# Implementing IBM® DB2® 9.7 Enterprise Server Edition with Microsoft Failover Clustering

March 2012

Jeremy Brumer
Michelle Chiu
Aslam Nomani
Lui Tang

## Notices, trademarks, service marks, and disclaimers

IBM, the IBM logo, DB2, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

The furnishing of this document does not imply giving license to any IBM patents.

References in this document to IBM products, Programs, or Services do not imply that IBM intends to make these available in all countries in which IBM operates.

## Contents

## Abstract

IBM(R) DB2(R) software is the first multimedia, web-ready relational database management system. It is strong enough to meet the demands of large corporations and flexible enough to serve medium-sized and small businesses. DB2 software integrates power for business intelligence, content management, and e-business with industry-leading performance and reliability.

Microsoft Failover Clustering strengthens the DB2 solution by helping to provide a high availability (HA) computing environment. An HA solution reduces the amount of time that an application is unavailable by removing single points of failure. Failover Clustering facilitates the automatic failover of applications and data from one system to another in a cluster after a hardware or software failure. However, an HA setup includes many parts in addition to the Failover Clustering software. A good HA solution also requires planning, design, customization, and change control.

This document takes you through sample configurations using IBM DB2 9.7 for Linux, UNIX, and Windows using Microsoft Windows Server 2008, Enterprise Edition. Note that Failover Clustering on Windows 2008 is supported with the following DB2 products:

- DB2 9.1 Fix Pack 5 and later fix packs (both DPF and non-DPF environments are supported)
- DB2 9.5 Fix Pack 1 and Fix Pack 2 (only non-DPF environments are supported)
- DB2 9.5 Fix Pack 3 and later fix packs (both DPF and non-DPF environments are supported)
- DB2 9.7 general availability and later fix packs (both DPF and non-DPF environments are supported).

This paper is not intended to provide a detailed understanding of Failover Clustering or DB2 software. It is assumed that you are already familiar with both Failover Clustering and DB2 software. The intention of this paper to provide an understanding of how DB2 Enterprise Server Edition integrates into the Failover Clustering environment and how to configure DB2 Enterprise Server Edition within that environment.

## Introduction

DB2 software depends on a core group of resources to ensure successful execution of database applications. If one of these resources were to become unavailable, DB2 would no longer be fully functional. Within a high availability (HA) environment, it is very important to understand the resources that are required and then to plan a strategy to ensure that these resources are continuously available to the application. Clustering software can be very beneficial in an HA environment because it provides a mechanism to ensure that all resources are continuously available to an application. The clustering software can also go one step further and ensure that the application is continuously available.

Failover Clustering provides the ability to fail over resources between multiple machines. These resources include such items as disks, IP addresses, file servers, and network names. DB2 software uses the ability of Failover Clustering to create additional resource types to develop a resource type called DB2 Server. By grouping various resources together by using the Failover Clustering group feature, a virtual machine is created that can float among all nodes in the cluster. Thus, if any resource in the group fails, the entire group (or virtual machine) fails over and restarts on another node.

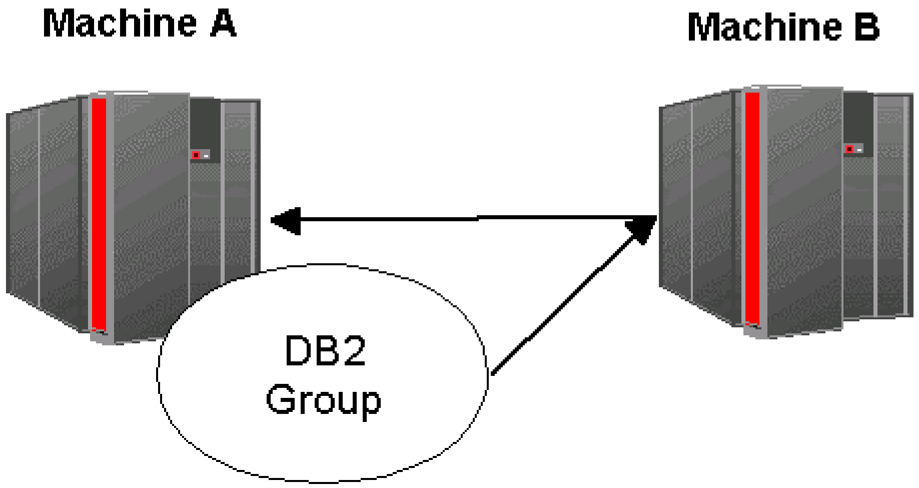


Figure 1: DB2 Server group floating between machines

**Tip:** Within an HA environment, it is important to try to alleviate any single points of failure (such as software applications, disks, networks, and processors) because any system is only as reliable as its weakest link.

## DB2 Enterprise Server Edition with Failover Clustering overview

You can use DB2 Enterprise Server Edition to create single-partition or multiple-partition database environments. In the case of a multiple-partition database environment, DB2 Enterprise Server Edition uses a highly scalable, shared-nothing architecture to spread the data across the database partitions. These partitions can be located on different physical machines. Data can be processed in parallel across partitions and in parallel within each partition.

If a partition fails and a query requires data from that partition, that query fails. DB2 Enterprise Server Edition provides the ability to issue the query against a subset of partitions. However, the query would not reflect the results of the entire data set and thus might not meet requirements for many environments. Thus, the failed partition must be restarted so that queries can access the data on that partition.

DB2 Enterprise Server Edition can use Failover Clustering to allow database partitions to fail over, enabling all partitions and thus all data to be highly available. The following simple example illustrates how DB2 Enterprise Server Edition works within the Failover Clustering environment. In this example, the DB2 instance comprises two partitions, similar to what is shown in Figure 2. Even though both database partitions are connected to the same disk subsystem, each partition accesses only the data that it owns.

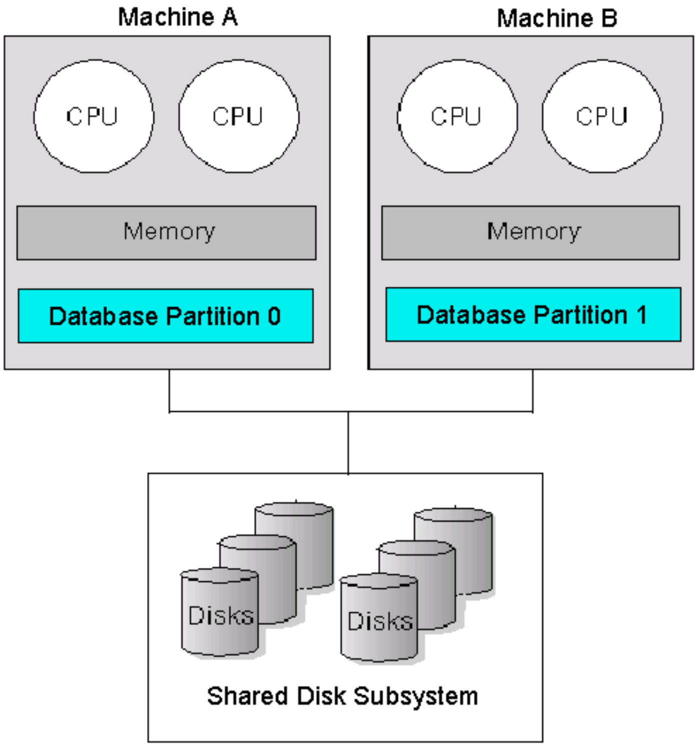


Figure 2. *Typical topology of DB2 Enterprise Server Edition with multiple partitions and Failover Clustering*

**Note:** In a single-partition DB2 Enterprise Server Edition environment, Database Partition 1 would not exist in Figure 2.

Initially, Partition 0 is active on Machine A. Assume that the data for Partition 0 is stored in the shared disk subsystem on Disk E. Also, initially, Partition 1 is active on Machine B. Assume that the data for Partition 1 is stored in the shared disk subsystem on Disk F. The `db2nodes.cfg` file stores the DB2 partition information. The file initially contains the following information:

```
0 macha macha 0 10.1.1.5
1 machb machb 0 10.1.1.6
```

In a single-partition environment, only Database Partition 0 would exist in the `db2nodes.cfg` file. For more information about the `db2nodes.cfg` file, see the DB2 Information Center.

If Machine B fails, Partition 1, Disk F, and TCP/IP address 10.1.1.6 fail over to Machine A, resulting in both Partition 0 and Partition 1 being active on Machine A. Partition 0 still accesses the data on Disk E, and Partition 1 still accesses the data on Disk F. The DB2 software automatically changes the host name and computer name of Partition 1 to macha and changes the port number that is associated with Partition 1 to alleviate any conflicts. The TCP/IP address does not change, because it is a highly available address that moves with the DB2 partition. The DB2 software  also updates the host name, computer name, and port number in the `db2nodes.cfg` file, which now contains the following information:

```
0 macha macha 0 10.1.1.5
1 macha macha 1 10.1.1.6
```

Instance information such as the `db2nodes.cfg` file is stored on a highly available network name, file server, and disk. If the machine with these resources fails, they will fail over to another machine and still be available to the DB2 software.

If remote clients were connected to Partition 0, they might have to reissue any transactions that were uncommitted at the time of the failure. If an uncommitted transaction did not require any information from the failed partition, the transaction would not have to be reissued. If remote clients were connected to Partition 1, they must reconnect to the database before executing any queries. The remote clients reconnect to the same highly available TCP/IP address that they were connected to before and are not aware that Partition 1 has moved to a different machine.

**Note:**  The default behavior of the DB2 client is to connect to the partition that has port 0. Thus, after the failover, a connection to Partition 1 is actually a connection to Partition 0.

## Conceptual overview

Because a Failover Clustering environment allows multiple machines to own the same disk resource, this disk must have shared direct access from all machines in the cluster. The cluster maintains a heartbeat between the nodes to determine which nodes in the cluster are available. In Windows 2003 and earlier, the heartbeat communication usually flows through an internal private network, and remote clients access the cluster through a public network. Thus, a typical cluster topology is as follows:

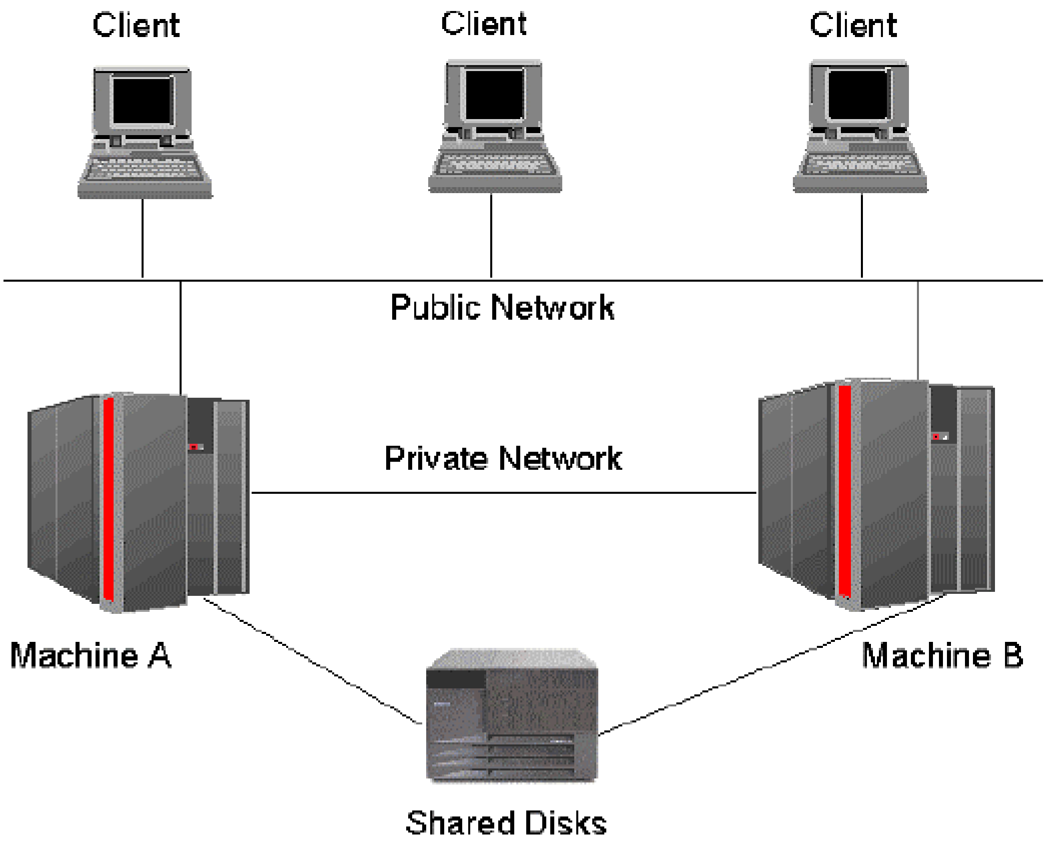
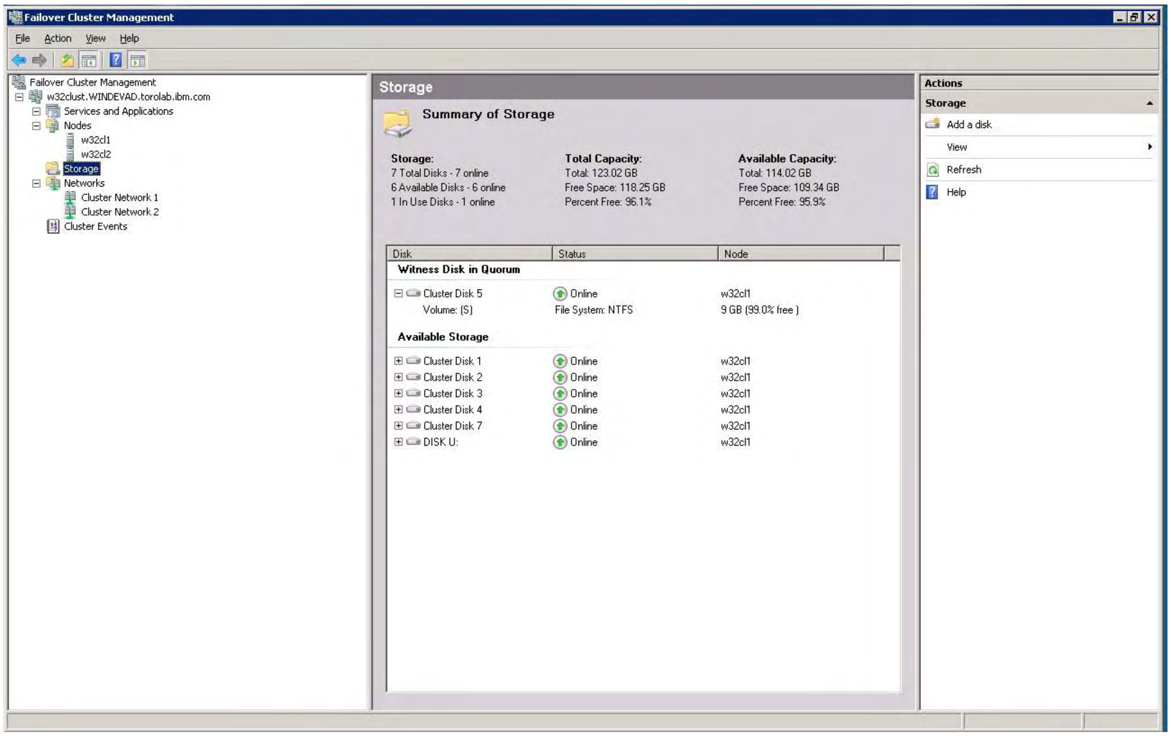


