



Getting started with DB2

IBM Information Management Cloud Computing Center of Competence
IBM Canada Labs

Agenda

- DB2 server editions, clients and drivers
- DB2 Express-C overview
- The DB2 Command Line Processor (demo)
- The DB2 Environment
- Configuring DB2
- Working with scripts
- Connecting to a DB2 server
- Data movement utilities

Supporting reading material & videos

- **Reading materials**

- Getting started with DB2 Express-C eBook
 - Chapter 1: What is DB2 Express-C?
 - Chapter 3: DB2 installation
 - Chapter 4: DB2 Environment
 - Chapter 5, section 5.10: Scripting
 - Chapter 6, section 6.3: DB2 Storage Model
 - Chapter 7: DB2 Client connectivity
 - Chapter 9: Data movement utilities

- **Videos**

- db2university.com course AA001EN
 - Lesson 2: Getting started with DB2
 - Lesson 4: Working with scripts
 - Lesson 5: Connecting to a DB2 server

05/24/2011

Template Documentation

Agenda



- **DB2 server editions, clients and drivers**
- DB2 Express-C overview
- The DB2 Command Line Processor (demo)
- The DB2 Environment
- Configuring DB2
- Working with scripts
- Connecting to a DB2 server
- Data movement utilities

DB2 server editions

DB2 Express-C

05/24/2011

Template Documentation

5

© 2011 IBM Corporation

A DB2 server (also called a DB2 data server, or a DB2 database server) is a system where you have installed one of the DB2 server editions. A DB2 server should accept incoming connections from local and remote DB2 clients. DB2 has been developed using the C/C++ language, and its tools have been developed using Java. This applies to all DB2 editions on Linux, UNIX, Windows and the Mac. There are different DB2 server editions, and they are built on top of each other:

DB2 Express-C is the core of DB2.

DB2 server editions

DB2 Express-C + Extra
functionality

When we add more functionality to DB2 Express-C,

DB2 server editions

DB2 Express Edition

DB2 Express-C

05/24/2011

7

Template Documentation

© 2011 IBM Corporation

it becomes DB2 Express.

DB2 server editions

DB2 Express Edition

DB2 Express-C

+ Extra
functionality

05/24/2011

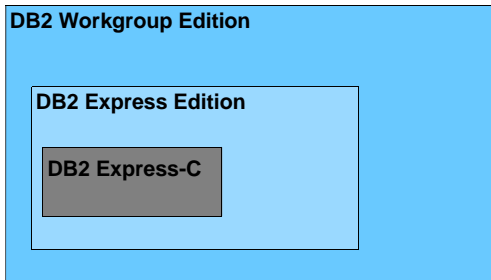
8

Template Documentation

© 2011 IBM Corporation

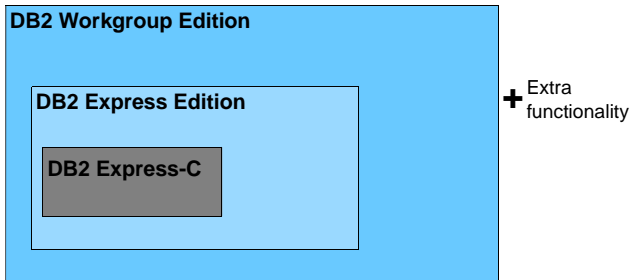
When we add more functionality to DB2 Express,

DB2 server editions



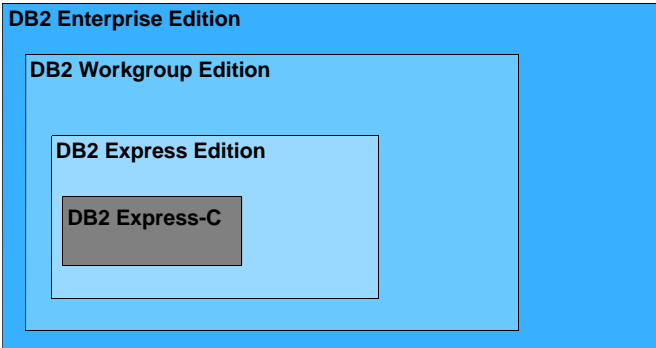
it becomes DB2 Workgroup,

DB2 server editions



and when we add more functionality to DB2 Workgroup,

DB2 server editions



05/24/2011

Template Documentation

11

© 2011 IBM Corporation

it becomes DB2 Enterprise. Choosing a server edition depends on your needs. An edition is just how we package the product; it gives users the choice of selecting what package they want, depending on their needs. For example, for small startup companies, their best option may be to use DB2 Express-C since it's free. On the other hand, for an established large enterprise looking for features such as compression, high availability, Database Partitioning, and so on, their best option may be DB2 Enterprise. As shown earlier, DB2 editions are built on top of each other using the same C/C++ code. So the code used to create DB2 Express-C is the same as the other editions. What this means is that, if you *learn* DB2 Express-C, you are also learning all the other editions of DB2 since DB2 Express-C is the core. Moreover, if you *develop* an application for DB2 Express-C, the same application with no modification will work on all the other DB2 editions; again, because DB2 Express-C is the core.

All DB2 server editions include a client component and the GUI Tools. A DB2 server can behave as a client system to another DB2 server. So for example, if I connect from a DB2 Express-C server to a DB2 Enterprise edition server, the DB2 Express-C server is basically behaving as the client.

DB2 server editions

DB2 Enterprise Edition

DB2 Workgroup Edition

DB2 Express Edition

DB2 Express-C

Latest version for DB2 on Linux, UNIX and Windows is DB2 9.7

05/24/2011

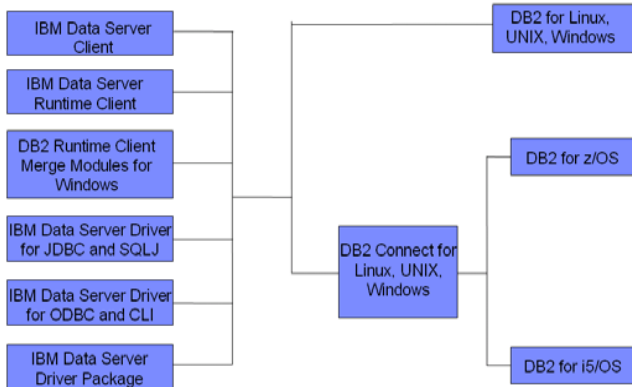
Template Documentation

12

© 2011 IBM Corporation

The latest version for DB2 for LUW is DB2 9.7

DB2 clients and drivers



05/24/2011

Template Documentation

13

© 2011 IBM Corporation

DB2 clients and drivers provide the necessary functionality to connect and work with a DB2 server. Drivers are included with the DB2 clients, or available separately. Depending on the programming interface, you may need to install just a DB2 driver, or an entire DB2 client. For example, a JDBC Type 4 application only requires a JDBC driver to be installed to connect to a DB2 server, you don't need to install the whole client. All DB2 servers include DB2 client components; therefore, once you install a server, it can connect to local databases (databases created on the same server), and also remote databases (databases created on other servers). DB2 clients and drivers are all free and available for download from the DB2 Express-C web site.

At the top left corner of the chart you have 3 clients listed. The clients with a name starting with "IBM Data Server" can work not only with DB2, but also with other IBM data servers, such as Informix. The IBM Data Server Client is the most complete. It includes graphical tools, and all required drivers for different programming languages. The IBM Data Server Runtime Client is a lightweight client with basic functionality, and includes the drivers. The DB2 Runtime Client Merge Modules for Windows is mainly used to embed a DB2 runtime client as part of a Windows application installation. At the bottom left corner of the chart you have 3 drivers listed:

- IBM Data Server Driver for JDBC and SQLJ: allows Java applications to connect to DB2 servers without having to install a client
- IBM Data Server Driver for ODBC and CLI may allow ODBC and CLI applications to connect to a DB2 server without having to install a client
- IBM Data Server Driver Package includes a Windows-specific driver with support for .NET environments in addition to ODBC, CLI and open source. On the right side we have the DB2 servers listed. The focus of this video series is on DB2 for Linux, UNIX, and Windows. To connect to DB2 servers on the mainframe (DB2 for z/OS) and on the midrange servers (DB2 for i5/OS) you need to purchase a software called DB2 Connect.

Agenda



- DB2 server editions, clients and drivers
- **DB2 Express-C overview**
- The DB2 Command Line Processor (demo)
- The DB2 Environment
- Configuring DB2
- Working with scripts
- Connecting to a DB2 server
- Data movement utilities

What is DB2 Express-C?

