLE Sales RESETDICTIONARY
  ```

  However, the next time that you run the `REORG TABLE` command, you must add the `KEEPDICTIONARY` clause to keep compression dictionary. The following example shows how to use this clause:

  ```sql
  db2 REORG TABLE Sales KEEPDICTIONARY
  ```

  After the table reorganization is complete, your table can leverage the advantages of compression such as less I/O and less storage space.

  You can use the `INSPECT` command to estimate storage savings from row compression. The following example shows how to estimate these storage savings:

  ```sql
  db2 INSPECT ROWCOMPESTIMATE TABLE NAME Sales SCHEMA KK
  RESULTS KEEP inspect_out.txt
  db2inspf inspect_out.txt db2inspf_out.txt
  ```

  The `INSPECT` command analyzes the entire table and gives estimates of how much space will be saved. However, you must format the output of the `INSPECT` command into readable text by running the `db2inspf` command.

### Index compression

By default, index compression is enabled for compressed tables, and disabled for uncompressed tables. You can override this default behavior by using the `COMPRESS YES` clause of the `CREATE INDEX` statement as shown in the following example:

```sql
db2 CREATE INDEX IX_kohli ON sales (ACTUAL_TYPE, VERSION)
  ALLOW REVERSE SCANS PAGE SPLIT SYMMETRIC COLLECT SAMPLED DETAILED STATISTICS
  COMPRESS YES
```

You can also use the `COMPRESS YES` clause with the `ALTER INDEX` statement to enable index compression on an existing index as shown in the following example:

```sql
db2 ALTER INDEX DB2ADMIN.IDX_ABSTRCOMPLISTSPEC_KINDID COMPRESS YES
```

You can disable index compression by using the `COMPRESS NO` clause when you create or modify an index. After enabling or disabling index compression, you must perform an index reorganization to rebuild the index. The following restrictions apply to index compression:

• Index compression is not supported in block indexes and XML path indexes
• Index specifications cannot be compressed
• Compression attributes for indexes on temporary tables cannot be altered with the `ALTER INDEX` command.

Backup compression
You can use backup compression to reduce the size of your database backups. To compress your backups, use the `COMPRESS` clause in the `BACKUP DATABASE` command as shown in the following example:

```
db2 BACKUP DATABASE KKOHLI COMPRESS
```

Although you can use backup compression with tables that use row compression and index compression, use backup compression only on tables that do not use compression due to the additional resources and time it requires. Especially if these tables contain index objects that are separate from LOB and long field data.

Automatic maintenance
The database manager provides automatic maintenance capabilities for performing database backups, keeping statistics current, and reorganizing tables and indexes. These activities are necessary for improving performance and recoverability. You can use database configuration parameters to enable or disable a specific capability. The following text is an extract of the `GET DB CFG` command that shows a list of all database configuration parameters for automatic maintenance and their hierarchy:

```
Automatic maintenance                      (AUTO_MAINT) = ON                         ON
Automatic database backup            (AUTO_DB_BACKUP) = OFF                        OFF
Automatic table maintenance          (AUTO_TBL_MAINT) = ON                         ON
Automatic runstats                  (AUTO_RUNSTATS) = ON                         ON
Real-time statistics            (AUTO_STMT_STATS) = ON                         ON
Statistical views              (AUTO_STATS_VIEWS) = OFF                        OFF
Automatic sampling                (AUTO_SAMPLING) = OFF                        OFF
Automatic statistics profiling    (AUTO_STATS_PROF) = OFF                        OFF
Statistics profile updates        (AUTO_PROF_UPD) = OFF                        OFF
Automatic reorganization               (AUTO_REORG) = OFF                        OFF
```

The `AUTO_MAINT` configuration parameter is the highest in the hierarchy and the main switch. If you set it to `OFF`, all the automatic maintenance capabilities are disabled regardless of the value in any of the children configuration parameters. To enable a given automatic maintenance capability, the corresponding database configuration parameter and all higher level switches must be set to `ON`.

For example, to enable automatic table reorganization, the `auto_maint`, `auto_tbl_maint`, and `auto_reorg` database configuration parameters, you must set all these database configuration parameters to `ON` as shown in the following example using the DB2 CLP:

```
db2 CONNECT TO kkohli
   db2 UPDATE DB CONFIG USING AUTO_MAINT ON AUTO_TBL_MAINT ON AUTO_RUNSTATS ON
```

You can also use IBM Data Studio to configure automatic maintenance capabilities by connecting to the database and using the `Configure Automatic Maintenance` task assistant in the Administration Explorer as shown in the Figure 9:
Furthermore, you can set up online maintenance windows and offline maintenance windows to define the time when maintenance activities can be performed. Also, you can define backup, reorg, and runstats policies by using the DB2 CLP or IBM Data Studio. Figure 10 shows the step in the Configure Automatic Maintenance task assistant to define a database backup policy:
**Figure 10. Defining backup policy step in the Configure Automatic Maintenance task assistant**

When you complete your selections, click on *Preview Command* to verify the DB2 command that will be issued. Then click on the *Run* button to issue the command.

**Configuration advisor**

You can use the Configuration Advisor to obtain recommendations for the initial values of the buffer pool size, database configuration parameters, and database manager configuration parameters.

To run the Configuration Advisor, issue the `AUTOCONFIGURE` command from the DB2 CLP. As shown in the following example, specify a value of NONE for the APPLY keyword to indicate that you want to view the configuration recommendations but not apply them:

```
    db2 AUTOCONFIGURE USING
        MEM_PERCENT 60
        WORKLOAD_TYPE MIXED
        NUM_STMTS 500
        ADMIN_PRIORITY BOTH
        IS_POPULATED YES
        NUM_LOCAL_APPS 0
        NUM_REMOTE_APPS 20
        ISOLATION RR
        BP_RESIZEABLE YES
        APPLY NONE
```
The recommendations generated by the AUTOCONFIGURE command are displayed in the command output. Figure 11 shows an example of these recommendations for the SAMPLE database:

**Figure 11. Recommendations generated by the AUTOCONFIGURE command**

![Image showing AUTOCONFIGURE output with recommendations for SAMPLE database]

**Utility throttling**

You can limit the performance degradation of a throttled utility on production workloads by defining an impact policy through the `util_impact_lim` database manager configuration parameter. The DBA can then run online utilities during critical production periods certain that their performance impact on production work is within acceptable limits. The default value is 100% of allowable impact on workload.

Most users benefit from setting `util_impact_lim` to a low value between 1 and 10. For example, if you want to limit a throttled backup invocation to have an impact on the production workload of 10%, set `util_impact_lim` to a value of 10 as follows:

```
db2 UPDATE DBM CFG USING util_impact_lim 10
```

**IBM Data Studio**

IBM Data Studio is included with all DB2 10.1 editions. IBM Data Studio provides a single integrated environment for database administration and application development. You can
perform tasks that are related to database modeling and design, develop database applications, administering and manager databases, tune SQL performance, and monitor databases all in one single tool which makes it ideal for a team environment with different roles and responsibilities.

IBM Data Studio consist of the following components:

- The IBM Data Studio client, which is an Eclipse-based tool that provides an integrated development environment for database and instance administration, routine and Java application development, and query tuning tasks.
- The IBM Data Studio web console, which is a web-based tool with health and availability monitoring, job scheduling, and database administration tasks.

These are some of the tasks that you can perform with IBM Data Studio:

- Connecting to databases
- Generating DDL scripts
- Performing database development tasks
- Scheduling jobs
- Visual Explain

The following sections of this tutorial show good examples on how to use this tool for these tasks.

**Connecting to databases**

To perform any task against a database, you must first establish a database connection.

To create database connections:

- Open the Database Administration Perspective as shown in Figure 12:
Figure 12. Database Administration Perspective

- Click the **New Connection to a Database** icon in the Administration Explorer toolbar to add a new database connection. Alternatively, right-click on **All Databases** and select **New Connection to a Database** as shown in Figure 13:
Figure 13. Adding a new Connection in the Administration Explorer

- In the New Connection window shown in Figure 14, select a database manager. You can select from different IBM data servers and other data sources. Then select the correct JDBC driver from the drop down menu, and enter the necessary connection information in Properties.
Figure 14. New Connection window

Click **Test Connection** to verify that you can successfully connect to the database using the information that you specified. Once the test connection is successful, click **Finish** to create the connection.

Generating DDL scripts

You can generate and run a DDL script that creates all the objects in a database.

- Right-click on database name and select **Connect**.
- Right-click on the database name and select **Generate DDL** as shown in Figure 15.
Figure 15. Generating DDL script for a database

- Select the options in the Generate DDL window to determine what elements to include in the DDL script as shown in Figure 16:
Figure 16. Generate DDL options

• Click **Next** and select the objects that you want to generate statements for as as shown in Figure 17:
Figure 17. Generate DDL objects

- Click **Next**, specify the location and name of the script, and select whether to run the generated DDL script or edit it in the SQL editor as shown in Figure 18:
Figure 18. Run or edit options

- Click **Next** and review your selections in the Summary. You can click **Back** to change any of your selections. Click **Finish** when you are ready to generate the DDL script as shown in Figure 19:
Figure 19. Summary in Generate DDL

- Edit generated DDL script using the SQL editor and run when you complete your updates as shown in Figure 20:
Performing database development tasks

You can perform database development tasks with IBM Data Studio such as creating and running SQL Statements, developing stored procedures and user defined functions, and developing database applications.

Creating and running SQL or Xquery Statements

To create and run SQL or Xquery Statements:

- Click Task Launcher as shown in Figure 21. Alternatively, select Task Launcher under Help.
Figure 21. Task Launcher

![Figure 21. Task Launcher](image)

- Select **Develop** and then click **Create and run SQL or Xquery Statements** as shown in Figure 22:

Figure 22. Creating and running SQL or Xquery Statements

![Figure 22. Creating and running SQL or Xquery Statements](image)
• Select a project name, enter a name for the DDL script, and select **SQL and XQuery editor** as shown in Figure 23:

**Figure 23. Creating a DDL script with SQL and XQuery editor**

![Creating a DDL script with SQL and XQuery editor](image)

Click **Finish** and the **Select Connection Profile** window opens as shown in Figure 24. Select an existing connection or create a new one.
Figure 24. Running a DDL script in the Data Explorer

The Data Project Explorer opens as shown in Figure 25. Start using it to create, save, and execute SQL scripts, stored procedures, and UDF’s.
Alternatively, If you want to use the SQL Query Builder, click the radio button for it after selecting the project name and entering the script name as shown in Figure 26. SQL Query Builder offers similar capabilities to the SQL and XQuery editor.
You can type the SQL statement in SQL Source pane (top) or use the Tables pane (middle) to select tables or define joins between tables for SELECT statements as shown in figure Figure 27. You can also open the Expression Builder wizard to guide you through the creation of complex expressions.
Figure 27. Running a DDL script with the SQL Query Builder

- Click on one of the following items to run the SQL statement:
  - Run > Run SQL
  - SQL > Run SQL
  - Run the SQL Statement in the main toolbar

Scheduling jobs
You can use the graphical interface provided by IBM Data studio web console to easily create, schedule, and manage jobs for DB2 databases.