Figure 3. *Typical cluster topology*

In Windows 2008, Microsoft reduced some of the management complexity by removing the public, private, and mixed network roles. If you enable a network for cluster use, heartbeats and internode communication are sent across it. Nonetheless, the previous configuration with two networks is still recommended to prevent the network being a single point of failure. For the purposes of this paper, *private network* refers to a network that is private between the machines in the failover cluster, regardless of the role of the network as displayed by the **`cluster net /prop`** command.

Failover Cluster Management is a graphical administration tool that is part of Failover Clustering. Failover Cluster Management shows the machines in the cluster, the resources, the groups, and the networks that are available to the cluster. In Windows 2003 and earlier, this tool was called Cluster Administrator.



Summary of Storage

| Storage: | Total Capacity: | Available Capacity: |
|---|---|---|
| 7 Total Disks - 7 online | Total: 123.02 GB | Total: 114.02 GB |
| 6 Available Disks - 6 online | Free Space: 118.25 GB | Free Space: 109.34 GB |
| 1 In Use Disks - 1 online | Percent Free: 96.1% | Percent Free: 95.9% |

| Disk | Status | Node |
|---|---|---|
| **Witness Disk in Quorum** | | |
| Cluster Disk 5 | Online | w32cl1 |
| Volume: (S) | File System: NTFS | 9 GB (99.0% free ) |
| **Available Storage** | | |
| Cluster Disk 1 | Online | w32cl1 |
| Cluster Disk 2 | Online | w32cl1 |
| Cluster Disk 3 | Online | w32cl1 |
| Cluster Disk 4 | Online | w32cl1 |
| Cluster Disk 7 | Online | w32cl1 |
| DISK U: | Online | w32cl1 |

Figure 4. *Failover Cluster Management displaying information after creation of a cluster*

The previous screen capture provides an initial snapshot of a cluster immediately after Failover Cluster Management has been installed in a two-node configuration. Windows 2003 Enterprise and Datacenter Edition supported up to eight nodes in a cluster, and Windows 2008 x64 Enterprise and Datacenter Editions support up to 16 nodes in a cluster. If a DB2 Enterprise Server Edition instance must span more nodes than are available in a single cluster, you can use multiple clusters.

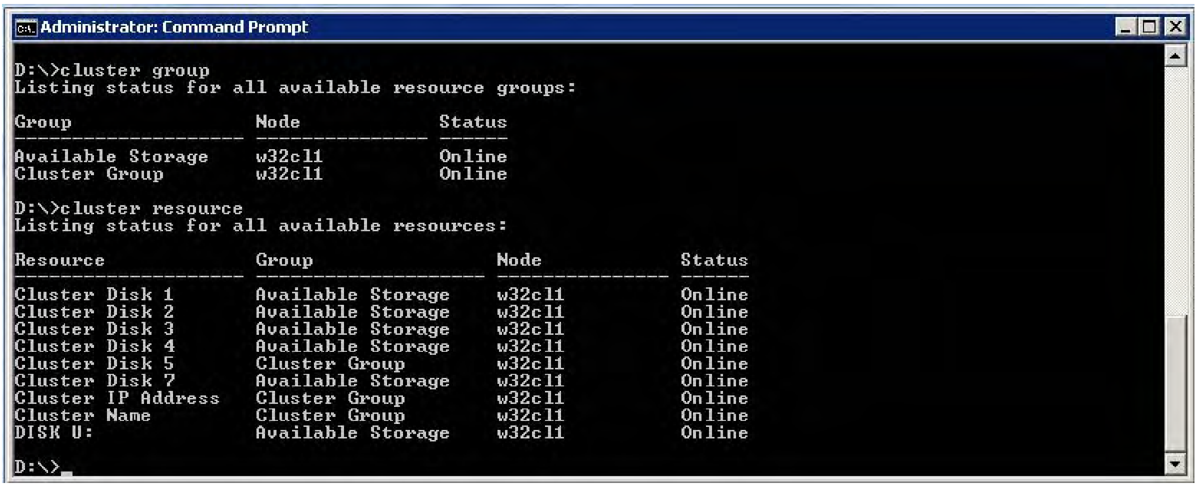
The cluster that is shown in Figure 4 is the starting point for examples used throughout this paper. Here are some of the notable points about the cluster:

- The name of the cluster is w32clust.
- The two machines in the cluster are w32cl1 and w32cl2.
- There are seven disks available to the cluster, and one of them was automatically chosen as the witness disk (Cluster Disk 5).
- Two networks exist, Cluster Network 1 and Cluster Network 2. One of them is a private network, and one is a public network.
- Currently, the cluster disks are available on machine w32cl1.

- By default, there are two cluster groups. The first group, called the Cluster Group, contains the witness disk, the Cluster IP address, and the Cluster Name. The second group, called Available Storage, contains all the other shared disks that are available to the cluster.

You can issue cluster commands such as the following ones from a command prompt to perform cluster operations:

- The **`cluster group`** command displays the available resource groups.
- The **`cluster resource`** command displays the available resources.

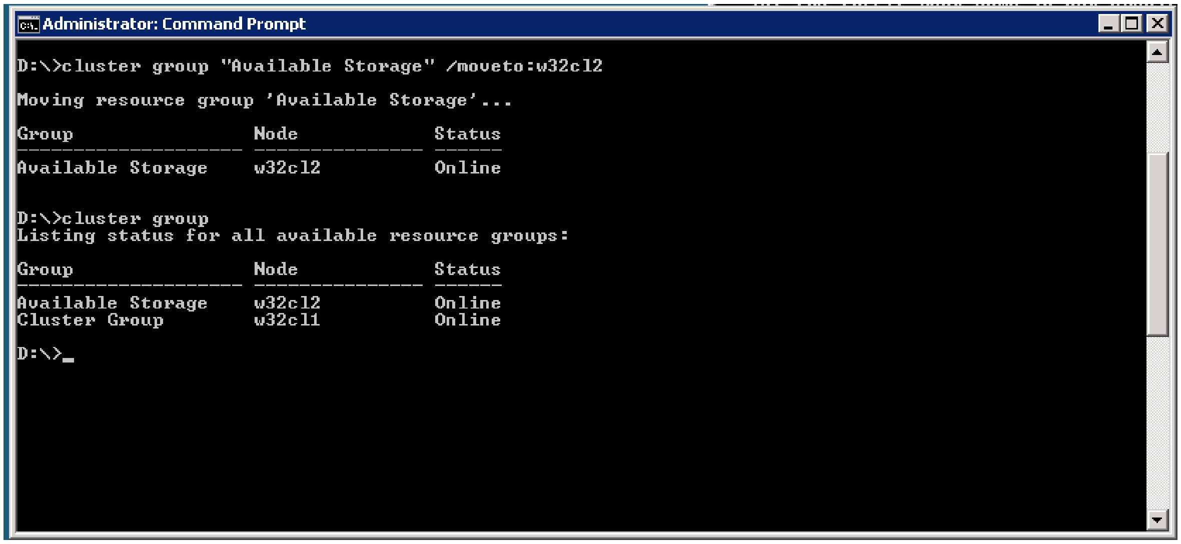


```
D:\>cluster group
Listing status for all available resource groups:

Group                Node            Status
-------------------- --------------- ------
Available Storage    w32cl1          Online
Cluster Group        w32cl1          Online

D:\>cluster resource
Listing status for all available resources:

Resource             Group                Node            Status
-------------------- -------------------- --------------- ------
Cluster Disk 1       Available Storage    w32cl1          Online
Cluster Disk 2       Available Storage    w32cl1          Online
Cluster Disk 3       Available Storage    w32cl1          Online
Cluster Disk 4       Available Storage    w32cl1          Online
Cluster Disk 5       Cluster Group        w32cl1          Online
Cluster Disk 7       Available Storage    w32cl1          Online
Cluster IP Address   Cluster Group        w32cl1          Online
Cluster Name         Cluster Group        w32cl1          Online
DISK U:              Available Storage    w32cl1          Online

D:\>_
```

Figure 5. *The cluster groups after creation of the cluster*

**Tip:** Ensure that all disks within a cluster can fail over successfully between all machines in the cluster before proceeding with configuring DB2 Enterprise Server Edition within the cluster environment. You can fail over a disk as part of a cluster group by issuing the **`cluster group`** command, as shown in the following example:

```
cluster group "Available Storage" /moveto:w32cl2
```



```
D:\>cluster group "Available Storage" /moveto:w32cl2

Moving resource group 'Available Storage'...

Group                Node            Status
-------------------- --------------- ------
Available Storage    w32cl2          Online


D:\>cluster group
Listing status for all available resource groups:

Group                Node            Status
-------------------- --------------- ------
Available Storage    w32cl2          Online
Cluster Group        w32cl1          Online

D:\>_
```

Figure 6. *Moving the Available Storage group*

Among the default resource types that are available to a cluster, there is no resource type that corresponds to a DB2 partition. The DB2 software creates a resource type called DB2 Server when you execute the **`db2mscs`** command, which is covered in a later section (In DB2 Universal Database Version 8, the resource type was called DB2). Each resource of type DB2 Server corresponds to a DB2 partition. Using that resource type, the Failover Clustering environment can monitor the status of the DB2 software, ensuring that it stays online, and monitor all the resources required by that software.

Each DB2 partition requires disks to store information, an IP address for internal communication (for a multiple-partition instance), and optionally an IP address (for remote connections). Therefore, DB2 Enterprise Server Edition uses the group feature within Failover Cluster to group multiple resources into a single logical entity. The combination of a DB2 Server resource, disks, and IP addresses represents almost all the resources that are required to successfully run a DB2 instance. A multiple-partition instance also uses the name of a highly available network and file server resource to store instance information that is available to all partitions. The other resources that are required are processors and memory. These last two resources are obtained from the machine on which the group is currently active and do not fail over between machines.

A single Failover Cluster group can contain a single partition or a grouping of partitions. If partitions are in the same group, they are always located on the same machine at the same time. If you want to have different partitions on different machines at the same time, you place the partitions in different Failover Cluster groups.

As already noted, each Failover Cluster group with DB2 partitions contains one or more DB2 Server resources, disks, and IP addresses. In a multiple-partition instance, the group with the instance-owning partition also contains a network name, a file server, and a file share resource to store configuration information for the entire DB2 instance. The instance-owning partition is always Partition 0. The order in which these resources come online is critical when you bring the group online. If the DB2 Server resource starts first, it might fail because the partition might require access to files on a disk that is not online yet. You can use the dependency feature within Failover Cluster to define which resources must be started before an attempt is made to start another resource. The DB2 Server resource is automatically configured to be dependent on the disks and the IP addresses, along with the network name and file server resource if the DB2 Server resource corresponds to the instance-owning partition. The dependency also ensures that the DB2 Server resource is stopped before any attempt is made to take the disks and IP addresses offline, along with the network name and file server resource if the DB2 Server resource corresponds to the instance-owning partition.

