

March 2004



**DB2** Information Management Software

## **Fastpath to installing DB2 UDB on Linux Clusters**

*By  
Rav Ahuja  
DB2 for Linux Solutions Center  
IBM Software Group*

---

CONTENTS	
<b>1. OVERVIEW</b>	2
<b>2. SUPPORTED ENVIRONMENT</b>	2
<b>2.1 Software Requirements</b>	2
<b>2.2 Hardware Requirements</b>	3
<b>3. INSTALLATION PROCEDURE</b>	4
<b>4. AUTOMATING THE INSTALL</b>	5
<b>4.1 Preparation</b>	5
<b>4.2 Partitioned Instance Install</b>	7
<b>4.3 Verification</b>	10
<b>5. SUMMARY</b>	11
<b>APPENDIX A</b>	12

## 1. OVERVIEW

Linux clusters are well known for their excellent price/performance benefits and IBM® has demonstrated innovative leadership in exploiting clustered environments for database implementations. It therefore comes as no surprise that deployments of IBM DB2® Universal Database™ (DB2 UDB) on Linux are enjoying widespread popularity.

DB2 UDB was the first commercial database on Linux to provide built-in capabilities for deploying across a cluster of servers. That is, DB2 UDB can be deployed across a Linux cluster right out of the box, without the need for additional clustering software or hardware. DB2 UDB on Linux clusters are ideal for running warehouses that involve large volumes of data and demanding transactional workloads. They provide the following benefits:

- *Faster processing time*
- *Very large databases*
- *Excellent scalability*
- *Increased availability*
- *Reduced cost*

DB2 UDB exploits Linux clusters using shared-nothing partitioned database technology which allows a single database to scale up to 1000 nodes! However deployment and management of DB2 UDB across a cluster of servers (even a few hundred or 1000 servers) does not have to be much more complex than doing so on a single machine. This paper provides an illustrated example of how to automate the installation of DB2 UDB across multiple servers for the purpose of creating partitioned databases.

## 2. SUPPORTED ENVIRONMENT

### 2.1 Software Requirements

In order to create partitioned databases that can span multiple servers you require the Enterprise Server Edition (ESE) of DB2 UDB. Although ESE contains all the code for creating partitioned databases, for production environments you are also required to purchase the Data Partitioning Feature (DPF) licensing option.

You can get an evaluation version of DB2 UDB ESE from:

**ibm.com/db2/linux** (click on downloads in the left column)

DB2 UDB is validated to run on a number of Linux distributions. For a complete list, please visit the validation web site at:

**ibm.com/db2/linux/validate**

Only the specified “stock” or errata kernels that come with the DB2 validated distribution or its service release, without any custom patches or modifications, are supported. No additional software or clustered file systems are needed as DB2 UDB has built in capabilities for distributing and coordinating data between nodes using standard protocols such as tcp/ip for inter-node communications. However you need to ensure that some of the packages that come with Linux distributions are installed and activated: rsh (or ssh), pdksh, nfs server (on primary node) and nfs clients (on all other nodes).

## 2.2 Hardware Requirements

Partitioned databases for DB2 UDB are supported on a variety of hardware and Linux platforms - from the very popular 32-bit x86 based systems (running Intel Pentium/Xeon or AMD Athlon processors) to 64-bit platforms such as AMD64 (AMD Opteron) and IA64 (Intel Itanium 2). They are also supported on IBM POWER systems running Linux – eServer pSeries and eServer iSeries. You may run on any type of hardware and model on these platforms, from uniprocessor systems, 2P 1U servers, blade servers, and SMP machines, as long as you are running one of the DB2 validated Linux distributions that supports the underlying hardware. The DB2 Integrated Cluster Environment (DB2 ICE) is an offering that includes pre-tested hardware from IBM eServer, which takes the guesswork out of DB2 UDB hardware compatibility.

Although you do not need identical hardware across all nodes of a partitioned database it needs to be the same platform. For example you could have a partitioned DB2 database across different kinds of Xeon based servers. However you should run the same versions and service levels of software (i.e. operating system and DB2) across all machines in a DB2 cluster.

