An Introduction to DB2 for OS/390 and z/OS System Structures for DB2 Distributed Users

Raul Chong

July 18, 2002

If you know DB2 for Linux, UNIX, and Windows, you can leverage those skills to learn DB2 on OS/390 and z/OS. Using analogies to DB2 distributed concepts, author Raul Chong explains the basics of system administration on DB2 for OS/390 and z/OS, including subsystems, logging, configuration parameters, and more.

© 2002 International Business Machines Corporation. All rights reserved.

Editor's Note: The following article is an abbreviated version of the author's longer article, Getting Started with DB2® for z/OS™ and OS/390™ Version 7 for DB2 Distributed Platform Users (PDF, 333 KB), that discusses application development, tools, locking and concurrency, and much more.

Introduction

If you understand DB2 for UNIX®, Linux and Windows®, it is not difficult for you to leverage that knowledge in understanding DB2 for OS/390™ and z/OS™. In this article, I compare system structures on DB2 for OS/390 and z/OS with structures you probably already understand on DB2 for Linux, UNIX, and Windows.

If you are interested in learning more, an expanded version of this article called Getting Started with DB2 for z/OS and OS/390 Version 7 for DB2 Distributed Platform Users provides much more detail, including information about application development, tools, locking and concurrency, and much more.

In this article, I use DB2 ULWO to refer to DB2 Universal Database Version 7.2 for UNIX, Linux, Windows and OS/2®. I use DB2 S/390® to refer to DB2 Universal Database for z/OS and OS/390 Version 7.

Figure 1 and Figure 2 show a simplified view of the DB2 ULWO system structure and the DB2 S/390 system structure, respectively.
Instances and subsystems

In DB2 ULW0, an instance provides an independent environment where database objects can be created and applications can be run against them. When an instance is created, links to the DB2 code are generated. Several instances can be created in the same machine. In Windows platforms, you can install only one version of DB2 at a given fixpack level. Thus, all instances
created in DB2 UDB for Windows will be linked to the same DB2 code. In UNIX platforms, you can install several versions of DB2 in the same machine because they are installed in different paths; however, only one fixpack level per version is allowed. Thus, in DB2 UDB for UNIX, you may have several instances linking to different code sets.

In DB2 S/390, a DB2 subsystem provides a separate DB2 environment similar to a DB2 ULWO instance. Several DB2 S/390 subsystems can be installed in the same machine logical partition (LPAR), and they can only communicate with each other through the Distributed Data Facility (DDF). (Data sharing provides another way that DB2 subsystems work together, but I don't cover that topic in this article.)

You can install different DB2 S/390 subsystems at different versions in the same LPAR. You can also have different DB2 S/390 subsystems at the same version but with different maintenance levels installed in the same LPAR. In both of these cases, different code sets are used. For example, you might have DB2 S/390 V6 at maintenance level 0112, DB2 S/390 V7 at maintenance level 0106, and DB2 S/390 V7 at maintenance level 0110 installed in the same machine LPAR. DB2 subsystems running with the same version and at the same maintenance level in a LPAR, are also allowed; in this case the DB2 code is shared.

**Naming**

In DB2 ULWO names you may need are:

- **An instance name**
  An instance name uniquely identifies an instance in your machine. As described in Directing commands, you use the instance name to direct commands to a particular DB2 instance.

- **One or many database names**
  Unlike DB2 S/390, a client connects to a particular database rather than a subsystem.

When connecting to a database using TCP/IP, you will also need the TCP/IP address and port for the instance. Other network protocols will need other information.

In DB2 S/390, several names are used to identify a subsystem:

- **Subsystem ID (ssid)**
  Because DB2 is a subsystem of MVS, it will have a subsystem name or subsystem ID (ssid) that MVS can use to identify it. The default ssid is DSN1. The command prefix identifies the particular DB2 subsystem that you connect to. You do not connect to a particular database. You are automatically able to access any database in DB2 to which you are authorized.

- **The location name**
  This is the name specified when using the CATALOG DCS DATABASE command from a DB2 Connect machine, which refers to it as the target database name. It can have from 1 to 16 characters.

- **The LU name**
  This is the name by which VTAM can recognize the local subsystem. The unique name must be unique within the network of connected systems and can have from one to eight characters.
When connecting to a DB2 S/390 subsystem from a DB2 ULWO or DB2 S/390 client using TCP/IP, the TCP/IP address, the port for the subsystem and the location name are needed.

**Directing commands**

In DB2 ULWO, you can direct commands to a specific instance by setting the value of the DB2INSTANCE variable (set DB2INSTANCE=<instance name>) or "attaching" to the instance using a node previously defined (attach to <nodename>). In DB2 S/390 you can execute DB2 commands from different places. The START DB2 command is the only command that is normally executed from the MVS console. Because you can have different DB2 subsystems installed, a command prefix is required for MVS to know to which DB2 should this command be applied to. This command prefix can have up to eight characters. The default is -DSN1, in which DSN1 is the default MVS subsystem name for DB2.