## Failover and failback

Failover Cluster determines whether resources have been restarted on the current machine or whether they should fail over to another machine in the cluster. It is very important that the cluster is aware of which resources have been started so that it knows which resources it must try to keep online. If you bring resources or groups online through Failover Cluster Management, Failover Cluster is aware that these resources and groups have been started and attempts to ensure that they stay available in the case of a failure. If you start the DB2 partition by using a non-cluster interface (the **`db2start`** command, **`NET START`**, or Service Manager, which performs an automatic start), Failover Cluster is not aware that the DB2 partition has been started. Therefore, Failover Cluster does not attempt to keep the DB2 partition up and running.

Failover Cluster monitors all resources and groups that you bring online through Failover Cluster Management or the cluster command-line interface. If a machine in the cluster fails, Failover Cluster moves all resources and groups to another cluster machine and ensures that any resources that were online are brought back online. When a resource fails, Failover Cluster attempts to bring that resource back online on the current machine first. If the resource continues to fail, Failover Cluster moves the whole group that is associated with the resource to another cluster machine and tries to bring the group online. The number of times that Failover Cluster retries bringing the resource online is configurable within Failover Cluster Management. The preferred machine for a group failover is also configurable within Failover Cluster Management.

Failures of the DB2 Server resource can occur because of exceptions within the DB2 software or because an operating system resource has run low. Because failure detection of the DB2 software is triggered by termination of the DB2 process, a hang situation does not automatically trigger a restart of the DB2 Server resource.

When a failover occurs because of a machine failure or other cause, database partitions might be in a transactionally inconsistent state. When the database partitions start on the surviving machine, they must go through a crash recovery phase that might invoke sideways recovery to other partitions to bring the database back to a transactionally consistent state. To maintain data integrity and transactional consistency, the database is not completely available until crash recovery has completed.

In a mutual-takeover environment, it is very important to plan for the highest potential machine requirements in case all Failover Cluster groups are online on a single machine at a particular time. If the machine is not capable of handling the workload, the results can range from performance degradations to further abnormal terminations.

Failback is the ability for a Failover Cluster group to move back to its preferred machine after that machine is back online within the cluster. The failback involves taking the group offline on its current machine, moving the group over to its preferred machine, and then finally bringing the group online on the preferred machine. One of the disadvantages of automatic failback is that every time the group containing the DB2 partition is brought offline, some connections might be forced off the database. Failover Cluster drives the failback based on configurations within Failover Cluster Management, with the default behavior being not to fail back. If you want to use automatic failback, you must manually configure the resource group failback property by using Failover Cluster Management.

During failback, the DB2 group must go offline. To allow DB2 group to go offline, the DB2 connections to that partition must be forced off. If the **`DB2_FALLBACK`** DB2 profile variable is set to `ON`, which is recommended, the DB2 connections are forced off during the offline operation. By default, the **`db2mscs`** utility causes the DB2 software to allow failback and sets the variable to `ON`. If you use the manual method to cluster DB2 partition, and want to allow failback, set the variable by issuing the following command:

```
db2set DB2_FALLBACK=ON
```

To disallow failback, set the **`DB2_FALLBACK`** variable to `OFF`.

## The db2mscs utility

The **db2mscs** utility is a stand-alone command-line utility that you can use to transform a non-HA instance into an HA instance. The utility creates all Failover Cluster groups, resources, and resource dependencies. It also copies all DB2 information that is stored in the Windows registry to the cluster portion of the registry and moves the instance directory to a shared cluster disk.

You must run the **db2mscs** utility from the instance-owning partition. You must stop the instance before running the utility. As input to the utility, provide a configuration file (db2mscs.cfg) specifying how to set up the cluster. The fields in the configuration file that are used for DB2 Enterprise Server Edition are as follows. Examples of the configuration file are shown in subsequent sections of this paper.

**DB2_INSTANCE.** The name of the DB2 instance. If you do not specify the instance name, the default instance name (the value that is specified by the **DB2INSTANCE** environment variable) is used. This parameter has a global scope; you should specify it only once in the db2mscs.cfg file.

**DAS_INSTANCE.** The name of the DB2 administration server instance. This parameter has a global scope; you should specify it only once in the db2mscs.cfg file. You cannot use this parameter with the **DB2_INSTANCE** parameter.

**CLUSTER_NAME.** The name of the Failover Cluster. All the resources that you specify following this line are created in this cluster until another **CLUSTER_NAME** parameter is encountered in the file.

**DB2_LOGON_USERNAME.** The user name of the domain account for the DB2 service (that is, *domain\user*). This parameter has a global scope; you should specify it only once in the db2mscs.cfg file.

**DB2_LOGON_PASSWORD.** The password of the domain account for the DB2 service. This parameter has a global scope; you should specify it only once in the db2mscs.cfg file.

**GROUP_NAME.** The name of the Failover Cluster group. If you specify this parameter, the Failover Cluster group is created if it does not exist. If the group exists, it is used as the target group. Any Failover Cluster resource that you specify after this parameter is created in this group or moved into this group until another **GROUP_NAME** parameter is encountered in the file. Specify this parameter once for each group.

**DB2_NODE.** The node number of the database partition server (node) to be included in the current Failover Cluster group. If multiple logical nodes exist on the same machine, each node requires a separate **DB2_NODE** parameter. Specify this parameter after the **GROUP_NAME** parameter so that the DB2 resources are created in the correct Failover Cluster group.

The value for this parameter can optionally contain the network name or IP address that the DB2 software uses for interpartition communication. Typically, in a Failover Cluster Management environment, there are two networks: a private network and a public network. The private network can be used to transfer data between multiple partitions of a DB2 instance. The public network is used for remote client connections. To ensure that the DB2 software always uses the private network for interpartition communication, you can specify the network name or IP address that is associated with the private network, as follows:

```
DB2_NODE = node_number network_name
```

**Note:** Use the next four IP keywords to create an IP Address resource.

**`IP_NAME`.** The name of the IP Address resource that is displayed in Failover Cluster Management. The value for the **`IP_NAME`** parameter is arbitrary and does not have to match the DNS name that is associated with the IP Address. However, the value of the **`IP_NAME`** parameter must be unique in the cluster. When you specify this parameter, a Failover Cluster resource of type IP Address is created. This parameter is required for remote TCP/IP connections. This parameter is optional in a DB2 Workgroup Server Edition single-partition environment.

**`IP_ADDRESS`.** The TCP/IP address for the IP Address resource that you specify for the **`IP_NAME`** parameter. The **`IP_ADDRESS`** parameter is required if you specify the **`IP_NAME`** parameter. The IP address cannot be already be used by any machine in the network.

**`IP_SUBNET`.** The TCP/IP subnet mask for the IP Address resource that you specify for the **`IP_NAME`** parameter. The **`IP_SUBNET`** parameter is required if you specify the **`IP_NAME`** parameter.

**`IP_NETWORK`.** The name of the Failover Cluster network to which the IP Address resource belongs. This parameter is optional. If you do not specify it, the first Failover Cluster network that is detected by the system is used. You must enter the name of the Failover Cluster network exactly as shown under the Networks branch in Failover Cluster Management.

**`NETNAME_NAME`.** The name of the Network Name resource. Specify this parameter to create the Network Name resource. You must specify this parameter for the instance-owning machine.

**`NETNAME_VALUE`.** The DNS name that is associated with the IP address of the IP Address resource that the Network Name resource depends on. You must specify this parameter if you specify the **`NETNAME_NAME`** parameter.

**`NETNAME_DEPENDENCY`.** The name for the IP Address resource that the Network Name resource depends on. Each Network Name resource must have a dependency on an IP Address resource. This parameter is optional. If you do not specify it, the Network Name resource has a dependency on the first IP Address resource in the group.

**DISK_NAME.** The name of the physical disk resource to move to the current group. Specify as many disk resources as you need. The disk resources must exist. Enter the name exactly as shown in Failover Cluster Management.

**INSTPROF_DISK.** The name of the Failover Cluster disk to contain the DB2 instance directory. If you do not specify a value for this parameter, when the **db2mscs** utility configures the DB2 instance for failover support, the instance directory is copied to the first Failover Cluster disk in the current group. If you use this parameter to specify a different Failover Cluster disk for the instance directory, specify the disk name exactly as shown in Failover Cluster Management.

**INSTPROF_PATH.** An optional parameter to specify the path of the DB2 instance directory on the Failover Cluster disk. Use this parameter to provide an alternative way for the **db2mscs** utility to locate the instance directory if the utility cannot obtain the drive letter of the disk resource. **INSTPROF_PATH** will take precedence over **INSTPROF_DISK** if both are specified.

**TARGET_DRVMAP_DISK.** The target Failover Cluster disk for database drive mapping. This parameter specifies the disk that the database will be created on by mapping the disk from the drive that the **CREATE DATABASE** command specifies. If you do not specify this parameter, you must manually map the database drive by using the **db2drvmp** command.

**DB2_FALLBACK.** The specification of whether to force off applications when a DB2 Server resource is taken offline. If you do not want the applications to be forced off, set the **DB2_FALLBACK** parameter to OFF. The default value is ON.

**SERVICE_DISPLAY_NAME.** The display name of the Generic Service resource. Specify this parameter to create the Generic Service resource.

**SERVICE_NAME.** The service name of the Generic Service resource. You must specify this parameter if you specify the **SERVICE_DISPLAY_NAME** parameter.

**SERVICE_STARTUP.** An optional startup parameter for the Generic Service resource.

## Drive mapping

Drive mapping is a mandatory step in implementing a DB2 database across a multiple-partition instance if the partitions are in multiple Failover Cluster groups.

When you use the DB2 **`CREATE DATABASE`** command, you can specify a disk drive where the database is to be created. If you do not specify a drive, a default value is used. The disk on which you create the database must be available to all partitions. However, if partitions are spread across multiple groups, a shared disk drive with the same disk letter cannot exist within multiple groups.

In the example in Figure 6, the group with Partition 0 owns Disk E, and the group with Partition 1 owns Disk F. A shared drive with the same drive letter is not available to both partitions. To alleviate this kind of issue, a drive-mapping mechanism can be used for the **`CREATE DATABASE`** command. On Partition 1, Disk E can be mapped to Disk F. When the **`CREATE DATABASE`** command is issued on Disk E, the data for Partition 0 is created on Disk E, and the data for Partition 1 is created on Disk F. Alternatively, the database could be created on Disk F, and on Partition 0, Disk F could be mapped to Disk E.

Drive mapping is automatically performed when you specify the **`TARGET_DRVMAP_DISK`** parameter in the **`db2mscs`** command input file. You can also manually map drives by using the **`db2drvmp`** command.

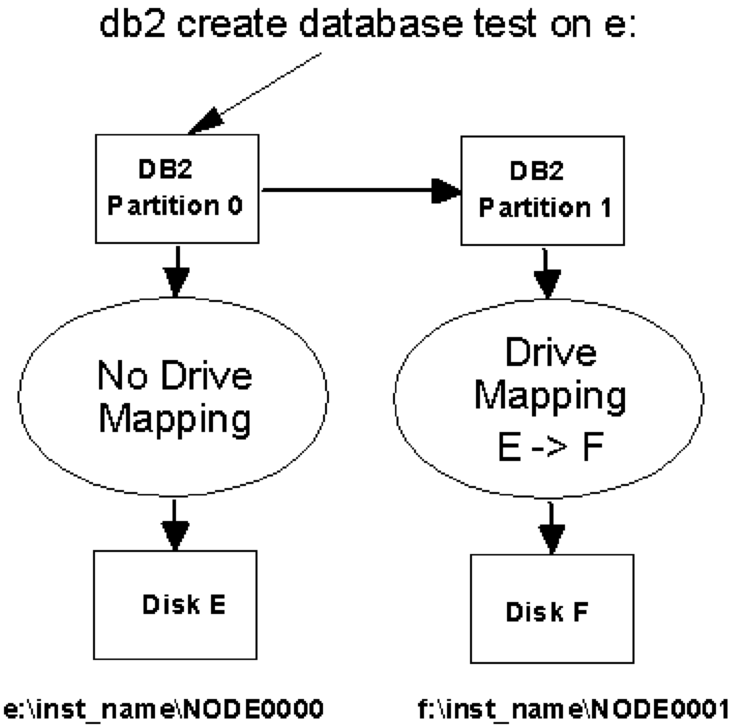


Figure 7. *DB2 drive mapping*

## Planning and preparation

The first step in planning your DB2 Enterprise Server Edition HA environment is to determine whether you want a hot-standby configuration, a mutual-takeover configuration, or a combination of both. You should then define the Failover Cluster configuration based on the requirements of your DB2 Enterprise Server Edition instance. For example, in the hot-standby configuration that is described in this paper, a two-node Failover Cluster is used with one machine being used as a hot spare.

The next step is determining whether multiple partitions will always fail over together. If you want multiple partitions to fail over together, you should place them in the same Failover Cluster group. In this paper, any Failover Cluster group that contains one or more partitions is referred to as a DB2 group. Partitions in the same DB2 group can share disk resources and TCP/IP resources. You should determine the preferred machine owner of each DB2 group along with the failover preferences of the DB2 group. Then, decide whether you want automatic failback within your environment.

Each DB2 group requires one or more Failover Cluster disk resources to store information for the partitions within that DB2 group. Allocate enough disk resources to satisfy the requirements of your database.