All of the nodes in the DB2 cluster need to be networked together and should be accessible from all other machines in the cluster using tcp/ip (i.e. you should be able to ping the hostname of each machine).

For test and development environments you could use 10/100 Mbps Ethernet however for intensive production environments Gigabit Ethernet or a higher-speed interconnects (e.g. InfiniBand, etc.) are recommended.

DB2 UDB does not require any shared disk infrastructure to run across a cluster. The DB2 UDB code needs to be installed on local disk of each machine in the cluster.

### 3. INSTALLATION PROCEDURE

Step by step instructions for installing DB2 UDB on a Linux cluster for partitioned database environments are provided in Chapter 2 of the RedBook: *Up and Running with DB2 for Linux* ([ibm.com/redbooks](http://ibm.com/redbooks)) and in DB2 UDB product documentation (Chapter 10 of *Quick Beginnings for DB2 Servers*). The main steps consist of:

1. *On each machine, create the DB2 users and groups, and install DB2 UDB ESE. A DB2 instance needs to be created on the primary machine during installation*
2. *Export the home directory of the DB2 instance owner from the primary machine using nfs and mount it from other machines*
3. *Update the DB2 node configuration file (db2nodes.cfg)*
4. *Enable the execution of remote commands (rsh for DB2 instance userid)*
5. *Enable communications between DB2 partition servers (reserving ports in /etc/services)*
6. *Verify the installation of the partitioned database environment*

As you can notice, the above procedure requires a number of similar steps to be performed repeatedly on each of the machines in the cluster. Instead of doing each of these steps manually on every machine, you can speed up the process significantly by automating these steps on a single machine and having them run on other machines using remote calls.

## 4. AUTOMATING THE INSTALL

This section outlines how to automate the steps listed in the previous section. It refers to the code in **DB2ClusterInstall.tar** downloaded along with this file from [ibm.com/db2/linux/papers](http://ibm.com/db2/linux/papers). This tar volume contains sample scripts used for illustrating the automated installation of DB2 on a Linux cluster. The code and contents of these scripts are for illustration purposes and are provided on an as-is basis without any warranty or support. You may create your own customized scripts or tools suited to your particular environment.

The tar package **DB2ClusterInstall.tar** contains scripts that automate the installation by performing remote root administration on all nodes in the cluster. The scripts accept two ways to execute remote commands as root: Remote Shell (RSH) and Secure Shell (SSH). However, setting up RSH for primary node's root user on all machines may not be desirable as it exposes security vulnerabilities. If enabling RSH for root is a concern for your environment, you can choose to use SSH instead. Either way there are different preparation steps, and are described below.

**Note:** At this point, ensure each machine in the cluster is able to ping all others using hostnames, including itself. This step is required for a successful instance creation. If any machine is already installed with a previous version or a different edition of DB2, please uninstall it first.

### 4.1 Preparation

1. On the primary node, logon as root, and mount the DB2 UDB ESE CDROM. For faster installation it is recommended that you copy the contents of the DB2 CDROM to the local disk.

e.g. `cp -R /mnt/cdrom /ese`

If you downloaded the DB2 UDB ESE image as a tar or zip file, please extract its contents on your hard disk.

e.g. `tar xvf DB2_V81_ESE_LNX_32_NLV.tar`

The resulting directory should contain files such as "db2setup", "db2\_install" ...etc. Now move the directory to something like "/ese".

e.g. `mv -f <unpacked directory> /ese`

2. Next you need to enable RSH service on all nodes for inter-node db2 communication. Per default the rsh service is disabled. Please check with your Linux distribution documentation or section 2.2.3 of the IBM redbook "Up and Running with DB2 for Linux" on how to switch it on.
3. To allow automated installation across the cluster using a script running on a single machine, the ability for root to issue remote commands on other servers is required. For this you need to either enable rsh or ssh for root.