- **Free version of DB2 database server**
- **The “C” in the name stands for “Community”**
- **Free to develop, deploy, and distribute...no limits!**
 - NO user limit
 - NO DB2 instance limits
 - NO database size limit

05/24/2011

15

Template Documentation

© 2011 IBM Corporation

DB2 Express-C is the FREE version of IBM's DB2 database server. The 'C' in the name stands for “Community”, so DB2 Express-C has been created for the community of developers, students, teachers, professionals, and everyone else interested in working with databases. It is FREE to develop, deploy, and distribute with no limits. Free to develop means that if you are an application developer and need a database for your application, you can use DB2 Express-C for free. Free to deploy means that you can use DB2 Express-C in production for free. For example, you may have an e-commerce Web site where you are selling toys. The pictures of the toys you sell along with the price, description, and so on, can be stored in a DB2 Express-C database. Free to distribute means that you can embed DB2 Express-C as part of your application. When you sell your application, even though DB2 Express-C is included and distributed with your application; you do not need to pay anything to IBM. Therefore DB2 Express-C can help you make money! “No limits” refers to the fact that there is no limit on the number of users to the database, no limit on the number of DB2 instances you can create, and no database size limit, that is, your database can grow to any size (Other vendors offering free editions of their products normally set a 4GB database size limit).

Where can DB2 Express-C run?

- **Windows**

- Windows workstation platforms (XP, Vista, Windows 7)
- Windows server platforms (2003, 2008)

- **Linux**

- Red Hat, Suse, Ubuntu and more

- **Mac OS X (beta)**

- **Solaris**

- **Supported on 32-bit and 64-bit systems**

DB2 Express-C can run on Windows (XP,Vista, and Windows 7) for workstation platforms, and on Windows server 2003 and 2008.It can run on Linux (Red Hat, Suse, Ubuntu and others). There is a beta version for the Mac OS X, and Solaris. It is supported on 32-bit and 64-bit systems.

What are DB2 Express-C requirements?

• Minimum requirements

- Memory:
 - 256MB, 512MB (DB2 GUI tools), 1GB (recommended)
- Disk:
 - Depends on number and size of databases to create
 - On Linux, 2GB on /tmp is recommended

• Maximum requirements:

- System of any size
- Uses at most 2 cores and 2 GB of RAM

DB2 can be installed in most systems today. In terms of minimum memory requirements, DB2 Express-C can run with as low as 256Mb. If using DB2 graphical tools, you need at least 512Mb. The recommendation is to have at least 1GB of RAM. In terms of minimum disk requirements, this will depend on the number of databases you plan to create and their sizes. On Linux, it is recommended to have 2GB on /tmp.

In terms of maximum requirements, DB2 Express-C can be installed on a system of any size. This means the system can be very large (for example, 16 CPU cores and 32GB of RAM); however DB2 Express-C will automatically limit itself to use 2 cores and 2GB of RAM. Other DB2 editions allow for larger usage of resources.

Downloading DB2 Express-C: ibm.com/db2/express

05/24/2011

18

Video demos can be found in: db2university.com

© 2011 IBM Corporation

Review the video in db2university.com to see how to download and install DB2 Express-C

Installing DB2 Express-C (Demo)

The screenshot shows the 'DB2 Setup Launchpad' window. On the left, a sidebar contains links: 'Welcome', 'Release Information', 'Installation Requirements', 'Upgrade Information', 'Install a Product' (highlighted with a red box), and 'Exit'. The main area is titled 'Install a Product' and features a 'Click Install New' button (also highlighted with a red box) to launch the DB2 Setup wizard. Below this, it describes 'DB2 Express-C Version 9.7' as a full-function DB2 data server. Further down, it mentions 'Database Management and Application Development Tools'. At the bottom of the main area, two buttons are visible: 'Install New' (highlighted with a red box) and 'Work with Existing'.

DB2 Setup Launchpad

Welcome

Release Information

Installation Requirements

Upgrade Information

► Install a Product

Exit

Install a Product

Click Install New to launch the DB2 Setup wizard for the desired product and to install the product to a new location. If you want to update, upgrade, or add features to an existing product, click Work with Existing. You can also launch other product installations by clicking Install for the corresponding product.

DB2 Express-C Version 9.7

DB2 Express-C contains a full-function DB2 data server, which provides an entry-level product for the Small and Medium Business (SMB) market. It comes with simplified packaging, and is easy to transparently install from within an application. It is available on Linux and Windows platforms, is fully compatible with, scalable to, and has all the automatic manageability features of its higher priced family of offerings.

DB2 Express-C comes with pureXML/IBM data storage capabilities and delivers flexible access to XML data using XQuery, XPath and SQL. It is optimized for powering Web Services, Web 2.0, and SOA-based solutions. If you require IBM support and maintenance subscription for this no-charge data server, you can purchase DB2 Express-C with renewable 12 Months Subscription. For more information, see <http://www.ibm.com/db2express>.

Install New Work with Existing

Database Management and Application Development Tools

A suite of GUI tools for managing DB2 for Linux, UNIX, and Windows data and data-centric applications is available to install. Individually these tools provide powerful capabilities that target specific data management roles and tasks; more importantly, the components interoperate seamlessly, enabling cross-role collaboration, productivity, and effectiveness.

IBM

05/24/2011

19

Video demos can be found in: db2university.com

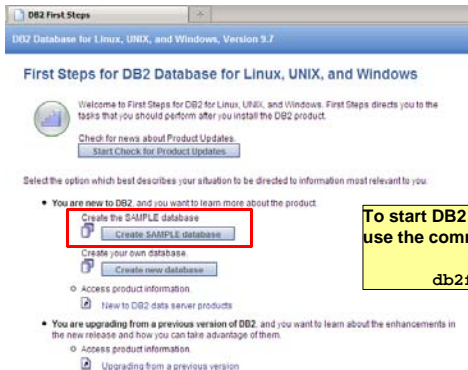
© 2011 IBM Corporation

Review the video in db2university.com to see how to download and install DB2 Express-C

Note that this screenshot shows two buttons "Install New" and "Work with Existing". If this is the first time you are installing DB2 Express-C, you will only see one button, "Install New". On the other hand, if you already had DB2 installed in your computer, then a second button "Work with Existing" will show up too. This basically means that you can install several versions of DB2 (or the same version several times), or install DB2 at different fixpack levels. Eg:

DB2 9.5, DB2 9.7, DB2 9.7 FP2, DB2 9.7 FP4, etc

DB2 First Steps & SAMPLE database (Demo)



To start DB2 First Steps,
use the command:

db2fs

Video demos can be found in: db2university.com

05/24/2011

20

© 2011 IBM Corporation

Right after the installation of DB2, your browser should have automatically launched "DB2 First Steps". If it did not, you can manually launch it from a DB2 command window or Linux shell using the command "db2fs".

If you are using Firefox, you may be prompted to create a DB2_FIRSTSTEPS profile. We recommend you to create it so you can use DB2 First Steps to its fullest. If you don't want to, click on "Do not create profile" or simply close the window as in my case. The First Step tool is self-explanatory. Most people use it to create the SAMPLE database for the first time.

The SAMPLE database is a database provided with DB2 so you can "play" with it. The SAMPLE database may have been created automatically for you during installation. You can verify this from the DB2 Command window by typing:

db2 connect to sample

If you get an error, you need to create it manually. Go back to the DB2 First Steps tool, and click on the "Create SAMPLE database" button. A window will appear. Choose the second option "XML and SQL objects and data". This will create the SAMPLE database and include XML data which can be used later to test how DB2 works with XML. Click OK to start creating the SAMPLE database. This will take a few minutes. After the SAMPLE database is created, you can verify you can connect to it and retrieve information. From the DB2 Command Window or Linux shell, type:

db2 connect to sample

db2 "select * from employee"

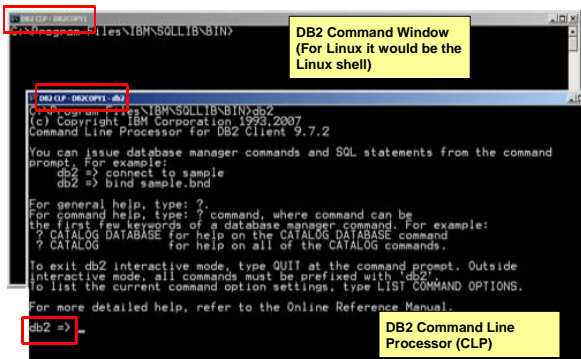
If both statements are successful (the SELECT should return several rows), you have successfully created the SAMPLE database. You can also use the command line interface command "db2samp" to create the SAMPLE database.