For a multiple-partition instance in multiple groups, each DB2 group should have one Failover Cluster TCP/IP resource on the private network. This TCP/IP resource specifies which network the DB2 software should use for internal communication. Optionally, define a Failover Cluster TCP/IP resource on the public network for one or more DB2 groups. This TCP/IP resource is used if remote clients directly connect to partitions within the same DB2 group. Only partitions that are used to receive client requests require this TCP/IP resource. One of the benefits of having multiple partitions accept incoming requests is that it can aid in balancing the workload across multiple partitions.

**Note:** It is important that all TCP/IP resources that you define for use with DB2 software in a Failover Cluster environment be static IP addresses. All the TCP/IP resources must be registered in your DNS or exist in the hosts file.

For each DB2 instance, determine the maximum number of DB2 partitions that can be located on any one machine at any one time. Set the DB2 **`DB2_NUM_FAILOVER_NODES`** environment variable to one fewer than the maximum number of partitions that can be located on one machine. For example, if four partitions can be located on one machine, issue the following command:

```
db2set DB2_NUM_FAILOVER_NODES=3
```

The default value of the environment variable is `2.`

## Hot-standby single-partition configuration (DB2 Enterprise Server Edition example)

In a hot-standby configuration, at least one machine in the Failover Cluster cluster is idle and dedicated as a standby machine in the event of a failure. The standby machine can act as the backup for one or more database partitions, depending on the configuration. In Figure 8, Partition 0 is active on one machine, and there is a single hot spare standby that is available in case the current machine fails.

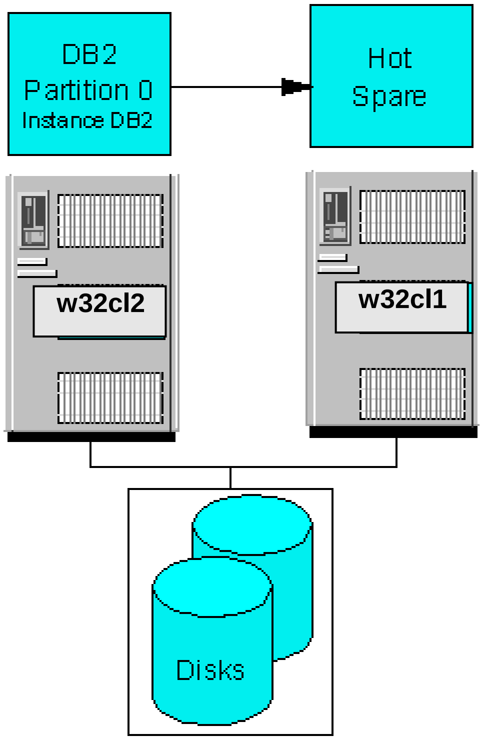


Figure 8. *Hot-standby configuration*

The following steps show how to set up a sample hot-standby configuration using a DB2 single-partition instance, similar to the configuration in Figure 8. To execute the **`db2mscs`** utility, you must be a domain user who has local Administrator authority on each machine in the cluster.

1. Configure the Failover Cluster cluster, and ensure it is healthy.

2. Install DB2 Enterprise Server Edition on every machine in the cluster, allowing the installation process to configure a single-partition instance. You must install the DB2 product on a local drive. (You can do Steps 1 and 2 in reverse order if necessary). The installation process creates an instance called DB2 on each machine.

3. Because the second machine (w32cl2) will be a hot-standby machine, remove the DB2 instance on this machine by issuing the **db2stop** command and then the **db2idrop** command:

   ```
   C:\>db2stop force
   C:\>db2idrop db2
   ```

4. Because the sqllib directory exists locally on each machine, ensure that any programs such as stored procedures or scripts exist on each cluster machine in the appropriate path. For all programs where you can specify a path name, place the program in the instance directory so only one copy of the program is required.

In our example, we created a new instance named myese:

5. Create an instance on w32cl1 named myese:

   ```
   db2icrt -s ese -u mydom\db2admin myese
   ```

6. Ensure that the four ports that are reserved in the services file for fast communication manager (FCM) communication are available on all machines in the cluster. These ports were reserved for the instance. It is also recommended that you run the DB2 instance under a domain account.

7. Stop the instance on the primary machine (w32cl1) by using the **db2stop** command.

8. Before you transform the myese instance into an HA instance, note that the instance directory is currently stored on a local drive:

   ```
   C:\>db2set DB2INSTPROF C:\PROGRAMDATA\IBM\DB2\DB2COPY1
   ```

9. Set up a configuration file to use as input to the **db2mscs** utility. The environment on which the configuration file settings are based is as follows:

| | |
|---|---|
| Cluster name | W32CLUST |
| DB2 instance name | MYESE |
| DB2 instance type | ESE, single partition<br>Check this in Windows services utility for the DB2 service name, for exam “DB2 – DB2COPY1 – MYESE-0.” If your DB2 service does not end with the instance is not a DB2 Enterprise Server Edition instance, and so you must skip the **DB2_NODE** parameter. In this example, because you are using a single-node DB2 Enterprise Server Edition instance, you must set the **DB2_NODE** parameter to 0. |

| | |
|---|---|
| Virtual IP address / subnet | 9.26.118.169 / 255.255.254.0<br>This is a non-pingable address before cluster setup. |
| Netname | mynetname<br>This name depends on the virtual IP address. You must register this name in the DNS with the virtual IP address, for example, mynetname 9.26.118.169.<br>Ensure that the values of the **NETNAME_DEPENDENCY** and **IP_NAME** parameters in the configuration file are equal. |
| Disk | DISK U:<br>This disk will be added to the cluster group.<br>It should already be available; you should be able to view it in the Failover Cluster Management interface under **Storage > Available Storage**. |

The **db2mscs** utility configuration file is as follows:

```
DB2_INSTANCE = myese
CLUSTER_NAME = w32clust
DB2_LOGON_USERNAME = mydom\db2admin
DB2_LOGON_PASSWORD = xxx

GROUP_NAME = MYESEGroup
DB2_NODE = 0
IP_NAME = MYESEIP
   IP_ADDRESS = 9.26.118.169
   IP_SUBNET = 255.255.254.0
   IP_NETWORK = Cluster Network 1

NETNAME_NAME = MYNETNN
   NETNAME_VALUE = MYNETNV
   NETNAME_DEPENDENCY = MYESEIP
DISK_NAME = DISK U:
INSTPROF_DISK = DISK U:
```

Notes:

- The **NETNAME_DEPENDENCY** parameter in the **db2mscs** input file is configured to use a TCP/IP address that is defined on Cluster Network 1, which in this configuration is the public network.
- The name of the disk that you specify in the configuration file must match the names that you define in the cluster manager; the name is case sensitive. The default disk names are in the form “Cluster Disk #”.

Back up all databases within the instance that you will transform into an HA clustered instance.

10. Run the **db2mscs** utility, specifying the **-f** parameter for the configuration file name,

and ensure that the utility is successful:

```
C:\>db2mscs -f:db2mscs.cfg.db2
DB21500I The DB2MSCS command completed successfully.
```

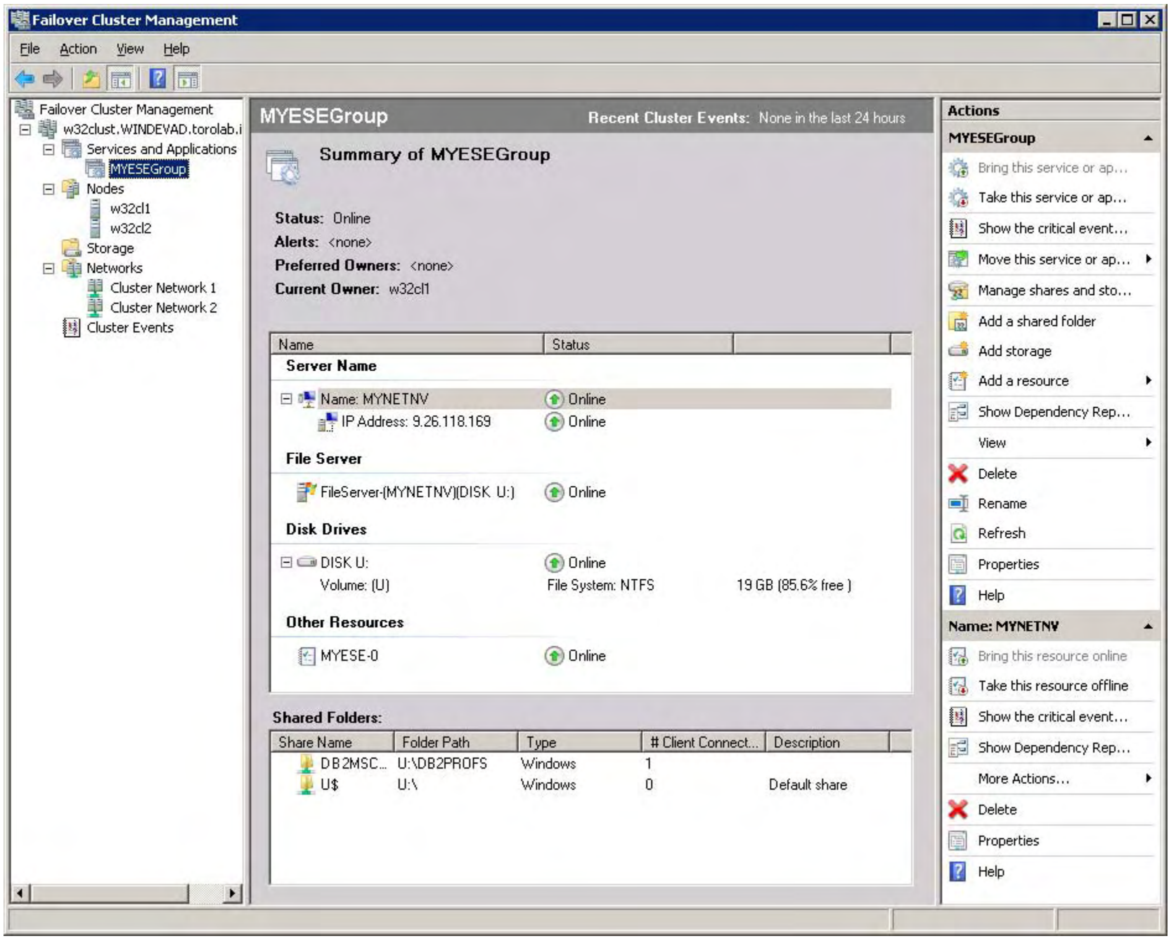
The execution of the **db2mscs** utility transforms the DB2 instance into a clustered instance in the w32clust cluster. The MYESEGroup group is created, which contains a single partition, one new IP address, and one disk resource. The MYESEGroup also contains a network name and file server resource because the group contains the instance-owning partition. The instance directory is moved to the new file share in the MYESEGroup group.

The **db2ilist** command indicates that the instance is clustered by showing C:*cluster name* after the instance name in the output:

```
C:>db2set db2instprof
\\mynetnv\DB2MSCS-MYESE
C:\>db2ilist
MYESE             C : w32clust
```

The following Failover Cluster Management screen capture provides a summary of the activities:

- A group called MYESEGroup has been created.
- The group contains one IP address, one File Server, and one disk resource.
- A resource of type DB2 Server has been created for the DB2 partition.
- A Network Name resource and File Server resource have been created in the group to hold the instance directory. This network name has been created on Cluster Network 1.