If you choose to use **RSH**:

Perform the following steps on **all** nodes in the cluster:

- i. Logon as root.
- ii. Create the text file `.rhosts` in root's home directory (`/root`) with the following line:  
`<node 0> root`  
where, `<node 0>` is the resolved host name of the primary node.
- iii. Change the permission of this file to 600 by running  
`chmod 600 .rhosts`
- iv. Verify that root from the primary server can issue commands on all other servers by trying the following:  
`rsh <hostname of other node> ls`

Now proceed to section 4.2.

If you choose to use **SSH**:

Per default SSH is enabled upon its installation. If SSH is disabled, please refer to your Linux manual for information on how to enable it. After enabling the SSH daemon on all machines, do the following:

On the primary node, as root run the following command to generate a RSA key pair stored in two files, `id_rsa` and `id_rsa.pub`, in `/root/.ssh/` as follows, simply press enter at all prompts:

```
# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
d0:97:4f:8c:79:3e:18:e8:08:da:6c:41:62:a1:fd:cd root@panda01
```

On all other nodes, as root copy the content of “id\_rsa.pub” file generated on the primary node to /root/.ssh/authorized\_keys. Create it if it does not already exist. After all these steps are completed, the root user at the primary node should be able to connect to all other nodes without entering a password. To verify, from the primary node issue the following command against all other nodes:

```
ssh <hostname of other node> ls
```

### 4.2 Partitioned instance install and setup

The tar package DB2ClusterInstall.tar contains the script **partInst** and two directories, **ssh** and **rsh**. Both directories contain the script, **installPartitionDB2**. Depending on which remote command mechanism (rsh or ssh) is chosen, the script installs DB2 ESE on all nodes and creates the partitioned instance. To run it, perform the following steps on the primary node:

1. Logon as root and unpack DB2ClusterInstall.tar, for example:

```
tar xvf DB2ClusterInstall.tar
```

You should see one startup script named “partInst”, and two directories, rsh and ssh.

2. In the directory where you extracted the contents of DB2ClusterInstall.tar, execute partInst. For example:

```
./partInst
```

Now you should see a welcome screen prompting you for an installation method. Depending on how you prepared your environment, in section 4.1, you can either choose rsh or ssh.

After selecting one of the methods you will be redirected to the corresponding installation wizard. The wizards for both methods work exactly the same way.

3. At the first prompt, "Location of DB2 UDB V8.1 ESE image in all participating machines", enter /ese (or specify the directory where the DB2 UDB ESE install image is located).

```
-----  
Welcome to DB2 partitioned instance setup wizard for Linux  
Prerequisite:  
1. Place the DB2 UDB V8.x ESE install (licensed or trial) image on  
the primary node.  
2. Enable RSH service/daemon on all machines in the cluster  
Otherwise you will not be able to use DB2 partitioned databases  
3. Allow primary node's root user to execute remote commands against  
root users on all other machines in the cluster using rsh or ssh  
  
*****  
* Please run this wizard on the primary node. *  
*****  
Please select an installation method  
[1] rsh  
[2] ssh  
=====
```

Make your selection [ 1-2 | Press enter to EXIT ]:1

4. At the next prompt, enter the size of the cluster i.e. the number of nodes/severs in the cluster (minimum allowable size is 2).
5. When prompted, indicate a name for the partitioned instance name, (default value is db2ib which is selected by pressing enter without specifying any value).
6. In the subsequent prompts, enter the hostnames of nodes participating in the database cluster. After all hostnames are entered and rsh or ssh settings are verified, the script will install DB2 on each node and create the partitioned instance for you.