Agenda



- DB2 server editions, clients and drivers
- DB2 Express-C overview
- **The DB2 Command Line Processor (demo)**
- The DB2 Environment
- Configuring DB2
- Working with scripts
- Connecting to a DB2 server
- Data movement utilities

DB2 Command Line Processor / DB2 Command Window



Video demos can be found in: db2university.com

05/24/2011

22

© 2011 IBM Corporation

On Windows you have the DB2 Command Window which looks like a regular MS-DOS window, but the window title says "DB2 CLP". From this window you can execute DB2 commands. On Linux there is no DB2 Command Window, but you use the Linux shell instead (as long as you are logged on with a user that has his environment set up to work with DB2, such as the DB2 instance owner which is normally db2inst1). On both, Linux and Windows, you have the Command Line Processor (CLP). What is the difference between the DB2 Command Window/Linux shell and the CLP? Take a look at the prompt. If the prompt is "db2 =>" then you are in the CLP and you don't need to prefix any DB2 command with "db2".

Eg:

db2 => connect to sample

On the other hand if the prompt is from the operating system, you are in the DB2 Command Window/Linux shell, and therefore you need to prefix DB2 commands with "db2".

Eg:

C:\> db2 connect to sample

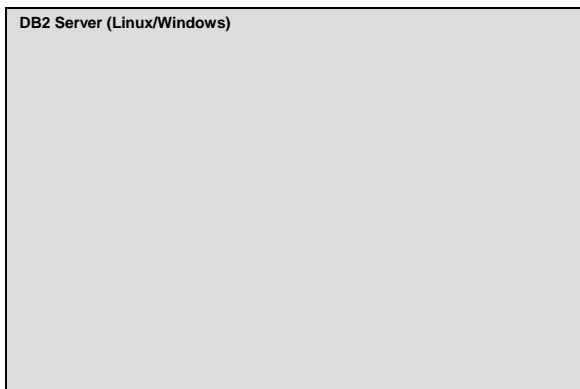
Agenda

- DB2 server editions, clients and drivers
- DB2 Express-C overview
- The DB2 Command Line Processor (demo)



- **The DB2 Environment**
- Configuring DB2
- Working with scripts
- Connecting to a DB2 server
- Data movement utilities

The DB2 environment



05/24/2011

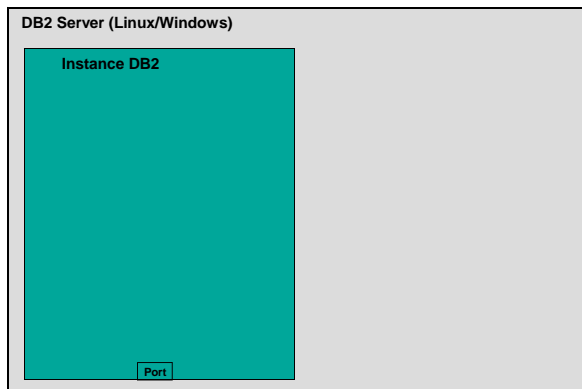
Template Documentation

24

© 2011 IBM Corporation

The box in the figure shows a representation of a DB2 data server after installing DB2 Express-C 9.7.

The DB2 environment



05/24/2011

Template Documentation

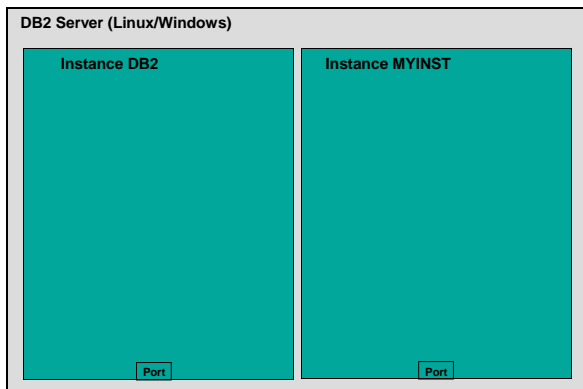
25

© 2011 IBM Corporation

As part of the installation in Windows, a default instance called DB2 (db2inst1 on Linux) is created. This is represented by the green box on the left of the figure. An instance is simply an independent environment where applications can run, and databases can be created. You can create multiple instances on a data server, and use them for different purposes. For example, one instance can be used to hold databases for production use, another instance can be used for test environment databases, and another one for a development environment. All of these instances are independent; that is, performing operations on one instance will not affect the other instances.

To create a new DB2 instance, use the command `db2icrt <instance name>`, where `<instance name>` is replaced with any 8 character name. For example, to create the instance `myinst`, we use this command: `db2icrt myinst`.

The DB2 environment



05/24/2011

Template Documentation

26

© 2011 IBM Corporation

The figure shows a new instance called myinst as a separate green box on the right side.

Note that each instance has a unique port number. This helps to distinguish between instances when you want to connect to a database in a given instance from a remote client using TCP/IP. If you use the DB2 Command Window, you can make any DB2 instance the active one by using this operating system command on Windows:

```
set db2instance=myinst
```

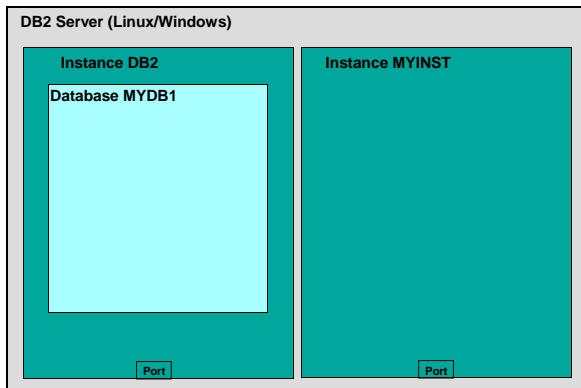
Note that there should not be any blank spaces before or after the equal (=) sign. In this example, if you now create a database from the Command Window, it would be created in the instance myinst.

To list the instances in your system, run the command:

```
db2ilist
```

On Linux, an instance must match a Linux operating system user; therefore, to switch between instances you can simply switch users. This user is known as the instance owner. You can logoff and logon with the instance owner user of the other instance, or use the su command.

The DB2 environment



05/24/2011

Template Documentation

27

© 2011 IBM Corporation

To create a database in the active instance, issue this command from the DB2 Command Window:

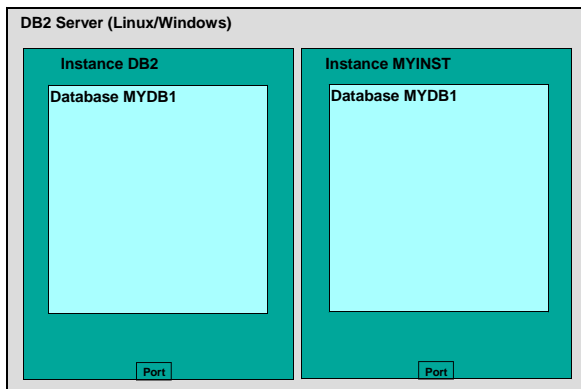
```
db2 create database mydb1
```

To list all the databases created, run the command:

```
db2 list db directory
```

Within any one instance, you can create many databases. A database is a collection of objects such as tables, views, indexes, and so on. Databases are independent units, and therefore, do not share objects with other databases. The figure shows a representation of a database MYDB1 created inside instance DB2.

The DB2 environment



05/24/2011

28

Template Documentation

© 2011 IBM Corporation

If we want to create another database with the same name (MYDB1) but in instance myinst, the following commands from the DB2 Command Window would be issued:

```
db2 list db directory
```

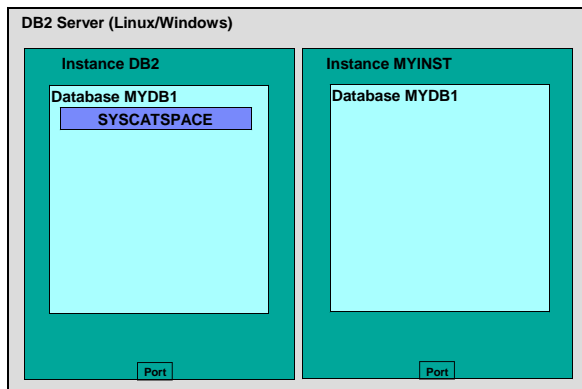
```
set db2instance=myinst (In Linux: su - myinst)
```

```
db2 create database mydb1
```

```
set db2instance=db2 (In Linux: su - db2inst1)
```

The figure shows the new database MYDB1 created in instance myinst.