MYESEGroup    Recent Cluster Events: None in the last 24 hours

Summary of MYESEGroup

Status: Online
Alerts: <none>
Preferred Owners: <none>
Current Owner: w32cl1

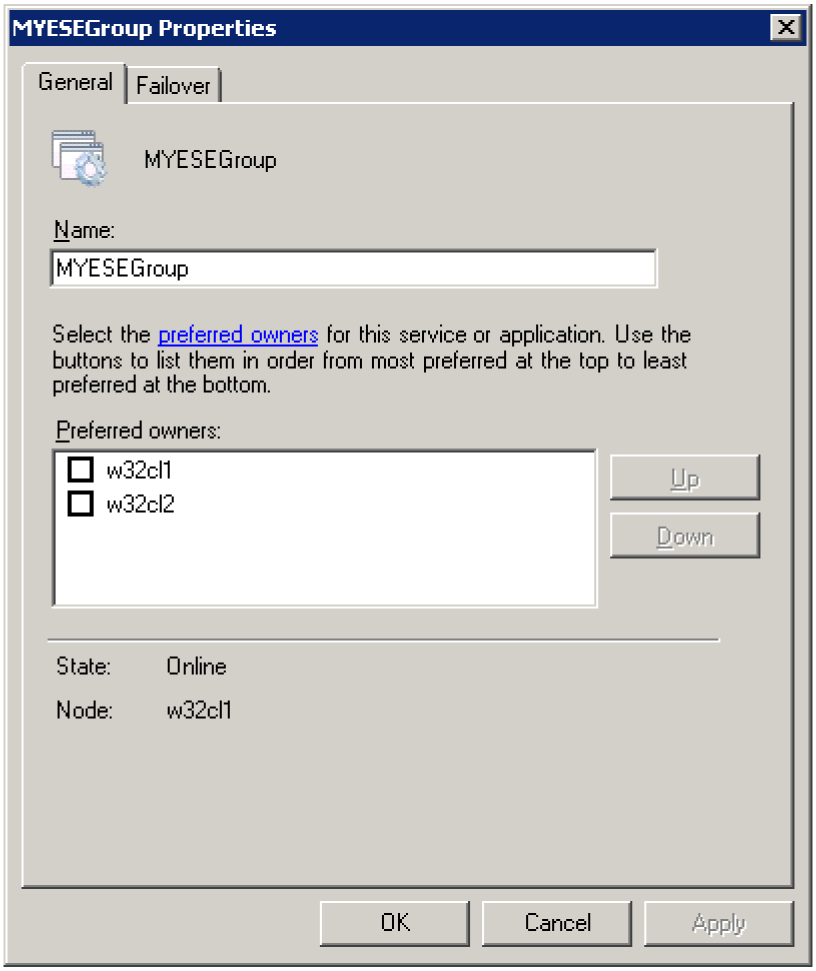
| Name | Status | |
|---|---|---|
| **Server Name** | | |
| Name: MYNETNV | Online | |
| IP Address: 9.26.118.169 | Online | |
| **File Server** | | |
| FileServer-(MYNETNV)(DISK U:) | Online | |
| **Disk Drives** | | |
| DISK U: | Online | |
| Volume: (U) | File System: NTFS | 19 GB (85.6% free ) |
| **Other Resources** | | |
| MYESE-0 | Online | |

Shared Folders:

| Share Name | Folder Path | Type | # Client Connect... | Description |
|---|---|---|---|---|
| DB2MSC... | U:\DB2PROFS | Windows | 1 | |
| U$ | U:\ | Windows | 0 | Default share |

*Figure 9. Failover Cluster Management information about instance myese*

11. Determine which machine to use as the primary machine and which machine to use as the standby machine. MYESEGroup currently has no preferred owners. In this example, choose w32cl1 as the preferred owner of MYESEGroup and w32cl2 as the hot spare. In Failover Cluster Management, right-click **MYESEGroup > Properties**, and select a preferred own as shown:



MYESEGroup

Name:

MYESEGroup

Select the preferred owners for this service or application. Use the buttons to list them in order from most preferred at the top to least preferred at the bottom.

Preferred owners:

□ w32cl1

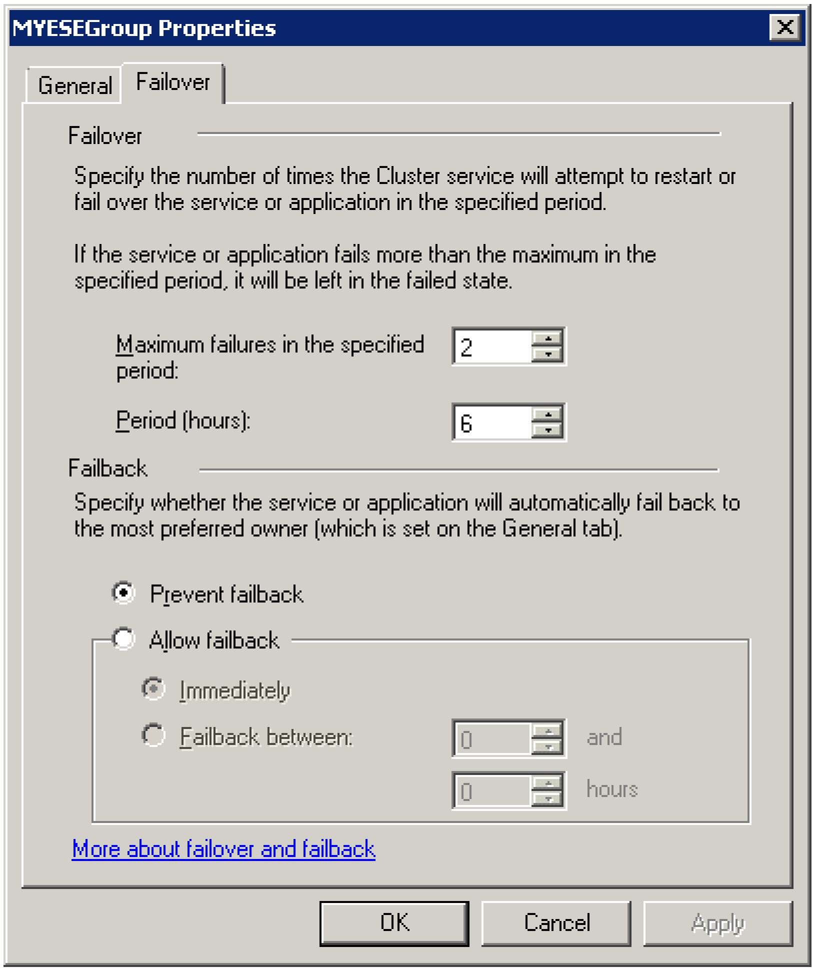
□ w32cl2

State: Online

Node: w32cl1

Figure 10. *Failover Cluster Management Properties for MYESEGroup*

12. Determine whether you want the DB2 group to automatically fail back to the preferred machine after it recovers from a failure. If you do not want to use automatic failback, no action is required because the default behavior is to prevent failback. If you want to use automatic failback, modify the information in the Properties as shown in the following screen capture:
    - On the **Failback** tab, click **Allow failback**.
    - If you want to enable failback when the preferred machine is back online, also click **Immediately.**
    - If you want to enable failback only for a particular period of the day, also click **Failback between**, and specify the times. If the preferred machine comes back online during this time window, failback occurs. If the preferred machine comes back online at a different time, failback does not occur. For example, if the window is 1 a.m. to 8 a.m. and the machine comes back online at 12:59 a.m., the group does not automatically fail back at 1 a.m.



MYESEGroup Properties

Failover

Specify the number of times the Cluster service will attempt to restart or fail over the service or application in the specified period.

If the service or application fails more than the maximum in the specified period, it will be left in the failed state.

Maximum failures in the specified period: 2

Period (hours): 6

Failback

Specify whether the service or application will automatically fail back to the most preferred owner (which is set on the General tab).

● Prevent failback

○ Allow failback

● Immediately

○ Failback between: 0 and 0 hours

Figure 11. *Failback settings for MYESEGroup*

13. Ensure all databases within the instance DB2 are on the disk drives in the same group as instance DB2. In this example, all data should be on disk U. To move existing databases to this shared drive, use a redirected restore. For information about how to perform the redirected restore, see the DB2 Information Center. If you create a new database, make sure that you create the database and its table spaces on disk U, as shown:

```
C:\>db2 create db sample on u:
DB20000I  The CREATE DATABASE command completed successfully.

C:\>db2 create tablespace ts in nodegroup ibmdefaultgroup managed by
database using (file ‘u:\container' 10000) on node (0)
DB20000I  The SQL command completed successfully.
```

If the DB2 instance fails over to another machine and does not have access to all its data files, database errors occur.

## Hot-standby single-partition configuration (DB2 Workgroup Server Edition example)

In a hot-standby configuration, at least one machine in the Failover Cluster cluster is idle and dedicated as a standby machine in the event of a failure. The standby machine can act as the backup for one or more database partitions, depending on the configuration. The following steps show how to set up a sample DB2 Workgroup Server Edition configuration. The DB2 software is active on one machine, and there is a single hot spare that is available in case the current machine fails. This example is similar to the DB2 Enterprise Server Edition example in the previous section and uses the same machine names.

1. Configure the Failover Cluster cluster, and ensure that it is healthy.

2. Install DB2 Workgroup Server Edition on every machine in the cluster, allowing the installation process to configure a single-partition instance. You must install the DB2 software on a local drive. (You can do steps 1 and 2 in reverse order if necessary). The installation process creates an instance called DB2 on each machine.

3. Because the second machine (w32cl2) will be a hot-standby machine, remove the DB2 instance on this machine by issuing the **`db2stop`** command and then the **`db2idrop`** command:

   ```
   C:\>db2stop force
   C:\>db2idrop db2
   ```

4. Because the `sqllib` directory exists locally on each machine, ensure that any programs such as stored procedures or scripts exist on each cluster machine in the appropriate path. For all programs where a path name can be specified, place the program in the instance directory so only one copy of the program is required.

In our example, we created a new instance named mywse:

5. Create an instance on w32cl1 named mywse:

   ```
   db2icrt -s wse mywse
   ```

6. The instance created by the install will have four ports reserved in the services file for FCM communication. Ensure these ports are available on all machines in the cluster. It is also recommended that the DB2 instance be run under a domain account.

7. Stop the instance on the primary machine (w32cl1) by using the **`db2stop`** command.

8. Before you transform the mywse instance into an HA instance, note that the instance directory is currently stored on a local drive:

   ```
   C:\>db2set DB2INSTPROF C:\PROGRAMDATA\IBM\DB2\DB2COPY1
   ```

9. Set up a configuration file to use as input to the **`db2mscs`** utility. The environment on which the configuration file settings are based is as follows:

| | |
|---|---|
| Cluster name | `W32CLUST` |
| DB2 instance name | `MYWSE` |
| DB2 instance type | WSE<br>*To verify this, bring up Windows services and confirm the DB2 service name, e.g., "DB2 – DB2COPY1 – MYWSE". If your DB2 service does not end with "-0" then it is a WSE instance and you must skip the DB2_NODE in the configuration file.* |
| Virtual IP address / subnet | `9.26.118.169 / 255.255.254.0`<br>*This is a non-pingeable address prior to cluster setup* |
| Disk | `Cluster Disk 3`<br>*This disk will be added to the cluster group.*<br>*It should already be available under Storage > Available Storage in the Failover Cluster Management interface* |

The **`db2mscs`** utility configuration file is as follows:

```
DB2_INSTANCE = mywse
CLUSTER_NAME = w32clust
GROUP_NAME = MYWSEGroup

IP_NAME = MYWSEIP
   IP_ADDRESS = 9.26.118.169
   IP_SUBNET = 255.255.254.0
   IP_NETWORK = Cluster Network 1

DISK_NAME = Cluster Disk 3
INSTPROF_DISK = Cluster Disk 3
```

**Note:** The name of the disk that you specify in the configuration file must match the names that you define in the cluster manager; the name is case sensitive.

Back up all databases within the instance that you will transform into an HA clustered instance.

10. Run the **`db2mscs`** utility, specifying the **`-f`** parameter for the file name, and ensure that the utility is successful:

```
C:\>db2mscs -f:db2mscs.cfg.db2
DB21500I The DB2MSCS command completed successfully.
```

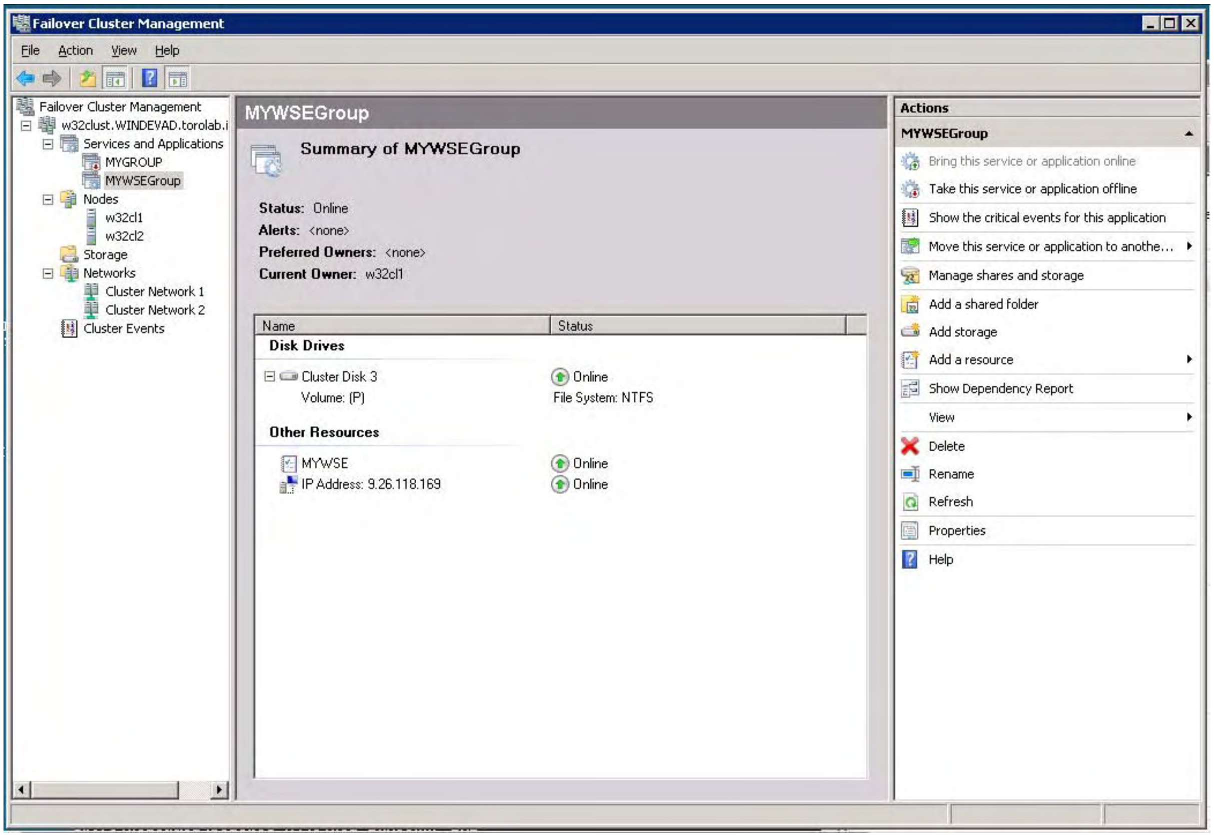
The execution of the **db2mscs** utility transforms the instance DB2 into a clustered instance that resides within the w32clust cluster. The MYWSEGroup group is created, which contains a single partition, one new IP address, and one disk resource. The instance directory is moved to the clustered disk drive.