You can perform the following actions in the job manager:

- Add, edit, delete and run jobs manually from the Job list page. For more information, see Creating jobs.
- Add, edit, and delete schedules on jobs from the Schedules page. You can also view all the enabled and disabled jobs. For more information, see Scheduling jobs.
- Add jobs together in chains. For more information, see Adding jobs together in chains.
- Add, edit, and delete notifications on jobs from the Notifications page. For more information, see Setting up job notifications.
• Check the status of completed jobs from the History page. You can also check other information such as start time, end time, and database name. For more information, see Reviewing job history.

Creating database connections

A database connection is required to start managing jobs in IBM Data Studio web console. To create a database connection:

• Click on the Open menu and select Databases as shown in Figure 28. Alternatively, click Add database connections in the Getting Started section of the Task launcher.

Figure 28. Open Database page

![Open Database page](image)

• Click Add a Database Connection in the Databases page as shown in Figure 29:

Figure 29. Database connection page

![Database connection page](image)

• Fill in the information in all mandatory fields marked with * character as shown in Figure 30:
Figure 30. Adding a database connection

Click **Test Connection** to verify that you can successfully connect to the database using the information that you specified. Once the test connection is successful, click **OK**.

- Verify the database connection is listed in the **Databases** page as shown in Figure 31:
Once you have created database connections, you can start scheduling jobs over these databases with the job manager.

Creating jobs

To open the Job Manager page in IBM Data Studio web console:

- First, open IBM Data Studio web console.
- Click on the Open menu and select Job Manager or click Manage database jobs in the Task Launcher page as shown in Figure 32:
Figure 32. Task launcher IBM Data Studio web console

- Click **Add Job** in the **Job Manager** page as shown in Figure 33:

Figure 33. Job Manager page

- Enter the job information as shown in Figure 34:
Figure 34. Adding a job

- Enter a script for the new job in the Job list page and click Save All as shown in Figure 35:
Figure 35. Adding a script

- Click Run Job to manually run a job without scheduling it for testing purposes as shown in Figure 36:
Figure 36. Running a job

Select the database on which you want to run the job and click **OK** as shown in Figure 37:
Figure 37. Selecting a database to run a job

A message box opens to let you know the job has been started and to open the **History** page to check on job status as shown in Figure 38:
Figure 38. Selecting a database to run a job

In the History page, you can see that job has been started as shown in Figure 39:
Figure 39. Job History

Scheduling jobs

A schedule defines when a job is run, whether the job is repeating, and whether the schedule is limited in number of runs or in time. The schedule is also associated with one or more databases on which the job is run. A job can have any number of schedules attached to it, but each schedule entry only applies to one job. To schedule a job:

- Select the job to schedule in the **Job List** page and click **Edit**.
- Click **Schedules** in **Job Components**. Then, click **Add Schedule** as shown in Figure 40:
Figure 40. Adding a schedule to a job

- Specify a start date and start time for the job as shown in Figure 41. If you want the job to repeat, select the **Repeats** checkbox and set the repetition parameters for the job. A schedule must be active to run the job.
Figure 41. Specifying parameters for a job schedule

- Specify one or more databases against on which you want to schedule this job in the **Databases** page as shown in Figure 42
Click **Apply Changes** or **Save All** to save all changes to schedule the job. The new schedule appears in the **Schedules** drop down list as shown in Figure 43:
Adding jobs together in chains

Adding jobs together in chains creates a sequence where the main job is followed by a secondary job dependent on the outcome of the main job, and where a finishing job is run last.

To add jobs together in a chain:

- Select the main job to add to a chain in the Job List page and click Edit.
- Click Chain in Job Components. Then, select the options for the secondary job that you want to add to the chain as shown in Figure 44:
Figure 44. Chain in Job Components

- Click **Save All** to save all changes and add the jobs together in a chain. Check the job now shows Yes in the Chain column as shown in Figure 45:
Figure 45. Chain information in Job list

Setting up job notifications

You can configure email notifications to be sent to one or more users depending on the success or failure of the job. To set up email notification:

- Select the job to which you want to set up email notifications in the Job List page and click Edit.
- Click Notifications in Job Components. Click Add Notification as shown in Figure 46:
Figure 46. Adding Job Notifications

- Enter the email address to which you want to send the notifications. If necessary, follow the link to configure the Email Service. Then, specify databases and notification criteria as shown in Figure 47:
Figure 47. Specifying information for Job Notification

- Click **Apply Changes** or **Save All** to complete the notification setup. The new notification appears in the **Notifications** drop down list as shown in Figure 48:
Reviewing job history

Once the job is scheduled, you can use the History page to review the job history and get detailed information by looking at the log entry for the job. You can configure the job manager to retain job history for all jobs that were run, or for a subset depending on the success or failure of the job. To review job history:

- Click History in the Job Manager page. The job history view shows a high-level overview of the job results. You can open the individual log for each job to find more details. Figure 49 shows the History page:
Figure 49. Job History page

The log shows the output of the job script and any exceptions or other messages related to the job.

Visual Explain

With Visual Explain, you can generate a diagram of the current access plan for an SQL or XPATH statement to find out how the SQL compiler processes the statement. You can use the information available from the graph to tune your SQL statements to improve their performance.

Visual Explain requires a function that configures the database to enable query tuning. You need to activate the InfoSphere Optim Query Workload Tuner (OQWT) license for advanced tuning capability prior to using Visual Explain. The IBM DB2 Advanced Enterprise Server Edition includes a license for this tool.

To generate these diagrams:

- Right-click on the database name in the Administration Explorer and select **Start Tuning** to open the Query Tuner Workflow Assistant as shown in Figure 50:
Figure 50. Start Tuning in the Administration Explorer

- Enter your SQL statement in **Text of SQL Statement** and click **Invoke Advisors and Tools** as shown in Figure 51:
Figure 51. Query Tuner Workflow Assistant

Under the DB2 for Linux, UNIX, and Windows sources in the left panel, see the tuning-related information such as Package Cache, Package, Explain tables, and event monitors. Also, the left panel shows the **Capture** section as the active one. The other sections of this wizard are Status, Manage, Invoke, Review, and Compare.

In the **Invoke** section, the Run Single-Query Advisors and Analysis Tools page opens.

- Click **Select What to Run** in the Run Single-Query Advisors and Analysis Tools page as shown in Figure 52:
Figure 52. Run Advisors and Analysis Tools

- Select **Display access plan graph** in the **Select Activities** window and click **OK** as shown in Figure 53:
Figure 53. Run Advisors and Analysis Tools

A window showing progress of the SQL statement execution opens with an option to run the statement in the background as shown in Figure 54:

Figure 54. Tune Query progress
This option is useful for long-running queries.

- After the statement execution is completed, select **Open Access Plan Graph** under **Single Query** in the **Review** section to see the diagram of the access plan as shown in Figure 55:

**Figure 55. Diagram of an access plan**

In the diagram, the nodes in the graph represent tables and indexes and each operation on them for this particular query. The links between the nodes represent the flow of data.

- Select **Open Summary Report** under **Single Query** in the **Review** section as shown in Figure 56:
Conclusion

In this tutorial, you have been introduced to the following concepts in preparation for the DBA certification exam 611:

- How to create, list, update, and drop instances in a DB2 data server.
- What are registry and environment variables and how to set them up.
- How to set up database manager and database configuration parameters.
- How to use IBM Data Studio to connect to a database and to set the database manager and database configuration parameters.
- What is autonomic computing and how to enable and disable it.
- What is STMM, automatic storage, automatic maintenance, configuration Advisor, and utility throttling.
- How to use IBM Data Studio to enable or disable STMM and automatic maintenance.
- How to query SQL and X query statements using IBM Data Studio and how to use the query editor tools.
• How to create manage and schedule jobs using the IBM Data Studio web console.
• How to use visual explain to generate diagrams for access plans.

For additional information to help you prepare for the DBA certification exam 611, see the Related topics section.
Related topics

- Read the Preparation Guide for DB2 10.1 DBA for Linux, UNIX, and Windows Exam 611 to learn in-depth information about each of the concepts presented in this tutorial. This guide is compilation of topics from the DB2 10.1 documentation.
- Use the DB2 documentation in IBM Knowledge Center to find more details about each of the concepts presented in this tutorial.
- Read the Preparation Guide for DB2 10.1 DBA for Linux, UNIX, and Windows Exam 611 to learn in-depth information about each of the concepts presented in this tutorial. This guide is compilation of topics from the DB2 10.1 Information Center documentation.
- Use the DB2 10.1 Information Center to find more details about each of the concepts presented in this tutorial.
- Use the Managing and scheduling database jobs with the Data Studio Web Console tutorial to learn more details about how to create and manage database jobs and to schedule command scripts to run automatically.
- Read an overview of the new capabilities in the DB2 10.1 Announcement letter.
- Get a DB2 10.1 trial download for AIX, HP-UX Itanium, Linux for System i, Linux for System p, Linux for System x86, Linux for System x86-64, Linux for System z (64 bit), Windows (32bit), Windows (64bit).
- Download IBM Data Studio at no charge from this developerWorks page that offers full product images for all platforms.