The DB2 environment



05/24/2011

Template Documentation

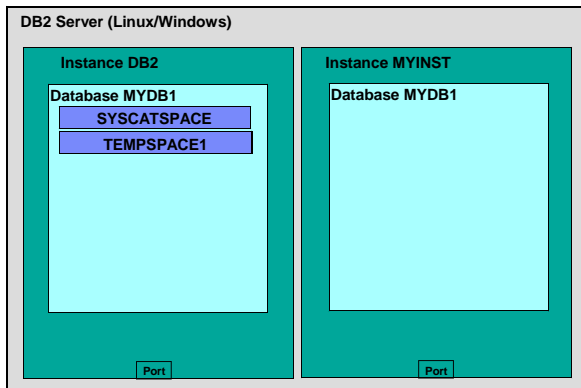
29

© 2011 IBM Corporation

When a database is created, there are several objects created by default: table spaces, tables, a buffer pool and log files. Creating these objects takes a bit of time, that's why the create database command requires a few minutes for processing. The figure shows one of these table spaces created by default on the left side of the figure. Table spaces will be discussed in more detail later on.

Table space SYSCATSPACE contains the System Catalog tables. The System Catalog is called the data dictionary in other relational database management systems. It basically contains system information that should not be modified or deleted; otherwise the database will not work correctly

The DB2 environment



05/24/2011

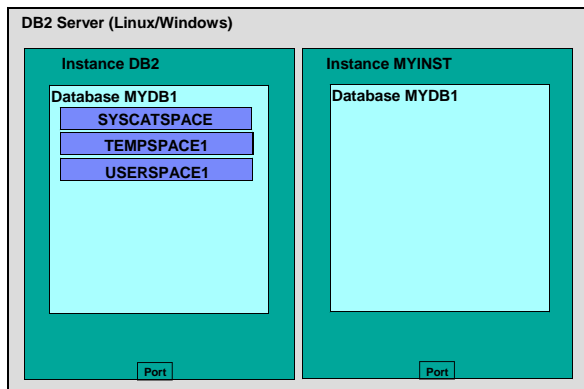
Template Documentation

30

© 2011 IBM Corporation

Table space **TEMPSPACE1** is used by DB2 when it needs additional space to perform some operations such as sorts. This is a **SYSTEM** temporary table space. There must be at least **ONE** system temporary table space at all times.

The DB2 environment



05/24/2011

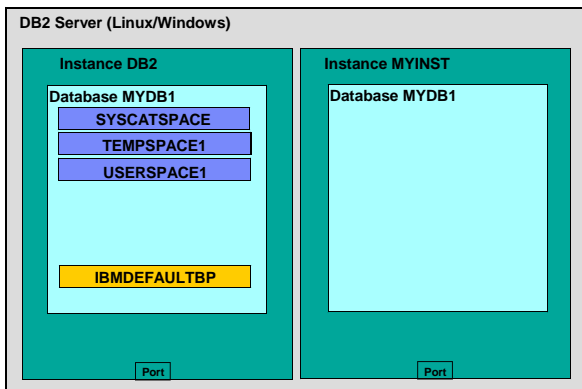
Template Documentation

31

© 2011 IBM Corporation

Table space USERSPACE1 is normally used to store user database tables if there is no table space specified when creating a table.

The DB2 environment



05/24/2011

Template Documentation

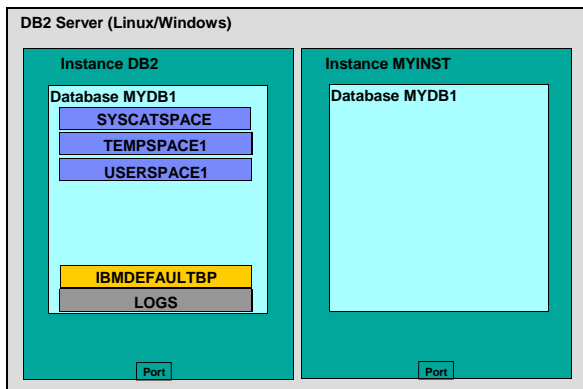
32

© 2011 IBM Corporation

Next we see a bufferpool also created by default when you created the database. The name of this default bufferpool is IBMDEFAULTBP.

A buffer pool is basically a memory cache used by the database. You can create one or more buffer pools, but there should always be one buffer pool with a page size that matches the page size of existing table spaces. A page is the minimum unit DB2 uses to work with your data. There can be pages of 4k, 8k, 16k, 32k

The DB2 environment



05/24/2011

Template Documentation

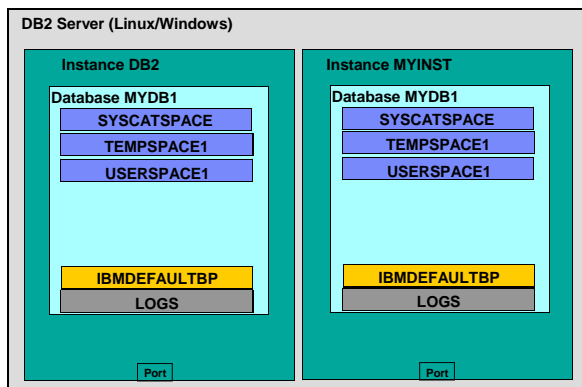
33

© 2011 IBM Corporation

Also created by default when you create a database are log files which are used for recovery.

When you work on a database, not only is information stored in the disks for the database, but while you are working on the database, log files store all the operations executed on the data. Think of logs as temporary files where an "autosave" operation is performed. Logs are discussed in more detail in the presentation about Backup and Recovery.

The DB2 environment



05/24/2011

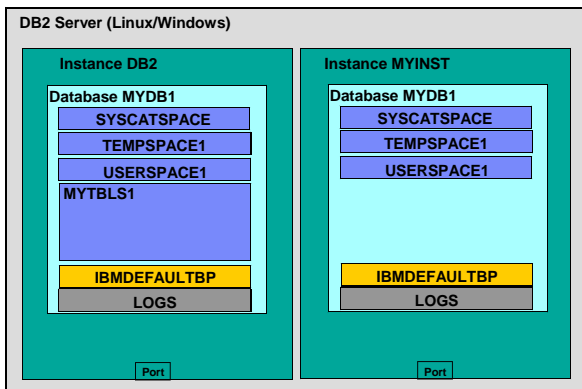
Template Documentation

34

© 2011 IBM Corporation

Note that the same default database objects are created in the other database MYDB1 in instance myinst

The DB2 environment



05/24/2011

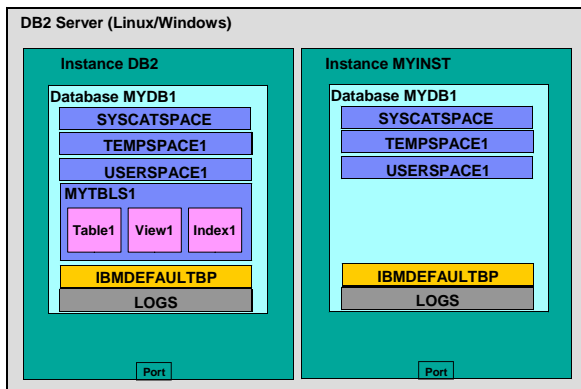
Template Documentation

35

© 2011 IBM Corporation

You can also create your own table spaces using the CREATE TABLESPACE statement. The figure shows the table space MYTBLS1 created inside database MYDB1 on instance DB2. When you create a table space, you specify the disks to use and the memory (buffer pool) to use. Therefore, if you have a "hot" table, that is, a table that is used very often, you can allocate the fastest disks and the most memory by assigning a table space with these characteristics.

The DB2 environment



05/24/2011

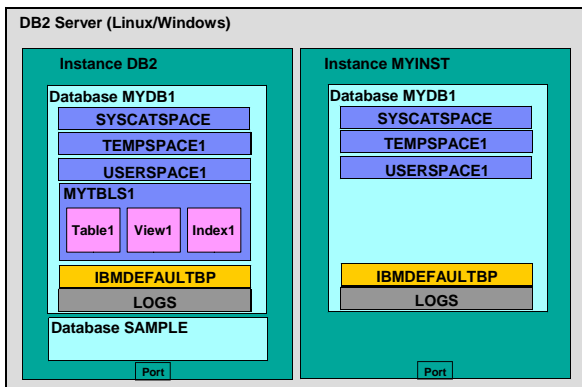
Template Documentation

36

© 2011 IBM Corporation

Inside a table space you can create objects like a table, view, index, etc.