The **db2ilist** command now indicates that the instance is clustered by showing *C:cluster name* after the instance name in the output:

```
C:>db2set db2instprof
P:\DB2PROFS
C:\>db2ilist
MYWSE             C : w32clust
```

The following Failover Cluster Management screen capture provides a summary of the activities:

- A group called MYWSEGroup has been created.
- The group contains one IP address and one disk resource.
- A resource of type DB2 Server has been created for the DB2 partition.


Failover Cluster Management
File
Action
View
Help
Failover Cluster Management
w32clust.WINDEVAD.torolab.i
Services and Applications
MYGROUP
MYWSEGroup
Nodes
w32cl1
w32cl2
Storage
Networks
Cluster Network 1
Cluster Network 2
Cluster Events
Actions
MYWSEGroup
Bring this service or application online
Take this service or application offline
Show the critical events for this application
Move this service or application to anothe...
Manage shares and storage
Add a shared folder
Add storage
Add a resource
Show Dependency Report
View
Delete
Rename
Refresh
Properties
Help

MYWSEGroup

Summary of MYWSEGroup

**Status:** Online
**Alerts:** <none>
**Preferred Owners:** <none>
**Current Owner:** w32cl1

| Name | Status |
|---|---|
| **Disk Drives** | |
| Cluster Disk 3 | Online |
| Volume: (P) | File System: NTFS |
| **Other Resources** | |
| MYWSE | Online |
| IP Address: 9.26.118.169 | Online |

Figure 12. *Failover Cluster Management information about instance mywse*

11. Determine which machine to use as the primary machine and which machine to use as the standby machine. MYWSEGroup currently has no preferred owners. In this example, choose w32cl1 as the preferred owner of MYWSEGroup and w32cl2 as the hot spare. In Failover Cluster Management, right-click **MYWSEGroup** > **Properties**, and select a preferred owner.

12. Determine whether you want the DB2 group to automatically fail back to the preferred machine after it recovers from a failure. If you do not want to use automatic failback, no action is required because the default behavior is to prevent failback. If you want to use automatic failback, modify the information in the Properties box for each DB2 group:
    - On the **Failback** tab, click **Allow failback**.
    - If you want to enable failback when the preferred machine is back online, also click **Immediately.**
    - If you want to enable failback only for a particular period of the day, also click **Failback between**, and specify the times**.** If the preferred machine comes back online during this time window, failback occurs; otherwise, failback does not occur. For example, if the window is 1 a.m. to 8 a.m. and the machine comes back online at 12:59 a.m., the group does not automatically fail back at 1 a.m.

13. Ensure that all databases within the instance are on disk drives that exist within the same group as instance mywse. In this example, all data should be on Disk P. To move existing databases to this shared drive, use a redirected restore. For information about how to perform the redirected restore, see the DB2 Information Center. If you create a new database, make sure that you create the database and its table spaces on Disk P, as shown:

```
C:\>db2 create db sample on p:
DB20000I  The CREATE DATABASE command completed successfully.

C:\>db2 create tablespace ts in nodegroup ibmdefaultgroup managed by
database using (file ‘p:\container' 10000) on node (0)
DB20000I  The SQL command completed successfully.
```

If the DB2 instance fails over to another machine and does not have access to all its data files, database errors occur.

## Mutual-takeover multiple-partition configuration

A mutual-takeover multiple-partition configuration has a DB2 partition running on each machine in the cluster. If a machine in the cluster fails, the result is multiple DB2 partitions running on the same machine. You must carefully plan to ensure that each machine in the cluster is capable of handling the workload that can be placed on it.

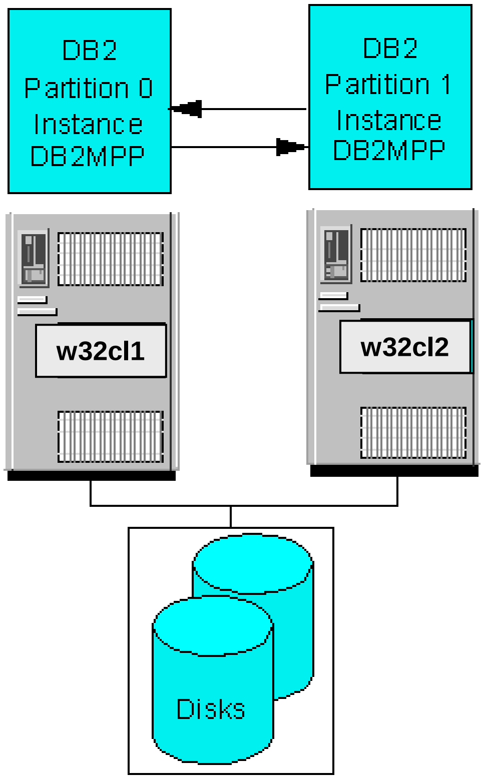


Figure 13. *Mutual-takeover multiple-partition configuration*

This example demonstrates how to configure a two-partition mutual-takeover configuration in a two-node cluster. Initially, there is one DB2 partition on each machine. Configuring the mutual-takeover environment is very similar to configuring the hot-standby environment, except that no machines are idle.

1. Install Failover Cluster.

2. Configure a two-partition DB2 instance.

3. Stop the instance:

```
C:\>db2nlist /s
List of nodes for instance "DB2MPP" is as follows:
Node: "0" Host: "w32cl1" Machine: "W32CL1" Port: "0" - "stopped"
Node: "1" Host: "w32cl2" Machine: "w32cl2" Port: "1" - "stopped"
```

**4.** Create the following **db2mscs** utility input file. In this configuration, it is assumed that only Partition 0 will receive incoming remote client requests and thus has an MSCS TCP/IP resource that is defined on the public network.

```
DB2_INSTANCE=DB2MPP
CLUSTER_NAME=W32CLUST
DB2_LOGON_USERNAME=mydom\db2admin
DB2_LOGON_PASSWORD=xxx

GROUP_NAME=DB2 Group 0
DB2_NODE=0
IP_NAME=MYMPPIP
   IP_ADDRESS=9.26.118.169
   IP_SUBNET=255.255.254.0
   IP_NETWORK=Cluster Network 1
NETNAME_NAME=MYNETN
   NETNAME_VALUE=MYNETV
   NETNAME_DEPENDENCY=MYMPPIP
DISK_NAME=DISK U:
TARGET_DRVMAP_DISK=DISK U:

GROUP_NAME=DB2 Group 1
DB2_NODE=1
DISK_NAME=DISK Y:
TARGET_DRVMAP_DISK=DISK Y:
```

5. For drive mapping to succeed, in Failover Cluster Management, manually rename the disk names from the default “Cluster Disk #” format to “DISK *X*:”.

6. Run the **db2mscs** utility, specifying the **–f** parameter for the file name, and ensure that the command is successful:

```
C:\>db2mscs -f:db2mscs.cfg.db2
DB21500I The DB2MSCS command completed successfully.
```

7. Determine which machine to use as the primary machine for each of the DB2 groups. In this example, the primary machine for DB2 Group 0 is w32cl1, and the primary machine for DB2 Group 1 is w32cl2.

```
C:\>db2nlist /s
List of nodes for instance "DB2MPP" is as follows:
Node: "0" Host: "w32cl1" Machine: "W32CL1" Port: "0" - "running"
Node: "1" Host: "w32cl2" Machine: "W32CL2" Port: "0" - "running″
```

8. Optional: Configure automatic failback for the DB2 groups.

9. Create or restore all databases on the highly available disks.

## DB2 administration server configuration

Note that starting in Version 9.7, the Control Center tools have been deprecated and might be removed in a future release. You can use instead IBM Data Studio and the IBM InfoSphere(R) Optim(TM) tools for managing DB2 for Linux, UNIX, and Windows databases and developing data-centric applications.

If you choose to continue using DB2 Control Center, here are the instructions for configuring DB2 administration server for high availability:

The DB2 Control Center uses the DB2 administration server to administer DB2 instances and databases. If you want the DB2 administration server to be highly available, you must cluster it. The steps to cluster the DB2 administration server are similar to the steps to transform any other regular instance. In the following example, you will use the DB2 administration server to administer the DB2MPP instance that you created in the mutual-takeover multiple-partition configuration. The following example also reuses the shared disk and IP address that you configured for Partition 0, resulting in Partition 0 and the DB2 administration server sharing these resources. Because they are sharing the same resources, you must place the DB2 administration server in the same group as Partition 0.

The instructions for configuring the DB2 administration server for instance DB2MPP are as follows:

1. Stop the DB2 administration server (DAS) on all machines:

   ```
   C:\>db2admin stop
   SQL4407W  The DB2 Administration Server was stopped successfully.
   SQLSTATE=00000
   ```

   If a DB2 administration server does not exist on W32CL1, use the **`db2admin create`** command to create this DAS instance on W32CL1.

*2.* Drop the DB2 administration server on all cluster nodes except the first node by issuing the following command on each machine:

   ```
   C:\>db2admin drop
   SQL4402W  The DB2ADMIN command was successful.
   ```

3. On the first node (W32CL1) where the DB2 administration server is located, in Windows Services, modify the DB2 administration server instance so that it is set to start manually. The name of the DB2 administration server is DB2DAS00, so it is displayed as DB2DAS-DB2DAS00 in Windows Services.

4. Create a configuration input file to use with the **`db2mscs`** utility to cluster the DB2 administration server:

   ```
   DAS_INSTANCE=DB2DAS00
   CLUSTER_NAME=MYCLUSTER
   ```

```
DB2_LOGON_USERNAME=mydom\db2admin
DB2_LOGON_PASSWORD=xxx
GROUP_NAME=DB2 Group 0
DISK_NAME=Disk U:
INSTPROF_DISK=Disk U:
```

Note that the group name DB2 Group 0 is the same group that you used for configuring Partition 0; thus, all resources will be created in the same group as for Partition 0. You are not configuring an IP address, because the IP address is already configured for Partition 0. However, you must specify the disk again because that is where all information that is associated with the DB2 administration server will be placed. A Generic Service will be created, which will allow the DB2 administration server to be monitored by MSCS.

5. From W32CL1, execute the **`db2mscs`** utility:

   ```
   c:\>db2mscs -f:db2mscs.cfg.admin
   DB21500I The DB2MSCS command completed successfully.
   ```

6. On all clients that you use for DB2 administration, use the DB2 Control Center to remove any references to the DB2 administration server. Then, use the DB2 Control Center to recatalog a reference to the DB2 administration server by using the highly available cluster IP address that you defined on the public network.

The DB2 administration server is now integrated within the cluster. Ensure that any remote client connections go through the highly available IP address. Also, ensure that you place any scripts or data that are needed by the DB2 administration server on the highly available disks that are associated with the DB2DAS00 instance. Failover Cluster Management now shows instance DB2MPP configured in the cluster, and the group containing Partition 0 also contains the DB2 administration server.

You can also use the previous steps to cluster the DB2 administration server for instance DB2 in the single-partition configuration example.

**Note:** Because only one DB2 administration server can run on a single machine at one time, you cannot cluster the DB2 administration server for each instance in a mutual-takeover single-partition configuration.

## Remote client connections

Failover Cluster provides the ability to create a highly available TCP/IP resource. Remote clients should use the highly available IP address that you define on the public network when connecting to the server.

When cataloging a TCP/IP connection from the client to the DB2 server, you must use the IP address that you configure for the cluster that is located in the same group as the partition that you want to connect to. Thus, when configuring a client connection to a particular database partition, you must use the highly available TCP/IP address that is in the same group as the DB2 partition. By cataloging database connections to multiple partitions, you also spread the DB2 coordinator functionality across multiple partitions. If you use an IP address that is associated with either a physical machine or an IP address in a group that is different from that of the partition, there is no guarantee that the IP address points to the machine where the database partition is actually located.

The other factor that you must consider is that the client connects to the server using a specified port. That identical port must be available to the instance on all machines within the cluster and you should define the port in the services file on each machine in the cluster.

The following steps provide an example of how to catalog connections from a remote client to database SAMPLE for instance MYESE, which you created in the hot-standby single-partition example:

1. To reserve unused ports for MYESE, add the following entries to the services file for all machines in the cluster:

   ```
   db2cMYESE         50000/tcp    #connection port for the DB2 instance MYESE
   db2iMYESE         50001/tcp    #interrupt port for the DB2 instance MYESE
   ```

2. Set the **`DB2COMM`** parameter to `TCPIP` for each partition:

   ```
   C:\>set DB2INSTANCE=MYESE
   C:\> db2set DB2COMM=TCPIP
   ```

3. Update the database manager configuration for instance MYESE so that it knows which port to listen on for incoming TCP/IP clients:

   ```
   C:\>db2 update dbm cfg using svcename db2cmyese
   ```

4. For the changes in steps 2 and 3 to take effect, take the partition offline and then bring it back online. In Failover Cluster Management, right-click the resource representing the DB2 partition and select **Take this resource offline**. After all DB2 resources are in an offline state, right-click the same resources and select **Bring this resource online**.