As an example of using a command prefix, if you want to start up a particular DB2 V7 subsystem and that subsystem has been assigned the command prefix of ";", use the command #START DB2 from the MVS console. If you would like to start DB2 S/390 V6 subsystem that has a command prefix of ";", execute the command -START DB2 from the MVS console. Each subsystem is running on its own address space, and can run concurrently.

The above command prefix is only required when performing DB2 commands from a place where you can access different DB2 subsystems (like the MVS console). If you are performing a DB2 command from applications that attach to only one subsystem at a time, you do not need to use specific prefixes, other than the hyphen (-), which should always be used in those cases.

**Connecting a client**

In DB2 ULWO, you attach to an instance to perform some administrative operations, and you connect to a database to perform database operations. In DB2 S/390 you connect to a subsystem, and perform both, administrative as well as database operations. Thus, a DB2 ULWO client connecting to DB2 S/390 (through DB2 Connect software and DDF) will not connect to a specific database, but to the entire DB2 S/390 subsystem. In DB2 ULWO you must connect to a specific database which has its own catalog. For DB2 S/390, a database does not have its own catalog, but there is one catalog for the entire DB2 subsystem; therefore, when you want to access a specific DB2 S/390 database, you actually connect to the entire DB2 subsystem.

**Databases and system table spaces**

This section contains information about databases and temporary work spaces. In addition, it briefly describes the different uses for directories.

**Databases**

In DB2 ULWO, an instance can have several databases. Each database is one closed and independent unit containing its own logs, catalog and database configuration files. You cannot perform queries that would involve tables of two different databases (unless you use Relational Connect or DataJoiner® which are out of the scope of this document). The catalog (SYSCATSPACE), temporary space (TEMPSPACE1) and the user space (USERSPACE1) are all table spaces. These are created automatically when you issue a CREATE DATABASE command.
In DB2 S/390, a DB2 subsystem can contain several databases. Databases in this subsystem interact with each other. In fact, as you can see from Figure 2, the catalog itself (DSNDB06) is a database. The work file database (DSNDB07) would correspond to the temporary space used in DB2 ULWO. The default database (DSNDB04) is used to store objects that users create without explicitly indicating the database the objects belong to. DSNDB04 would correspond to USERSPACE1 in DB2 ULWO. The catalog, directory, and other system structures are created once at DB2 subsystem installation time, not like in DB2 ULWO where a catalog and other system structures are created for every CREATE DATABASE that is executed.

The DB2 ULWO catalog tables use the schema SYSIBM. In addition, DB2 ULWO provides read-only catalog views, which use the schema SYSCAT. Updatable catalog views are in the schema SYSSTATS. DB2 S/390 catalog tables use the qualifier of SYSIBM.

Some of the DB2 S/390 catalog tables are updatable, including some of columns that hold statistics information about the data.

Under the DB2 S/390 structure, you can perform SQL operations using tables from different databases. For example, say you have table TS56692.testtbl in database MYDB1 and table DSN8710.emp in the default database DSNDB04. Then, you can execute the following query:

```sql
SELECT B.name, B.salary
FROM  TS56692.testtbl A, DSN8710.emp B
WHERE A.id = B.edlevel
```

Without federation, this query would not work in DB2 ULWO because the two tables are in two different databases.

**Temporary table spaces**

DB2 ULWO has two types of temporary table spaces: system and user. You must always have a system temporary table space available, because this is the work area for the database manager to perform operations like joins and sorts. User temporary table space, on the other hand, are used to store declared global temporary tables. These tables are not persistent, they only "live" for a given connection or while the application that declared them is running.

Similarly, DB2 S/390 provides two types of temporary space. The work file database (DSNDB07 is the default name) corresponds to DB2 ULWO's system temporary table space. DB2 S/390's TEMP database would correspond to DB2 ULWO's user temporary table space. The TEMP database is used also for server side scrollable cursors, so client applications using this type of cursors may get an error if such a database has not been created ahead of time. The concept of global temporary table is the same in these platforms.

**Directories**

DB2 ULWO has a database directory, a node directory, and a dcs directory. These directories contain connectivity information and perform a similar function to DB2 S/390's communication database (CDB). Don't confuse this use of the term directory with the use of the term in DB2
S/390. The DB2 S/390 directory contains vital information about the subsystem and it is stored in an internal format. It is not intended to be used by regular end users.

Logging

DB2 ULWO uses active and archive logs for recovery purposes. It keeps track of its logs using the file SQLLOGCTL.LFH, which is stored in the same directory as the database files (<instance name>.NODE0000.SQL000x). Version 7 allows dual logging capability. DB2 S/390 also has active and archive logs. A data set called the bootstrap data set (BSDS) is used to keep track of its logs. Dual logging has been available in DB2 S/390 for many versions of the product.