The DB2 environment



05/24/2011

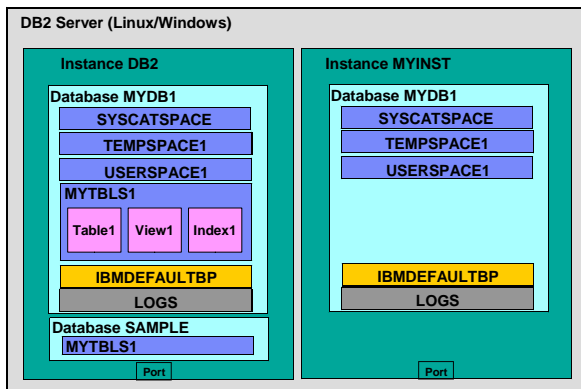
Template Documentation

37

© 2011 IBM Corporation

Earlier we discussed that instances are independent environments, and therefore, a database with the same name could be created in several instances. Just like instances, databases are also independent units; therefore, objects in one database have no relationship to objects in another database.

The DB2 environment



05/24/2011

Template Documentation

38

© 2011 IBM Corporation

Therefore in the figure, we show a table space mytbls1 inside both the MYDB1 database and the SAMPLE database, within instance DB2. This is valid because the databases are independent units. Note that the figure does not show the other default objects of database SAMPLE due to space constraints in the figure.

Summary of instance-level commands

Command	Description
<code>db2start</code>	Starts the current instance
<code>db2stop</code>	Stops the current instance
<code>db2icrt</code>	Creates a new instance
<code>db2idrop</code>	Drops an instance
<code>db2ilist</code>	Lists the instances you have on your system
<code>db2 get instance</code>	Lists the current active instance

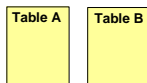
This is a summary of the most popular commands you can use at the instance level

Summary of database-level commands

Command/SQL statement	Description
<code>db2 create database</code>	Creates a new database
<code>db2 drop database</code>	Drops a database
<code>db2 connect to <database_name></code>	Connects to a database
<code>db2 create table/create view/create index</code>	SQL statements to create table, views, and indexes respectively

This is a summary of the most popular commands you can use at the database level

Table spaces



Logical

Physical



05/24/2011

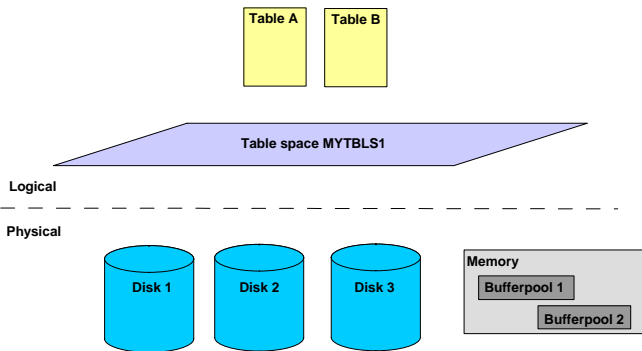
Template Documentation

41

© 2011 IBM Corporation

In the chart, you have physical resources below the dashed line such as disks and memory. Above the dashed line, you have logical objects, such as tables. Now, let's say, "Table B" is a "hot" table, that is, it is used a lot by many users and you want to have the fastest response time when accessing this table. How can you achieve this?

Table spaces



05/24/2011

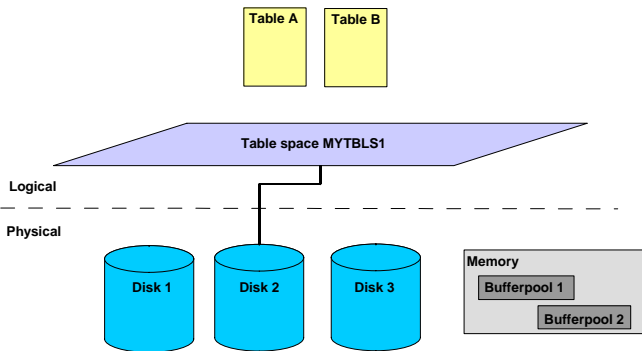
Template Documentation

42

© 2011 IBM Corporation

Well, you can use a table space (let's call it "Table space MYTBLS1") as a logical layer between your fastest physical resources, and this logical table.

Table spaces



05/24/2011

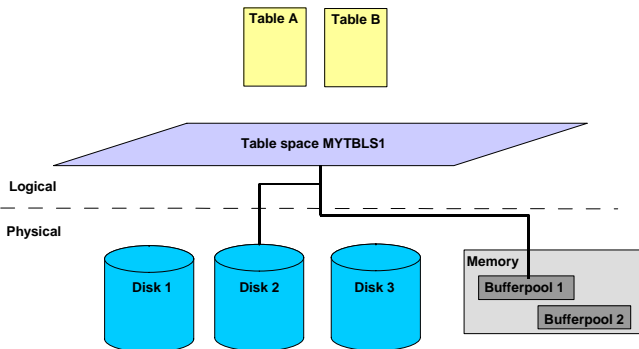
Template Documentation

43

© 2011 IBM Corporation

So if for example, "Disk 2" is the fastest disk, and "Bufferpool 1" is the largest bufferpool (which is database cache), then you associate Disk 2 to tablespace MYTBLS1,

Table spaces



05/24/2011

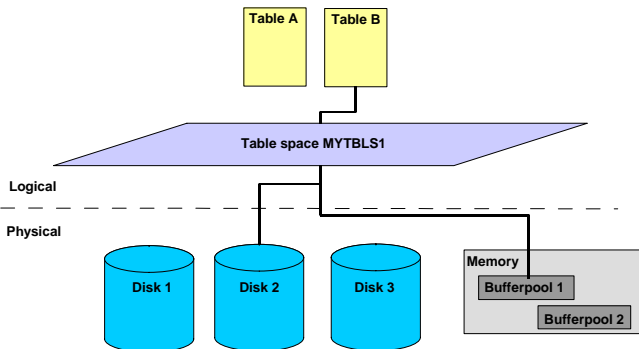
Template Documentation

44

© 2011 IBM Corporation

and then you associate Bufferpool 1 to tablespace MYTBLS1. This is actually done when you create a tablespace.

Table spaces



05/24/2011

Template Documentation

45

© 2011 IBM Corporation

Now, all you need to do is associate Table B to tablespace MYTBLS1.

Table Space Management

MANAGED BY SYSTEM (SMS)

- Managed by the operating system
- Containers are directories

MANAGED BY DATABASE (DMS)

- Managed by DB2.
- Containers are files or raw devices pre-allocated

MANAGED BY AUTOMATIC STORAGE

- Intended as a “single point of storage management” for table spaces
- Create a database and associate a set of storage paths with it
- No explicit container definitions are provided
- Containers automatically created across the storage paths
- Growth of existing containers and addition of new ones completely managed by DB2

Managed by system

This type of table space is known as System Managed Storage (SMS). This means the operating system manages the storage. They are easy to manage, and the containers are file system directories. The space is not pre-allocated, but the files grow dynamically. Once you specify the containers, these are fixed at creation time and other containers cannot be added later, unless a redirected restore is used. When using SMS table spaces the table data, index and LOB data cannot be spread across different table spaces.

Managed by database

This type of table space is known as Database Managed Storage (DMS). This means that DB2 manages the storage. Management of the space requires more manual intervention from a DBA. Containers can be pre-allocated files or raw devices. For raw devices, data is written directly without O/S caching.

Containers can be added, dropped or resized using the ALTER TABLESPACE statement. DMS table spaces are best for performance, and table data, index, and LOB data can be split into separate table spaces, which improves performance.

Managed by automatic storage

Managed by automatic storage tries to bring the best of both worlds:

The ease of management from SMS tablespaces, and
the good performance/flexibility of DMS tablespaces.

You first need to create the DATABASE with automatic storage (Which will be on by default on the create database command). Next you can create tablespaces to use automatic storage. This applies to both SMS and DMS. By default, the containers are created across storage paths. An extent is written to one container, then to the next, and so on. Since the containers are created across storage paths, it is good for performance because, assuming the storage paths reside in different disks, then there will be no bottleneck on just one disk.

Buffer Pool Basics

- Real memory cache for table/index data
- Reduces direct sequential I/O
- At least one buffer pool required per database
- Allocates memory in units of 4K,8K,16K and 32K pages
- At least one matching bufferpool for a table space based on page size
- STMM (Self-tuning memory manager) can automatically resize the bufferpool as needed

A bufferpool is basically cache (memory) that is used for a database. It will basically cache data for tables/views/indexes. A bufferpool will always improve performance because it reduces the number of times DB2 needs to access the disk (direct sequential I/O). Using the bufferpool, DB2 can also pre-fetch pages. This means that based on what you are doing, DB2 may be able to guess which pages you will need from the disk beforehand, and prefetch them into memory = bufferpool. This is done asynchronously, that is, DB2 does not need to do it in sequence, waiting for something else.