Now that all partitions are ready to receive incoming remote requests, the following steps will catalog the database partition from a remote machine.

5. You must update the services file on the remote client by using entries that are similar to

those on the server machines.

6. Issue the following commands to catalog database SAMPLE for instance MYESE:

```
C:\>db2 catalog tcpip node nodea remote myeseip server db2cmyese
C:\>db2 catalog db sample  at  node nodea
C:\>db2 terminate
C:\>db2 connect to sample user db2admin using xxx
```

   For cataloging the tcpip node, you can replace myeseip with its corresponding IP address in the cluster.

   Alternatively, use the DB2 Configuration Assistant to manually catalog connections to the database using a graphical interface. If you use this mechanism, ensure that you use the highly available IP address to catalog the database connection.

### Automatic Client Reroute

Another method of achieving a highly available TCP/IP connection for remote clients is to use the DB2 client reroute feature. You can use client reroute to specify an alternative server. When clients connect to the DB2 server, they automatically pick up the alternative server name and port number. If the connection to the DB2 server fails or drops, client reroute automatically reroutes the client to the alternative server. For further details on client reroute, see the DB2 Information Center.

Note: automatic client reroute can be used in conjunction with virtual IP address by specifying the virtual IP address as the alternate server.

## User scripts

You can execute a batch script before and after you bring each DB2 resource online and before and after you take each DB2 resource offline. These batch scripts are in the instance directory of each instance, so each instance has a separate copy of these scripts. To determine the instance directory, issue the **db2set** command, and then append the instance name to the results, as shown in this example:

```
c:>set DB2INSTANCE=DB2MPP
c:\>db2set  db2instprof
\\mynetname\DB2MSCS-DB2MPP
```

In this particular case, the instance directory for instance DB2MPP is located in the \\mynetname\DB2MSCS-DB2MPP\DB2MPP path. The script that executes before each DB2 partition is brought online is called db2cpre.bat; the script that executes after each DB2 partition is brought online is called db2cpost.bat. These scripts are also referred to as the pre-online and post-online scripts. The script that executes before each DB2 partition is taken offline is called db2apre.bat; the script that executes after each DB2 partition is taken offline is called db2apost.bat. These scripts are also referred to as the pre-offline and post-offline scripts. These batch files are optional; they do not exist by default and are executed only if they exist. These batch files are launched by Failover Cluster and are run in the background. The script files must redirect standard output to record any output from commands that are run from within the script files.

The pre-online script must finish running before any attempt to bring the DB2 resource online is made. Thus, it is important that the commands in the pre-online script execute in a reasonable amount of time so Failover Cluster does not time out on its attempt to bring the DB2 resource online. Because the pre-offline script also runs synchronously before the DB2 resource is taken offline, you should ensure the script executes efficiently so it does not significantly affect failback time.

Note: The user scripts are executed with the **DB2INSTANCE** and **DB2NODE** environment variables set to the values corresponding to the resource executing the scripts. The **DB2NODE** environment value corresponds to the partition number.

The pre-offline and post-offline scripts are not executed if the DB2 process terminates abnormally.

## Testing a configuration

When testing a high availability configuration, you ideally want to test all points of failure to ensure that a redundant path is used. For example, you should test such things as disk failures, network failures, machine failures, and software failures.

The objective of this section is not to provide detail on how to test the whole system but to show you how to test the DB2 portion of the system.

1. From the remote client, connect to the highly available database, and issue a query against a table that is distributed across all partitions.
2. Move the group containing a partition to another machine.
3. From the remote client, attempt to reissue the query from step 1. If the query fails, reconnect to the same highly available database, and then reissue the query. This attempt should succeed.

   **Note:** If a partition failed because of a hard crash such as a machine failure, the DB2 software does not attempt to force off any database applications. Any uncommitted transactions that used the failed partition are rolled back, leaving the database in a transactionally consistent state.

4. Move the group containing the partition back to the primary machine.
5. From the remote client, attempt to reissue the query from step 1. If the query fails, reconnect to the same highly available database, and then reissue the query. This attempt should succeed.
6. Repeat steps 1 through 5 using various simulated hardware and software failures. Use a client workload that closely simulates the workload that you expect in the production environment.

**Note:** If a cluster spans more than two machines, you should test multiple machine failures.

## Migration

For information about 32-bit to 32-bit migration from DB2 9.1 to DB2 9.5 and 64-bit to 64-bit migration from DB2 9.1 to DB2 9.5, see “Migrating DB2 servers in Microsoft Cluster Server environments” in the DB2 9.5 Information Center. For upgrading to DB2 9.7, see “Upgrading DB2 servers in Microsoft Cluster Server environments” in the DB2 9.7 Information Center.

Direct 32-bit to 64-bit migration of a Failover Cluster DB2 instance on the same machine is not supported. To migrate a 32-bit clustered instance on a 64-bit machine to a 64-bit clustered instance:

1. Uncluster the instance by entering `db2mscs –u:` *`instance name`* on the same machine that you originally clustered the instance or using manual steps (see Appendix D).
2. Migrate the 32-bit instance to a 64-bit instance, following the steps in “Migrating DB2 32-bit servers to 64-bit systems (Windows)” in the DB2 Information Center.
3. Re-cluster the instance by entering `db2mscs –f:`*`input file`*.

Microsoft does not support the direct upgrade of a Windows 2003 cluster to Windows 2008. To migrate a cluster from Windows 2003 to Windows 2008, see the Microsoft document entitled “Step-by-Step Guide for Migrating Settings from a Cluster Running Windows Server 2003 to a Cluster Running Windows Server 2008.”

## Rolling upgrades

A rolling upgrade upgrades software on the cluster while keeping the application online. A Failover Cluster environment is ideal for performing a rolling upgrade because the partitions can be online on one machine while you are upgrading the other machine. Rolling upgrades are supported for DB2 installations that do not require either database or instance migration.

**Important:** Ensure that multiple database partitions are not actively running on a different code level at the same time if you are upgrading a multiple-partition configuration.

The following example demonstrates how to do a rolling upgrade of the mutual-takeover multiple-partition configuration that was described earlier. The strategy is to move all partitions to the second half of the machines and then upgrade the first half. Next, take all partitions offline, move them back to the first half of the machines, and then bring them online. Finally, you can upgrade the second half of the machines and move partitions back to the location that you want.

Initially, Partition 0 is on machine SVTHA05, and partition 1 is on machine SVTHA06.

1. Move Partition 0 to SVTHA06 so that SVTHA05 is idle.
2. Stop the cluster service on SVTHA05 by issuing `net stop clussvc`.
3. Apply the DB2 fix pack on SVTHA05
4. Take all DB2 resources offline.
5. Start the cluster service on SVTHA05 by issuing `net start clussvc`.
6. Move all DB2 groups to SVTHA05.
7. Bring the groups online, which results in SVTHA06 being idle.
8. Stop the cluster service on SVTHA06.
9. Apply the DB2 fix pack on SVTHA06.
10. Start the cluster service on SVTHA06.
11. Move Partition 1 back to SVTHA06 at a convenient time.

## DB2 authentication

You should use domain security (domain users and domain groups) so that if the DB2 software fails over to another machine, the same domain user can connect to the database with the same authority.

By default, domain administrators have full access to the database. To restrict SYSADM authority to domain users and groups:

1. Create a domain group. The group name must conform to the DB2 naming conventions.
2. Add any domain users who require SYSADM authority to this domain group.
3. From the machine that runs the DB2 software, set the **`SYSADM_GROUP`** database manager configuration parameter to the name of the domain group.
4. Restart the DB2 instance.

# Appendix A. Limitations and restrictions

When you run DB2 software in a Failover Cluster environment, the following limitations and restrictions apply:

- For drive mapping to succeed, the drive letter is used when referring to Physical Disk resources in the Failover Cluster interface. Use drive letters when specifying the path for DB2 databases and table space containers.
- Because Failover Cluster cannot manage raw disk devices, do not configure the DB2 software to use raw disk devices.
- You must manage the DB2 partition from a Failover Cluster interface such as Failover Cluster Management. Failover Cluster does not monitor resources that you started outside its control, because it is not aware that you started the resources. Failover Cluster also treats any attempts to stop a DB2 partition outside its control as a resource failure if you initially brought that resource online by using Failover Cluster Management. Thus, you should not use mechanisms such as the **`db2stop`** and **`db2start`** commands and the DB2 Control Center to start and stop a DB2 instance in a Failover Cluster environment.
- If the automatic creation of a computer object in Active Directory is disabled in your environment, you must manually create the computer object by using the same **`NETNAME_VALUE`** parameter value that is in your **`db2mscs`** utility configuration file.

## Appendix B. Frequently asked questions

1. Q. My DB2 Enterprise Server Edition instance does not start on other machines in the cluster.
   A. The DB2 Enterprise Server Edition instance reserves ports in the services file that is used during startup. If the ports are not allocated in the services file, DB2 Enterprise Server Edition automatically adds these entries to the services file. Ensure that these ports are available on all machines in the cluster. The following example shows entries from a services file for instance DB2MPP:

   ```
   DB2_DB2MPP 60000/tcp
   DB2_DB2MPP_1 60001/tcp
   DB2_DB2MPP_2 60002/tcp
   DB2_DB2MPP_END 60003/tcp
   ```

2. Q. I ran the **db2mscs** utility and I got the following error:

   ```
   c:\>db2mscs -f:db2mscs.cfg.badip
   DB21524E Failed to create the resource "mscs5". System error: "The
   cluster IP address is already in use.”
   ```

   A. The resource mscs5 has a corresponding IP address that is already in use on the network. Ensure that the IP address is not already in use.

3. Q. I ran the **db2mscs** utility and I got the following error:

   ```
   c:\>db2mscs -f:db2mscs.cfg.baddisk
   DB21526E Failed to move resource “O:”. Win32 error: “The cluster
   resource could not be found.”
   ```

   A. The Physical Disk resource that is specified does not exist within the cluster. Ensure that the name of the disk in the input configuration file is identical to the name in Failover Cluster Management. In this example, we entered “O:” for the disk name when it should have been “Disk O:”.

4. Q. I ran the **db2mscs** utility and it did not execute successfully.
   A. Refer to the message reference to determine the course of action based on the return code from the **db2mscs** utility.

5. Q. When I execute the **db2nlist** or **db2nchg** command, it fails after a couple of minutes with a communication error.
   A. Ensure that all TCP/IP addresses that are used have a corresponding entry in the DNS or have an entry in the hosts file on each machine.

6. Q. My partition does not fail over.
   A. You must start the partition through a cluster interface, such as Failover Cluster Management. Otherwise, the cluster is not aware that it must try to keep this DB2 partition

online.

7. Q. I issue the **db2stop** command, and the partitions automatically restart.
   A. If you started the DB2 partitions through Failover Cluster Management, you must stop them through a cluster interface. Failover Cluster treats the outcome of **db2stop** command as a resource failure and attempts to bring the DB2 resource back online.

8. Q. I try to take the group containing the DB2 partition offline, but the DB2 resource does not go offline.
   A. Ensure that the **DB2_FALLBACK** parameter is set to `ON` for that partition. For example, issue `db2set DB2_FALLBACK=ON`. To successfully take the group containing the DB2 partition offline at this point, stop the cluster service on the machine where the DB2 partition is trying to go offline.

9. Q. My **CREATE DATABASE** command fails in my multiple-partition configuration.
   A. Ensure that you successfully mapped the drives. You must take all DB2 resources offline and then bring them back online for the drive mapping to take effect.

10. Q. My remote client successfully connects to the database partition, but when I fail over the partition, I cannot successfully reconnect.
    A. Ensure that you cataloged the client to connect to the highly available IP Address resource that you configured in the same group as the database partition. Also, ensure that the port that is defined by the **SVCENAME** parameter in the database manager configuration file is not already in use.

11. Q. When I issue DB2 commands locally from the DB2 command line processor, I get one of the following errors:

    ```
    C:\>db2 connect to sample
    SQL1039C  An I/O error occurred while accessing the database
    directory. SQLSTATE=58031

    C:\>db2 connect to test
    SQL6048N  A communication error occurred during START or STOP
    DATABASE MANAGER processing.
    ```

    A. The database partition is not on the current machine. Issue the command on the machine where the partition is located.

12. Q. After the failover, it seems that some of my transactions are waiting on a lock.
    A. There might be indoubt transactions within your database. Issue the **db2 list indoubt transactions with prompting** command to manually resolve the indoubt transactions. For more information about indoubt transactions, see the DB2 Information Center.

13. Q. I ran the **db2mscs** utility with the **–u** parameter to decluster my instance, but the command failed.
    A. When declustering an instance, ensure that you run the **db2mscs** utility from the

instance-owning partition. If the instance is only partially declustered, continue the process using the manual steps that are described in Appendix C of this paper.

14. Q. There seems to be a general problem with file server access. What should I do?
    A. Ensure that you registered the netname value in DNS. If DB2_EXTSECURITY is configured, a Computer Object representing the new cluster must exist in the Active Directory. You must also add this computer object to the DB2ADMNS group and reboot the clustered machines for this change to take effect. For full details, see Appendix E.