Data caching and page sizes

DB2 ULWO uses buffer pools to improve the performance of a database. In DB2 ULWO, use the command CREATE_BUFFERPOOL to create a new buffer pool. In DB2 S/390, there are predefined buffer pools, most of them starting with a size of zero. In order to "create" a buffer pool, you have to use the ALTER BUFFERPOOL command and set a size greater than zero.

In DB2 ULWO, you specify the data page size as part of the CREATE_TABLESPACE statement. A buffer pool with the correct page size needs to be created before creating the table space that uses this page size. In DB2 S/390, there is no parameter in the CREATE_TABLESPACE statement that indicates the page size to be used; however, by specifying the buffer pool for the table space, you implicitly determine the page size.

DB2 ULWO uses Extended Storage (which can be set with the database configuration parameters ESTORE_SEG_SZ and NUM_ESTORE_SEGS) to provide a second level of caching; DB2 S/390 uses hiperpools.

Configuration parameters

DB2 ULWO has parameters at the instance level (dbm cfg) as well as at the database level (db cfg). Changes to instance level parameters currently require that you start and stop DB2. Changes to database level parameters require that all connections be terminated before the changes take place on the next connections. In DB2 S/390, these parameters are often called "zparms" (for the default name of the parameter module, which is DSNZPARM). There is only one set of parameters that would affect the entire DB2 subsystem and its databases. The job DSNTIJUZ is used to specify the desired values for these parameters. When run, this job will assemble and link-edit the DSNZPARM module as well as the application program’s default module DSNHDECP. The assembled zparm module can be specified when starting DB2. If it is not specified, the module with name DSNZPARM will be used. In versions prior to V7, changes to zparms required DB2 S/390 to be recycled (stopped and started) to load the new parameter module into memory. With V7, this is still the case for some parameters but not for all. The new SET SYSPARM command allows you to load a new parameter module without recycling DB2.

Governing queries

DB2 ULWO uses the governor to monitor and limit the activity of applications against a given database. A configuration file with rules is provided, and the db2gov command is used to start the
governor. Similarly, DB2 S/390 uses the Resource Limit Facility (RLF). This facility is started with the `START RLIMIT` command, and the rules are stored in database DSNRLST.

**Summary**

Table 1, below, summarizes and compares the system structure concepts described in this article.

### Table 1. Comparing DB2 S/390 and DB2 ULWO system structures

<table>
<thead>
<tr>
<th>DB2 S/390 Concept</th>
<th>DB2 ULWO Analogy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog database (DSNB06)</td>
<td>SYSCATSPACE tablespace</td>
</tr>
<tr>
<td>Directory database (DSNB01)</td>
<td>N/A</td>
</tr>
<tr>
<td>Communications database (CDB), part of the catalog</td>
<td>Database directory, node Directory, DCS directory</td>
</tr>
<tr>
<td>Active and archive logs concept</td>
<td>Similar concept as in DB2 S/390</td>
</tr>
<tr>
<td>Dual logging supported</td>
<td>Dual logging supported</td>
</tr>
<tr>
<td>Bootstrap dataset (BDS)</td>
<td>SQLOGCTL.LFH</td>
</tr>
<tr>
<td>Predefined buffer pools are “created” with -ALTER BUFFERPOOL</td>
<td>Buffer pools are created with CREATE BUFFERPOOL</td>
</tr>
<tr>
<td>Hiperpools</td>
<td>Extended Storage (ESTORE)</td>
</tr>
<tr>
<td>Resource Limit Facility (DSNRLST)</td>
<td>- The DB2 Governor (db2gov)</td>
</tr>
<tr>
<td></td>
<td>- Query Patroller</td>
</tr>
<tr>
<td>Work file database (DSNB07)</td>
<td>TEMPSPACE1 tablespace (system temporary tablespace)</td>
</tr>
<tr>
<td>TEMP database, for global temporary tables</td>
<td>User temporary tablespace, for global temporary tables</td>
</tr>
<tr>
<td>Default database (DSNB04)</td>
<td>USERSPACE1 tablespace</td>
</tr>
<tr>
<td>Can execute queries involving tables of different databases</td>
<td>Cannot execute queries involving tables of different databases (without federation)</td>
</tr>
<tr>
<td>Client connects to a DB2 subsystem, not to a particular database</td>
<td>Client connects to a database</td>
</tr>
<tr>
<td>DSNZPARM (SET SYSPARM command allows DSNZPARM module to be loaded in memory while DB2 is up, but for some parameters, a -stop db2, -start db2 is still required)</td>
<td>DBM CFG (db2stop, db2start required for new values to be in effect) and DB CFG (all connection need to be terminated for the new values to be in effect on next connection)</td>
</tr>
</tbody>
</table>

© Copyright IBM Corporation 2002

**Trademarks**
(www.ibm.com/developerworks/ibm/trademarks/)