The bufferpool should be allocated with a page size of 4k, 8k, 16k and 32k. A page is the minimum unit in which DB2 works. The larger the page, the larger the row can be, however, the more memory and space it will take. Eg: You can create a table as: **create table mytable (col1 varchar(1024), col2 varchar(1024), col3 varchar(1024), col4 varchar(1024))...** this table would not fit into a 4K page, so you need to use 8K page; but you could've used as well a 16K or a 32K page.

As discussed earlier there is an 8k bufferpool called IBMDEFAULTBP which is associated to tablespaces SYSCATSPACE, TEMPSPACE1 and USERSPACE1. With DB2 9, there are some hidden bufferpools per page size as well. For large objects (LOBs) DB2 normally doesn't use the cache. LOBs are too large, and if brought to memory they would overwrite or cause swapping of "good" pages that are needed.

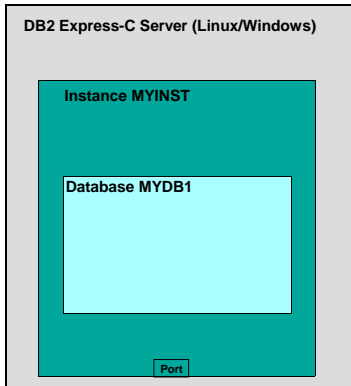
Agenda

- DB2 server editions, clients and drivers
- DB2 Express-C overview
- The DB2 Command Line Processor (demo)
- The DB2 Environment



- **Configuring DB2**
- Working with scripts
- Connecting to a DB2 server
- Data movement utilities

DB2 configuration



05/24/2011

49

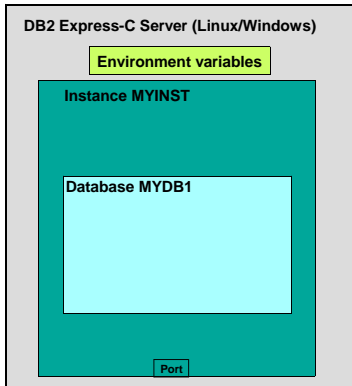
Template Documentation

© 2011 IBM Corporation

A DB2 server can be configured at four different levels:

- Environment variables
- Database manager configuration file (dbm cfg)
- Database configuration file (db cfg)
- DB2 profile registry

DB2 configuration



05/24/2011

Template Documentation

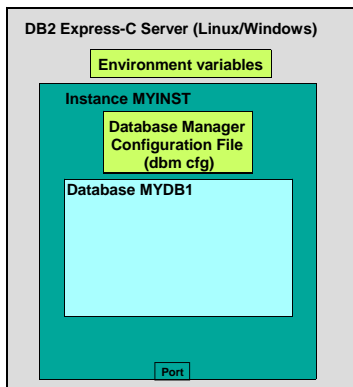
50

© 2011 IBM Corporation

The first level is at the operating system level through “Environment variables”. One key environment variable is DB2INSTANCE. This variable indicates the current instance you are working on, and for which your DB2 commands would apply. For example, to set the active instance to myinst in Windows, you can run this operating system command:

```
set db2instance=myinst
```

DB2 configuration



05/24/2011

Template Documentation

51

© 2011 IBM Corporation

The second level is using the Database manager configuration file (dbm cfg). The dbm cfg file includes parameters that affect the instance and all the databases contained within. To review dbm cfg configuration parameters, issue:

```
db2 get dbm cfg
```

to make changes to a parameter, for example, parameter "INTRA_PARALLEL", issue

```
db2 update dbm cfg using intra_parallel yes
```

Many parameters are dynamic meaning that changes take effect immediately; however, changes to some parameters may require stopping and starting the instance. From the Command Line, this can be done using the db2stop and db2start commands.

Before an instance can be stopped, all applications must disconnect. If you wish to forcefully stop the instance, you can use the db2stop force command.

For more detail on the output of the "db2 get dbm cfg" command, or to see if the value of a parameter is in effect or has been deferred, you can add "show detail" at the end of the command, as in "db2 get dbm cfg show detail" For this clause to work, you need to be attached to the instance first with this command:

```
db2 attach to <instance_name>
```

For example, to attach to the instance "DB2" you can do:

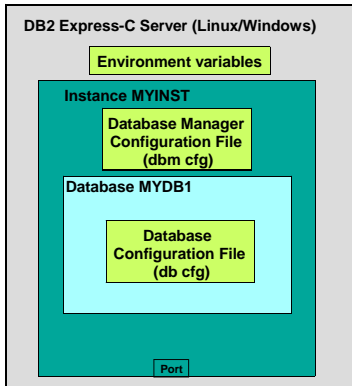
```
db2 attach to db2
```

And then you can do:

```
db2 get dbm cfg show detail
```

You rarely have to explicitly attach to an instance to work with it. Normally attachments are done implicitly.

DB2 configuration



05/24/2011

Template Documentation

52

© 2011 IBM Corporation

The third level is using the Database configuration file (db cfg). This will configure parameters at the database level. To review db cfg configuration parameters, issue:

```
db2 get db cfg for SAMPLE
```

to make changes to a parameter, issue:

```
db2 update db cfg for SAMPLE using LOGFILSIZ 1234
```

If you were already connected to the database, you can remove the “for SAMPLE” clause in the above examples as shown here:

```
db2 connect to sample
```

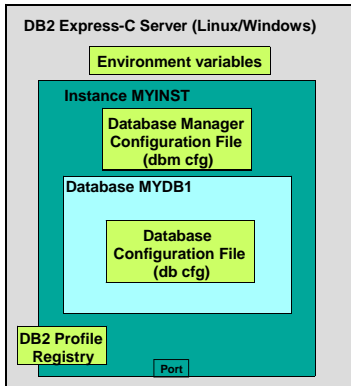
```
db2 get db cfg
```

```
db2 update db cfg using LOGFILSIZ 1234
```

Some DB CFG parameters are dynamic so the change is immediate. Others will not change until all connections are terminated. When the first connection is performed, the new value for the parameter will then take effect.

For more detail on the output of the “db2 get db cfg” command, or to see if the value of a parameter is in effect or has been deferred, you can add “show detail” at the end of the command, as in “db2 get db cfg show detail”. For this clause to work, you need to be connected to the database first.

DB2 configuration



05/24/2011

Template Documentation

53

© 2011 IBM Corporation

Finally, the last level to configure DB2 is using the DB2 profile registry which can be set either at the operating system or instance level. Note that this DB2 registry has no relation whatsoever with the Windows registry. DB2 registry variables may be used for example, to set some special behavior for DB2 for a given operating system to take advantage of an operating system feature.

To make changes to the DB2 Profile Registry variables, use the "db2set" command. After the change is done, you need to issue a db2stop/db2start to ensure the change takes effect.

The parameters at the 4 levels (operating system, dbm, db, DB2 profile registry) are all different. In other words, it's NOT that it's the same parameter that you can set at any of the different 4 levels, but the parameters are all different at each of the 4 configuration levels.

Summary of commands for DB2 configuration

Commands to manipulate the dbm cfg (instance-level)

Command	Description
<code>db2 get dbm cfg</code>	Retrieves information about the dbm cfg
<code>db2 update dbm cfg using <parameter_name> <value></code>	Updates the value of a dbm cfg parameter

Commands to manipulate the db cfg (database-level)

Command	Description
<code>get db cfg for <database_name></code>	Retrieves information about the db cfg for a given database
<code>update db cfg for <database_name> using <parameter_name> <value></code>	Updates the value of a db cfg parameter

v5r4/z/11

Template Documentation

54

© 2011 IBM Corporation

As mentioned earlier, for the db cfg, if you are connected to the database, you do not need to include "for <database_name>" in the syntax.

Summary of commands for DB2 configuration

Commands to manipulate the DB2 Profile Registry

Command	Description
<code>db2set -all</code>	Lists all the DB2 profile registry variables that are currently set
<code>db2set -lr</code>	Lists all the DB2 profile registry variables
<code>db2set <parameter>=<value></code>	Assigns a parameter with a given value

Note:

There should not be spaces at all between the equal sign and the parameter name, and the value in a db2set command. For example:

`db2set db2comm = tcpip` ❌ (incorrect)

`db2set db2comm=tcpip` ✅ (correct!)

05/24/2011

Template Documentation

55

© 2011 IBM Corporation

To review the DB2 profile registry variables that have been set, issue:

```
db2set -all
```

To review all DB2 profile registry variables that can be set issue:

```
db2set -lr
```

To set a variable, say the DB2COMM variable to the value "tcpip" issue something like:

```
db2set db2comm=tcpip
```

Note there must be no spaces before or after the equals sign.