**Tip:** You can trace the execution of the **db2mscs** utility by executing the following command:

```
db2mscs -f:db2mscs.cfg.ese -d:trace.out
```

This trace can help IBM support perform problem determination. To isolate the cause of an issue, you can also try configuring manually, without the **db2mscs** utility (see Appendix D).

For runtime issues, you can generate the Windows Server Failover Clustering log by issuing the following command:

```
cluster log /g
```

# Appendix C. Instance declustering

You can decluster your DB2 administration server instance or DB2 instance by using the **db2mscs** utility or manually. The following examples show how to decluster the instance from the mutual-takeover multiple-partition configuration.

## Using the db2mscs utility to decluster an instance

1. Back up the database in instance DB2MPP.
2. If you used DB2 drive mapping or if you must place the database on a different drive, drop the database.
3. Put all the DB2 Groups on the machine that they were originally on after you executed the **db2mscs** utility.
4. From the instance-owning partition, run the **db2mscs** utility with the `–u` option. Decluster the DB2 administration server before declustering the DB2 Enterprise Server Edition instance.

   ```
   db2mscs –u:db2das00
   db2mscs –u:db2mpp
   ```

5. Restore the databases that you backed up.

## Manually declustering an instance

1. Back up the databases in instance DB2MPP.
2. If you used DB2 drive mapping or if you must place the database on a different drive, drop the database.
3. In Failover Cluster Management, take only the DB2 resources offline, leaving the disks, IP addresses, network name, and file share resources online.
4. Drop the DB2 administration server and the DB2MPP instance from one of the machines in the cluster. This action drops them from all machines.

   ```
   db2admin drop
   db2idrop db2mpp
   ```

   Ensure that you save any instance information that you might require in the future, such as database manager configuration parameter settings and DB2 profile variable settings. When you drop an instance, all instance information is lost.

   ```
   db2 get admin cfg > admincfg.out
   db2 get dbm cfg > db2cfg.out
   db2set -all > db2set.out
   ```

5. In Failover Cluster Management, drop the DB2 Server resources corresponding to each DB2 partition, the DB2 administration server, and the corresponding IP addresses, network name, and file share.

6. Move all Physical Disk resources back to their initial groups.
7. Drop the groups that are associated with each partition, because no resources exist within these groups.
8. If you configured no other DB2 instances for the MSCS cluster, drop the DB2 resource type. Issue the following command from one machine in the cluster:

   ```
   db2wolfi u
   ```

9. Re-create the DB2 administration server and the DB2MPP instance.

10. Restore the databases on the new instance.

11. Reconfigure the configuration parameters as required.

**Note:** If you want to keep the instance and undo the MSCS migration only, you can replace Step 4 with the following command to change the MSCS instance back to a non-MSCS instance:

```
db2iclus unmigrate
```

## Appendix D. Manual configuration of an instance

The following example shows how to manually configure a single-node instance called MYESE. You should normally use the **db2mscs** utility to configure an instance. However, decomposing the steps might help during problem determination if errors occur when you use the **db2mscs** utility. The result of the following manual steps is the same as that of the "Hot-standby single-partition configuration (DB2 Enterprise Server Edition)" example.

1. Install MSCS and DB2 Enterprise Server Edition on all machines. In this example, the cluster name is *w32clust*, and the machine names are *w32cl1* and *w32cl2*.

2. Create an instance called MYESE:

   ```
   db2icrt MYESE -s ese -P \\w32cl1\k$\sqllib -u mydom\db2admin,passwd

   C:\>db2nlist /s
   List of nodes for instance "MYESE" is as follows:
   Node: "0" Host: "w32cl1" Machine: "w32cl1" Port: "0" - "stopped"
   ```

3. If the MYESE instance is running, stop it by issuing the **db2stop** command.

4. From *w32cl1*, install the DB2 Server resource type:

   ```
   c:>db2wolfi i
   ok
   ```

   If the **db2wolfi** command returns `Error : 183`, the resource type is already installed. To confirm that, you can drop the resource type and add it again. The resource type will not be displayed in Failover Cluster Management if it is not installed:

   ```
   c:>db2wolfi u
   ok
   c:>db2wolfi i
   ok
   ```

   **Tip:** The **cluster** command shows all valid resource types. You can verify that the DB2 Server resource type was created by issuing the **cluster restype** command:

   ```
   C:\>cluster restype
   Listing all available resource types:

   Display Name                         Resource Type Name
   ------------------------------------ ------------------------------------
   DB2 Server                           DB2 Server
   DHCP Service                         DHCP Service
   Distributed File System              Distributed File System
   Distributed Transaction Coordinator  Distributed Transaction Coordinator
   File Server                          File Server
   File Share Quorum Witness            File Share Witness
   ```

```
Generic Application                  Generic Application
Generic Script                       Generic Script
Generic Service                      Generic Service
IP Address                           IP Address
IPv6 Address                         IPv6 Address
IPv6 Tunnel Address                  IPv6 Tunnel Address
iSNSClusRes                          Microsoft iSNS
(Resource Type Unavailable)          MSMQ
(Resource Type Unavailable)          MSMQTriggers
Network Name                         Network Name
NFS Share                            NFS Share
Physical Disk                        Physical Disk
Print Spooler                        Print Spooler
Volume Shadow Copy Service Task      Volume Shadow Copy Service Task
WINS Service                         WINS Service
```

5. Create a new “Services and Application” group, as follows:

   a. From Failover Cluster Management, under “Services and Applications”, click **Create Empty Service or Application**.
   b. Right-click the property of the “New Service or Application,” and change the name to *MYESEGroup*.
   c. Set the preferred owner to the machine that the group is located on.

6. Add Storage to the new group:

   From Failover Cluster Management, right-click on *MYESEGroup*, and click **Add storage** to add *Disk U:* into *MYESEGroup*.

7. Add a Client Access Point and IP Address resource to the new group:

   a. From Failover Cluster Management, right-click MYESEGroup, click **Add a resource**, and then click **1-Client Access Point.**
   b. In the **Name** field, enter `MYNETNV`. MYNETNV will be the network name.
   c. In the **Address** field, enter `9.26.118.169`. This will be a highly available IP address, and this address should not correspond to any machine on the network.
   d. Bring the IP Address resource online.
   e. Ensure that you can ping the address from a remote machine.

8. Create a file share for the DB2Insprof disk:

   a. Create a subdirectory on disk U called `db2profs`.
   b. From Failover Cluster Management, right-click MYESEGroup, and click **Add a shared folder**. The Provision a Shared Folder Wizard (MYNETNV) window is displayed.
   c. Under Location, choose `u:\db2profs`, and click **Next**.
   d. Under “Share name” for SMB, enter `DB2MSCS-MYESE`, and click **Next**, **Next** … to finish creating the shared folder. A file server resource named FileServer-

(MYNETNV)(DISK U:) is created.
e. Verify that the file server has dependencies on Network Name MYNETNV and DISK U: by right-clicking **Properties > Dependencies**. The file server resource should be online automatically after creation.

f. Right-click the shared folder DB2MSCS-MYESE, and change the permissions:
add the correct share permissions and NTFS permissions.

**Tip:** When running DB2 Enterprise Server Edition instances on a cluster, it is best to use Domain Groups for DB2 extended security. If you use DB2 extended security, during the DB2 installation, you may have enabled Operating System Security (by default, it is enabled), and provided the name of the DB2ADMINGROUP and the name of the DB2USERSGROUP. If so, then you must add the administration users to the DB2ADMINGROUP and regular users to the DB2USERSGROUP. In this sample configuration, the name of the DB2ADMINGROUP is mydom\db2admns, and the name of the DB2USERSGROUP is mydom\db2users. Files and folders that the DB2 software creates are protected by ACLs that are based on these groups. Because you are creating the shared folder with Failover Cluster Management (not through the DB2 software), you must manually add the correct ACLs to this folder. If you do not do this, the DB2 code cannot access the content of this shared folder, which contains the DB2 instance directory. If DB2 code cannot access this folder, many things will fail.

In this example, under Share Permission, add the following groups:

- mydom\db2admns: Full control
- mydon\db2users: Change and Read
- Administrators: Full control
- CREATOR OWNER: Full control
- SYSTEM: Full control
- LOCAL SERVICE: Full control
- NETWORK SERVICES: Full control

Apply the same settings for NTFS Permissions.

**Tip**: You must add the Computer Object representing the Windows 2008 cluster, in this example, **W32CLUST$,** to the mydom\db2admns group (see Appendix B – Q14 and Appendix E).

9. Verify that the shared folder is accessible:

```
C:\>dir   \\MYNETNV\DB2MSCS-MYESE

Directory of \\MYNETNV\db2mscs-myese

05/23/2009  02:49 PM    <DIR>          .
05/23/2009  02:49 PM    <DIR>          ..
06/03/2009  03:58 PM    <DIR>          MYESE
```

10. Create a DB2 Server resource corresponding to the DB2 partition of type DB2 server. Since the instance used is MYESE, the resources must be named MYESE-0.

    a. From Failover Cluster Management, right-click MYESEGroup, and click **Add DB2 Server**. (Note: If the DB2 Server resource is not available, verify that the **`db2wolfi i`** command in step 4 was successful.)
    b. Change the properties of the new DB2 Server resource.
       i. Change the resource name to MYESE-0, to correspond to the instance name of MYESE.
       ii. Add dependencies. The DB2 Server resource depends on everything else in MYESEGroup.
       iii. Under Advanced Policies, enable "Run this resource in a separate Resource Monitor."

    Do not bring the new DB2 server resource online yet.

11. From W32CL1, use the **`db2iclus`** command to transform the DB2 instance into a clustered instance.

    ```
    C:\>db2iclus migrate /i:myese /c:w32clust /m:w32cl1
    /p:\\mynetnv\db2mscs-myese
    ```

    This step also places the instance directory into the newly created shared folder.

12. From W32CL1, use the **`db2iclus`** command to add the remaining cluster machines to the DB2 cluster list:

    ```
    C:\> db2iclus add /i:myese /c:w32clust /m:w32cl2 /u:mydom\db2admin,passwd

    DBI1912I  The DB2 Cluster command was successful.
    ```

13. Optional: Configure MYESEGroup for failback, by using Failover Cluster Management.

14. Configure DB2 Enterprise Server Edition such that when MYESE-0 goes offline, the DB2 software forces off all connections:

    ```
    C:\> db2set DB2_FALLBACK=YES
    ```

15. Bring all DB2 resources online by using Failover Cluster Management.

16. Create or restore all databases, putting all data on the shared drives.

17. Test the failover configuration.

## Appendix E. Changes in Windows 2008 Server Cluster

Because of changes in the Failover Clustering feature of Windows Server 2008, the following additional setup might be required:

In Windows Server 2008 failover clusters, the Windows cluster service is run under a special Local System account, whereas in Windows Server 2003, the Windows cluster service was run under an administrator account. This change affects the operations of the DB2 resource (db2server.dll), which is run under the context of the cluster service account.

- As of DB2® Version 9.5 Fix Pack 3, you can use Windows® Server 2008 failover clusters for failover support in partitioned DB2 database environments. In partitioned database environments, if you set the **`DB2_EXTSECURITY`** registry variable to `YES` on a Windows failover cluster, the DB2ADMNS and DB2USERS groups must be domain groups.

- When a multiple-partition instance is running on a Windows failover cluster, the **`INSTPROF`** path must be set to a network path (for example, `\\NetName\DB2MSCS-DB2\DB2PROFS`). A network path is automatically used if you use the **`db2mscs`** command to cluster the DB2 database system. If you are forming the DB2 Enterprise Server Edition cluster manually, you must manually create a Network Name, File Server, and a shared folder. This is described in an example in this appendix.

  When a Windows Server 2008 failover cluster is formed, a computer object representing the new cluster is created in the Active Directory. For example, if the name of the cluster is W32CLUST, a computer object named W32CLUST$ is created in the Active Directory. If you cluster a multiple-partition instance and the **`DB2_EXTSECURITY`** registry variable is set to `YES`, you must add the computer object to the DB2ADMNS group. For example, if the DB2 Administrator group is MYDOMAIN\DB2ADMNS, you must add the computer object W32CLUST$ to MYDOMAIN\DB2ADMNS. You must add the computer object so that the DB2 resource DLL can access the `\\NetName\DB2MSCS-DB2\DB2PROFS` path. This step is required whether you are using the **`db2mscs`** utility or a manual method to form the DB2 Enterprise Server Edition cluster.

- In Windows Server 2008 Failover Clustering, the cluster file share resource is no longer supported. The cluster file server is used instead. The file share (a regular file share) is based on the cluster file server resource. Microsoft requires that the cluster file servers that you create in the cluster use DNS for name resolution. In a multiple-partition instance, a file server resource is required to support the file share. The values of the **`NETNAME_NAME`**, **`NETNAME_VALUE`**, and **`NETNAME_DEPENDENCY`** parameters in the `db2mscs.cfg` file are used to create the file server and file share resources. The netname is based on an IP address, and this netname must be in DNS.

For example, if a `db2mscs.cfg` file contains the following parameters, a file share named \\MSCSV\DB2MSCS-DB2 is created:

```
...
```

```
NETNAME_NAME = MSCSN
NETNAME_VALUE = MSCSV
...
```

The name MSCSV must be registered in DNS. Otherwise, the file server or the file share that was created for the DB2 cluster fails when DNS resolution is not successful.