A sample session to create a 4-node DB2 cluster (or partitioned instance) is shown in the following snapshots.



```
-----
Welcome to DB2 partitioned instance setup wizard for Linux

*****
* Please run this wizard on the primary node. *
*****

=====
Location of DB2 UDB V8.1 ESE image in the primary node: /ese
Number of machines participating in the cluster? 4
Name of DB2 partitioned instance(At most 8 characters, default: db2ib):

Hostname of node#0(primary node): db2r01
Hostname of node#1                : db2r02
Hostname of node#2                : db2r03
Hostname of node#3                : db2r04

Automatic creation of partitioned instance "db2ib" starting at Thu Mar
4 15:55:09 EST 2004

Step 0 - Enable Network File System(NFS) and share /ese from the primary
node
db2r01: Restart NFS server
Shutting down kernel based NFS server
done
Starting kernel based NFS server
done
db2r01: Export /ese
db2r02: Mount /ese
db2r02: Mount successful!!!
db2r03: Mount /ese
db2r03: Mount successful!!!
db2r04: Mount /ese
db2r04: Mount successful!!!

Step 1 - Install DB2 and create instance owner "db2ib" in group "db2ib"
for other nodes

db2r01: Installing DB2
DBI1191I db2setup is installing and configuring DB2 according to
the response file provided. Please wait.

db2r02: Installing DB2
DBI1191I db2setup is installing and configuring DB2 according to
the response file provided. Please wait.

db2r02: Creating user "db2ib"

db2r03: Installing DB2
DBI1191I db2setup is installing and configuring DB2 according to
the response file provided. Please wait.

db2r03: Creating user "db2ib"
db2r04: Installing DB2
DBI1191I db2setup is installing and configuring DB2 according to
the response file provided. Please wait.

db2r04: Creating user "db2ib"
```

```
Step 2 - Share /home/db2ib from the primary node
db2r01: Export /home/db2ib
db2r02: Mount /home/db2ib
db2r02: Mount successful!!!
db2r03: Mount /home/db2ib
db2r03: Mount successful!!!
db2r04: Mount /home/db2ib
db2r04: Mount successful!!!

Step 3 - Adapt /home/db2ib/sqllib/db2nodes.cfg file

Step 4 - Setup rsh for db2ib

Step 5 - Duplicating DB2 service ports in /etc/services file on all
other nodes

Step 6 - Start the partitioned instance "db2ib"
db2r01: Starting partition #0
03/04/2004 15:56:31      0  0      SQL1063N  DB2START processing was
successful.

db2r02: Starting partition #1
03/04/2004 15:56:34      1  0      SQL1063N  DB2START processing was
successful.

db2r03: Starting partition #2
03/04/2004 15:56:37      2  0      SQL1063N  DB2START processing was
successful.

db2r04: Starting partition #3
03/04/2004 15:56:39      3  0      SQL1063N  DB2START processing was
successful.

Install DB2 and create partitioned instance "db2ib" finished at Thu Mar
4 15:56:39 EST 2004

Time elapsed: 90 seconds
```

### 4.3 Partitioned Database Verification

Once the product is installed, verify that a partitioned database can be created<sup>1</sup>. On any of the nodes login as the DB2 instance userid specified during installation (default value is db2ib). Issue "db2start" and ensure all DB2 partitions are started. Next issue the command "db2samp". This will create the sample database. Connect to the database by issuing "db2 connect to sample". If this works, you have successfully installed DB2 on a Linux cluster and created a sample partitioned database.

<sup>1</sup>See Appendix A for preparation steps before creating the sample database

```
db2r01:~ # su - db2ib
db2ib@db2r01:~> db2sampl
db2ib@db2r01:~> db2 connect to sample

      Database Connection Information

Database server          = DB2/LINUX 8.1.0
SQL authorization ID    = DB2IB
Local database alias    = SAMPLE

db2ib@db2r01:~> db2 "select * from org"

DEPTNUMB DEPTNAME          MANAGER DIVISION      LOCATION
-----
      10 Head Office          160 Corporate    New York
      66 Pacific              270 Western      San Francisco
      15 New England          50 Eastern       Boston
      20 Mid Atlantic         10 Eastern       Washington
      38 South Atlantic       30 Eastern       Atlanta
      42 Great Lakes          100 Midwest      Chicago
      84 Mountain             290 Western      Denver
      51 Plains               140 Midwest      Dallas

8 record(s) selected.
```

## 5. SUMMARY

DB2 UDB running on Linux clusters provides a powerful environment for enhancing performance, availability and handling large volumes of data at a remarkably low cost when compared to traditional solutions on large SMP boxes. Deploying and managing DB2 UDB in a clustered environment can be relatively quick and simple with a little automation, DB2's silent installation features and autonomic management capabilities.