Agenda

- DB2 server editions, clients and drivers
- DB2 Express-C overview
- The DB2 Command Line Processor (demo)
- The DB2 Environment
- Configuring DB2
- **Working with scripts**
- Connecting to a DB2 server
- Data movement utilities



SQL Scripts

- **Pros**

- Simple to understand
- Platform independent

- **Cons**

- Script parameters not supported

SQL scripts, as we will show you soon, are fairly easy to understand.

They are platform independent, that is they are ported with no modification between different platforms such as Windows or Linux.

However, SQL scripts do not allow you to use parameters. Therefore, to really take full benefit of SQL scripts you should combine them with Operating System scripts.

A Basic SQL Script

script1.db2

```
CONNECT TO SAMPLE;  
DROP TABLE arfchong.mytable;  
CREATE TABLE arfchong.mytable  
  ( col1 INTEGER NOT NULL,  
    col2 VARCHAR(40),  
    col3 DECIMAL(9,2) );  
SELECT * FROM arfchong.mytable;  
COMMIT;
```

Let's take a look at a very basic SQL script. Assume you have created the file script1.db2. The name and extension does not matter, you can use any extension you wish. Let's take a look at the contents of this file.

As you can see, it consists of several SQL statements. Note that each statements ends with a semi-colon. This is the default statement terminator in DB2. It allows DB2 to know where a given statement ends. In this example, we first connect to the SAMPLE database, then we drop the table arfchong.mytable, then we create it again. We select all the records from the table, at the end we issue a COMMIT statement.

Note that the DROP TABLE statement is included to "clean things up". Adding the DROP TABLE statement will only give us an error the first time we run the script, indicating the table arfchong.mytable does not exist. Subsequent runs of the same script will run smoothly deleting the table arfchong.mytable first, before creating it again. Including DROP statements is a common practice for DBA and developers. and its common to put them at the beginning of the script so it cleans things up. To execute the script you can use either IBM Data Studio or the DB2 Command Window on Windows or the Linux shell.

Executing SQL Scripts

- An SQL script can be executed from the DB2 Command Window (Windows), Linux shell, or IBM Data Studio
- To run the previous script from the DB2 Command Window or Linux shell, use the following command:

```
db2 -t -v -f script1.db2 -z script1.log
```

- t indicates statements use the default statement termination character (semicolon)
- v indicates verbose mode; which echoes the command being executed
- f indicates the following filename contains the SQL statements
- z indicates the following message filename will store the output

To execute it from the command window or Linux shell issue the following command:

```
db2 -t -v -f script1.db2 -z script1.log
```

We first type “db2” to invoke the DB2 command line processor which will take these options as we will describe. The -t means means that there will be a terminator and it will be the default statement terminator (the semicolon).

The -v stands for “verbose”. It means we want to echo the statement being executed before providing the output.

The -f followed by a filename (in this case, “script1.db2”) means that the statements to execute are coming from file script1.db2.

The -z followed by a filename (in this case, “script1.log”) means that all the output will be logged in that file. It is a good idea to delete these message files before the execution of DB2 scripts so that output from a previous script execution is not mixed with output from the current script execution

Using a different statement termination character

```
CREATE TABLE  
arfchong.mytable  
  
  ( col1 INTEGER NOT NULL,  
    col2 VARCHAR(40),  
    col3 DECIMAL(9,2)  
  );
```

Here we show situations where you may need to change the statement terminator. For example, in this case have the same table we had created in the previous script. The little red square highlights the statement terminator which is a semi-colon. Now the little ellipses highlight the commas that are part of the create table syntax. If I were to specify, as I will show you later that a comma be used as a statement terminator then DB2 would get confused and think that the statement would end right where it finds the first comma, right here after the null. So using the comma in this example would be a bad choice for a statement terminator.

Using a different statement termination character

procs.db2

```
CONNECT TO SAMPLE!  
CREATE PROCEDURE P1( )  
BEGIN  
    DECLARE X INT;  
    SET X = 1;  
END!
```

- Execute as follows:

db2 -td! -v -f procs.db2 -z procs.log

On this other slide we have another example this time using the create procedure statement. As part of the syntax, each statement or each line ends with a semi-colon so this poses a problem for me if I were to use the default statement terminator which is the semi-colon in this case.

Therefore, in this example I must change the terminator and I chose to use the exclamation mark.

Now to execute this script, I need to tell DB2 I'm using a different terminator

and this is done with a -t flag followed by -d flag (delimiter) followed by the terminator character I'm using, as I show here. so I'm going to type

db2 -td! .. and the rest is the same as the previous example.

Operating system scripts

- **Pros:**

- Greater flexibility
- Additional logic possibilities
- Supports parameters/variables

- **Cons:**

- Platform-dependent

As the name implies, these are scripts from the operating system therefore they not really related to DB2 and I will not explain how to create an operating system in detail.

What I will show you is how you can combine an SQL script with an operating system script.

Operating system scripts are very flexible and allow you to use parameters.

Operating system scripts are platform independent. That means that Windows and Linux have different script syntax.

A Simple Operating System (Shell) Script

create_database.bat

```
set DBPATH=c:\
set DBNAME=MYDB
set MEMORY=25
db2 CREATE DATABASE %DBNAME% ON %DBPATH% AUTOCONFIGURE USING
MEM_PERCENT %MEMORY% APPLY DB AND DBM
db2 CONNECT TO %DBNAME% USER %1 USING %2
del schema.log triggers.log app_objects.log
db2 set schema user1
db2 -t -v -f schema.db2 -z schema.log
db2 -td@ -v -f triggers.db2 -z triggers.log
db2 -td@ -v -f functions.db2 -z functions.log
```

Execute as follows:

```
create_database.bat myuserid mypassword
```

63

© 2011 IBM Corporation

For example, let me show you these Windows operating system scripts called 'create database.bat'. Since it's Windows, you need to use the .bat extension to identify it as an operating system script. For Linux or Unix you just need to chmod + x to allow for execution of the file; so basically, change permissions to make it executable. You also need to change the way parameters are identified so they use "\$" instead of "%"

Inside the file we're using several parameters that we are highlighting.

So we have DBPATH and we're using it later on in this statement. We have another parameter called DBNAME which has been assigned to MYDB.

As you can see anything that has db2 here is invoking either a DB2 Command or DB2 Script. So this 'db2 CREATE DATABASE' is invoking a DB2 command and passing its parameters. So within this operating system script we're also invoking SQL scripts.

We have these other parameters called memory we use it again in the CONNECT statement. We are passing the user ID as a parameter %1

So when you execute this script you will invoke it as 'create database.bat' and then you put your user ID which would be replacing this '%1' and then you have using %2, so that %2 would be matching whatever you put for the password.

Agenda

- DB2 server editions, clients and drivers
- DB2 Express-C overview
- The DB2 Command Line Processor (demo)
- The DB2 Environment
- Configuring DB2
- Working with scripts
- **Connecting to a DB2 server**
- Data movement utilities



Directories

- **System database directory**

list db directory

- **Local database directory**

list db directory on <drive/path>

- **Node directory**

list node directory

- **DCS directory**

list dcs directory

DB2 directories are binary files where we store information about databases that we can connect to. There are four directories:

1) The System Database Directory

Think of this as the table of contents of a book. It will list all the databases you can connect to whether they are local or remote.

2) The Local Database Directory

Lists the databases that are local, that is, on the same server where you are issuing the "list db" command.

3) The Node Directory

This is the directory where you store connectivity information such as the port number and IP address if you are using TCPIP protocol.

4) The DCS directory

This is not covered in this course, but it would appear if the "DB2 Connect" software is installed, which allows you to connect to servers on the mainframe (DB2 for z/OS) or midrange a.k.a AS/400 (DB2 for i5/OS)

Requirements to connect to a DB2 server

I) Local connections

System DB directory is automatically populated with the info to connect when issuing CREATE DATABASE command

II) Remote connections

Setup required at the server:

- 1) Turn on TCPIP listeners
- 2) Specify DB2 instance port

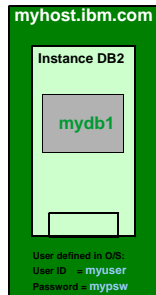
Setup required at the client:

- 1) Create entry in client's node directory
- 2) Create entry in client's system db directory

If the connection is local, that is the DB2 client and DB2 server are on the same system then normally there is no setup required, the information required to connect is added to the system local DB2 directories at the time the Create Database command was executed.