To test drive DB2 UDB for Linux for clusters today, get the Enterprise Server Edition by visiting [ibm.com/db2/linux](http://ibm.com/db2/linux) and click on downloads in the left column.

For further information, send an e-mail to: [DB2forLinux@ca.ibm.com](mailto:DB2forLinux@ca.ibm.com)

## APPENDIX A

DB2 UDB on Linux uses the shared-nothing approach for partitioned databases. That is, each DB2 partition (node) in the cluster owns data on the disk that only it can access directly. Therefore prior to creating a database it is wise to allocate storage for each partition in the cluster. For production environments this needs to be a well thought out process where table spaces and their containers are defined. For the purposes of creating the sample database this process can be simplified by creating a directory (e.g. /db2) for db2 data on each node in the system, and updating the default database path to point to that directory. This is illustrated using the screen capture of a sample session below.

```
db2r01:~ # ssh db2r01 mkdir /db2
db2r01:~ # ssh db2r02 mkdir /db2
db2r01:~ # ssh db2r03 mkdir /db2
db2r01:~ # ssh db2r04 mkdir /db2
db2r01:~ # ssh db2r01 chown db2ib:db2ib /db2
db2r01:~ # ssh db2r02 chown db2ib:db2ib /db2
db2r01:~ # ssh db2r03 chown db2ib:db2ib /db2
db2r01:~ # ssh db2r04 chown db2ib:db2ib /db2

db2r01:~ # su - db2ib
db2ib@db2r01:~> db2 get dbm cfg | grep DFTDBPATH
Default database path          (DFTDBPATH) = /home/db2ib
db2ib@db2r01:~> db2 update dbm cfg using DFTDBPATH /db2
DB20000I The UPDATE DATABASE MANAGER CONFIGURATION
command completed successfully.
db2ib@db2r01:~> db2 terminate
DB20000I The TERMINATE command completed successfully.
db2ib@db2r01:~> db2stop
03/04/2004 16:15:08      0  0  SQL1064N  DB2STOP processing
was successful.
03/04/2004 16:15:50      2  0  SQL1064N  DB2STOP processing
was successful.
03/04/2004 16:15:50      1  0  SQL1064N  DB2STOP processing
was successful.
03/04/2004 16:15:50      3  0  SQL1064N  DB2STOP processing
was successful.
SQL1064N  DB2STOP processing was successful.
db2ib@db2r01:~> db2start
03/04/2004 16:15:59      0  0  SQL1063N  DB2START processing
was successful.
03/04/2004 16:15:59      1  0  SQL1063N  DB2START processing
was successful.
03/04/2004 16:15:59      2  0  SQL1063N  DB2START processing
was successful.
03/04/2004 16:15:59      3  0  SQL1063N  DB2START processing
was successful.
SQL1063N  DB2START processing was successful.

db2ib@db2r01:~> db2sampl
```



© Copyright IBM Corporation, 2003  
IBM Canada Ltd.  
8200 Warden Avenue  
Markham, ON  
L6G 1C7  
Canada

All Rights Reserved.

Neither this documentation nor any part of it may be copied or reproduced in any form or by any means or translated into another language, without the prior consent of the IBM Corporation.

IBM makes no warranties or representations with respect to the content hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. IBM assumes no responsibility for any errors that may appear in this document. The information contained in this document is subject to change without any notice. IBM reserves the right to make any such changes without obligation to notify any person of such revision or changes. IBM makes no commitment to keep the information contained herein up to date.

The information in this document concerning non-IBM products was obtained from the supplier(s) of those products. IBM has not tested such products and cannot confirm the accuracy of the performance, compatibility or any other claims related to non-IBM products. Questions about the capabilities of non-IBM products should be addressed to the supplier(s) of those.

DB2, DB2 Universal Database, IBM, and the IBM logo are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries or both.

Other company, product and service names may be trademarks or service marks of others.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.