If the connection is remote then you need to perform some setup at the server and some setup up at the client. At the server you need to turn on the TCPIP listeners and specify the DB2 instance port they listen to. At the client you need to add entries to the node and system directories

Remote connections - Setup required at the server

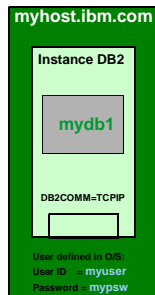


TWO things need to be set up at the server:

Remote connections - Setup required at the server

1) Turn on TCP/IP listeners

db2set DB2COMM=TCPIP



1) DB2COMM

This DB2 registry variable determines which communication protocol listeners should be monitoring requests from clients. Typically TCP/IP is the communication protocol most often used. Changing this parameter requires an instance re-start.

db2set DB2COMM=TCPIP

db2stop

db2start

Remote connections - Setup required at the server

1) Turn on TCP/IP listeners

db2set DB2COMM=TCPIP

2) Specify DB2 instance port

update dbm cfg using svccname 50000

or

update dbm cfg using svccname db2c_DB2

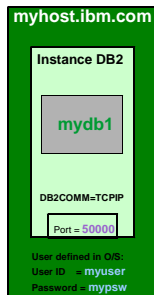
If you use a service name, ensure to update these files at the client:

Linux: /etc/services

Windows: c:\winnt\system32\drivers\etc\services

Example entry in the services file:

db2c_DB2 50000/tcp



2) SVCENAME

This database manager configuration parameter should be set to the service name (as defined in the TCP/IP services file) or to the port number to use when you want to access databases of this instance. Use:

update dbm cfg using svccname <port # or service name>

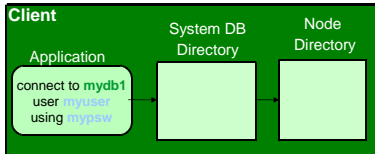
If you use a service name, ensure to update these files:

Linux: /etc/services

Windows: c:\winnt\system32\drivers\etc\services

Example: db2cdb2inst1 50000/tcp

Remote connections - Setup required at the client

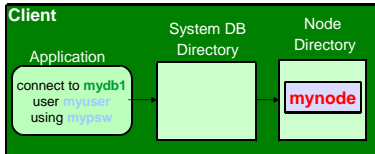


Two commands to run at the client:

At the client, you need to know this information beforehand:

- The name of the database you want to connect to
- The port number of the DB2 instance at the server where the database resides. You can also use a service name, as long as there is a matching entry in the TCP/IP services file
- The operating system user ID and password to connect to the database. This user ID must have been previously defined at the server

Remote connections - Setup required at the client

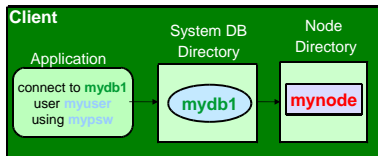


Two commands to run at the client:

```
1) catalog tcpip node mynode  
remote myhost.ibm.com server 50000
```

The first command (catalog tcpip) will put an entry into the Node directory

Remote connections - Setup required at the client



Two commands to run at the client:

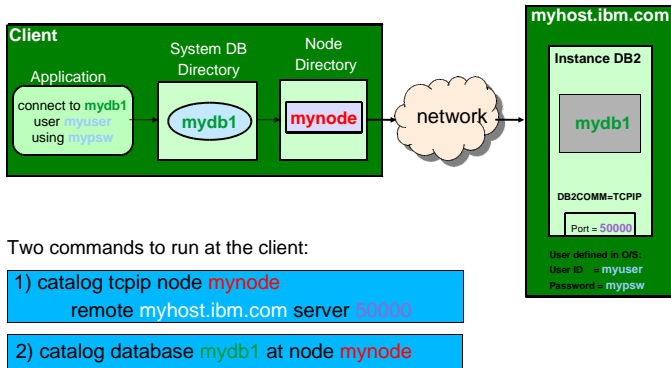
```
1) catalog tcpip node mynode
   remote myhost.ibm.com server 50000
```

```
2) catalog database mydb1 at node mynode
```

The second command (catalog db) will put an entry into the System Database Directory and will be related to the node directory based on the node name.

Once set up, to connect from an application, provide the userID and password of a user at the server where you want to connect to.

Putting both setup together – Client and server



This shows the whole picture put together with the setup required at the client and the server

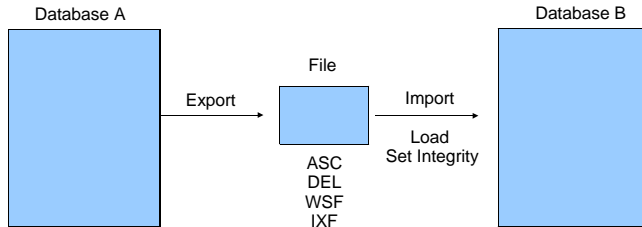
Agenda

- DB2 server editions, clients and drivers
- DB2 Express-C overview
- The DB2 Command Line Processor (demo)
- The DB2 Environment
- Configuring DB2
- Working with scripts
- Connecting to a DB2 server



- **Data movement utilities**

Data movement utilities



This figure provides an overview of some DB2 data movement utilities. On the left side we have database A and on the right side we have database B. Use the EXPORT utility to export rows from a table in database A into a file. The file can have any of these formats: ASC stands for ASCII, DEL stands for delimited ASCII (Basically ASCII format, but we use a delimiter character); WSF stands for Work-Sheet Format (This is normally used for software such as Excel or Lotus-1-2-3); IXF stands for IBM eXchange Format. This format is specific to IBM, and is useful because it includes the DDL or structure of the table as part of the file. The other formats do not include the DDL, but they are more standard.

Now, once you have stored the table rows in a file with any of these formats, you can use either the IMPORT or LOAD utility to load it into a table in the same or another database, in this case, database B. If you are using format IXF, there is no need to create a table ahead of time because with IXF, the table will be created since the DDL is exported in the file, the input file. But if you are using another format, like ASCII, delimited ASCII or WSF then you do need to create manually the table on the target location ahead of time. We talked about IMPORT, but we also have another utility which is LOAD. A LOAD is the same as IMPORT, but it's faster. IMPORT uses SQL INSERT statements behind the scene, therefore these statements are handled by the DB2 engine which will be activating triggers, checking for constraints, and logging changes. Once this is done, it will insert the data into pages on the disk.

LOAD on the other hand, does not go through the DB2 engine but directly stores data into the data pages. This is a lot faster, but it means that triggers are not activated, and constraints are not checked. Therefore there can be integrity issues with the data loaded. Using LOAD would be ideal when you already know the data has integrity and there's no need to have the database check for this. You also can run the SET INTEGRITY command after a LOAD to check for integrity in your data.

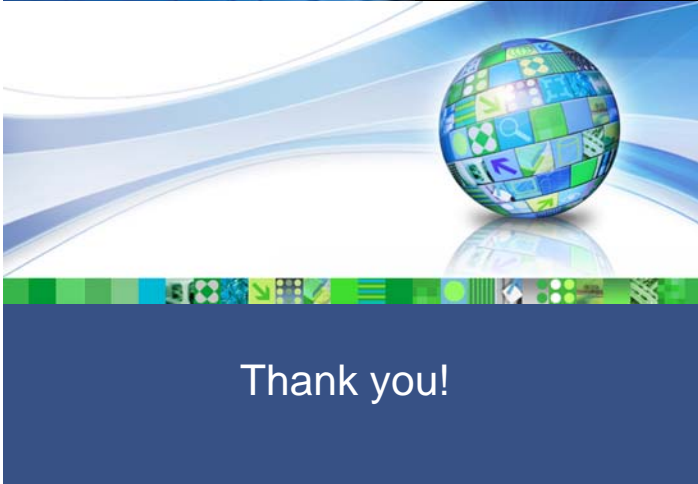
Data movement utilities

- EXPORT, IMPORT, LOAD
- db2move
 - Easily export/import/load/copy set of tables & data (IXF)
- db2look
 - Extracts:
 - DDL
 - Permissions
 - Database statistics
 - Table space characteristics

Export, import and LOAD are utilities that work at the table level; so they work per table.

If you want to execute these utilities on a bunch of tables, you could either create a script, or you can use a utility called db2move. With db2move, you can specify a given schema, so that you could move or export a given number of tables based on the schema, and then import them to another database. You can use db2move with export, import and load, and it only works with the IXF format.

We also have another tool, called db2look. db2look is used to extract information like the DDL, permissions, database statistics and so on. It basically allows me to clone the structure of a database.



Use the forum in the db2university.com course AA001EN if you have technical questions about the materials covered in this course. Fellow students, faculty and IBMers can help you!