Implementing IBM â DB2 â Universal Database Ô V8.1 Enterprise Server Edition with Microsoft â Cluster Server

January 2003

Aslam Nomani
DB2 UDB System Verification Test

Trademarks

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States and/or other countries: IBM, DB2, DB2 Universal Database.

Windows and Windows-based trademarks and logos are trademarks or registered trademarks of Microsoft Corp.

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.

Contents

Abstract

IBM® DB2® Universal Database™ (UDB) is the industry's first multimedia, Web-ready relational database management system, strong enough to meet the demands of large corporations and flexible enough to serve medium-sized and small e-businesses. DB2 UDB combines integrated power for business intelligence, content management, and e-business with industry-leading performance and reliability. This, coupled with Microsoft® Cluster Server (MSCS) , strengthens the solution by providing a highly available computing environment.

MSCS facilitates the automatic failover of applications and data from one system to another in the cluster after a hardware or software failure.

A complete high-availability (HA) setup includes many parts, one of which is the MSCS software. A good HA solution includes planning, design, customization, and change control. In the event of failure, a high-availability solution reduces the amount of time that an application is unavailable by removing single points of failure.

This document takes you through sample configurations using DB2 UDB Enterprise Server Edition (ESE) V8.1 using Microsoft Windows 2000® Advanced Server.

This paper is not intended to provide a detailed understanding of MSCS or DB2 UDB. We assume you are already familiar with both MSCS and DB2 UDB. It is our intent in this paper to provide an understanding of how DB2 UDB ESE integrates into the MSCS environment and how to configure DB2 within that environment.

About the Author

Aslam Nomani has been working with the IBM Database Technology team in the Toronto Laboratory for six years. For the past five years, he has worked in the DB2 UDB System Verification Test department. Aslam is currently focused on high-availability solutions.

Introduction

DB2 is dependent 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 important to understand the resources 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 as 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 the application is continuously available.

Failover capability allows for the automatic transfer of workload from one machine to another when there is hardware failure. Microsoft Cluster Server (MSCS) provides the ability to failover resources between multiple machines. These resources include such items as disks, IP addresses, file shares, and network names. DB2 uses the ability of MSCS to create additional resource types to develop a resource type called DB2. By grouping various resources together using the MSCS 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) will failover and restart on another machine.

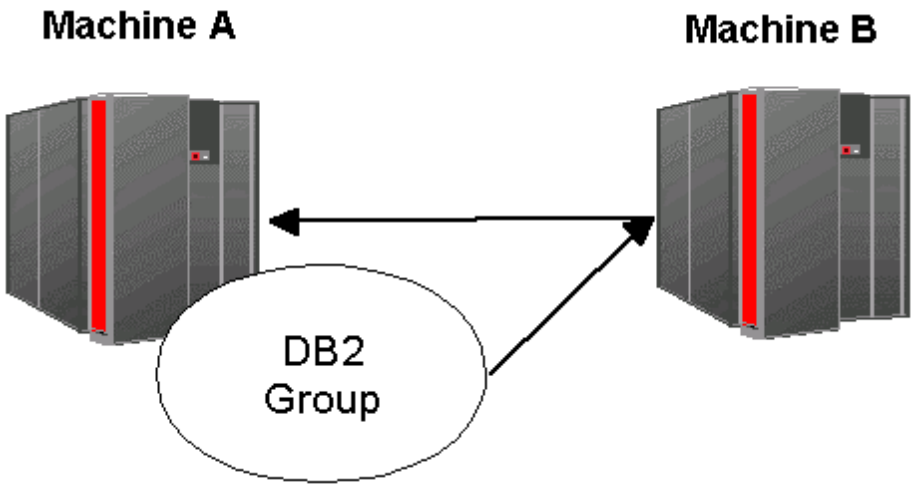


Figure 1. DB2 group floating between machines

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

DB2 UDB ESE overview

DB2 UDB ESE is an edition of DB2 UDB that allows users to create single-partition or multiple-partition database environments. DB2 UDB ESE uses a highly scalable shared-nothing architecture that allows users to spreads data across multiple database partitions that may reside on different physical machines. The data on each partition can be processed in parallel across partitions as well as in parallel within each partition. If a partition fails and a query requires data from that partition, then that query will fail. DB2 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 may not be desirable for many environments. Thus, the failed partition must be restarted so users can have access to the data on that partition.

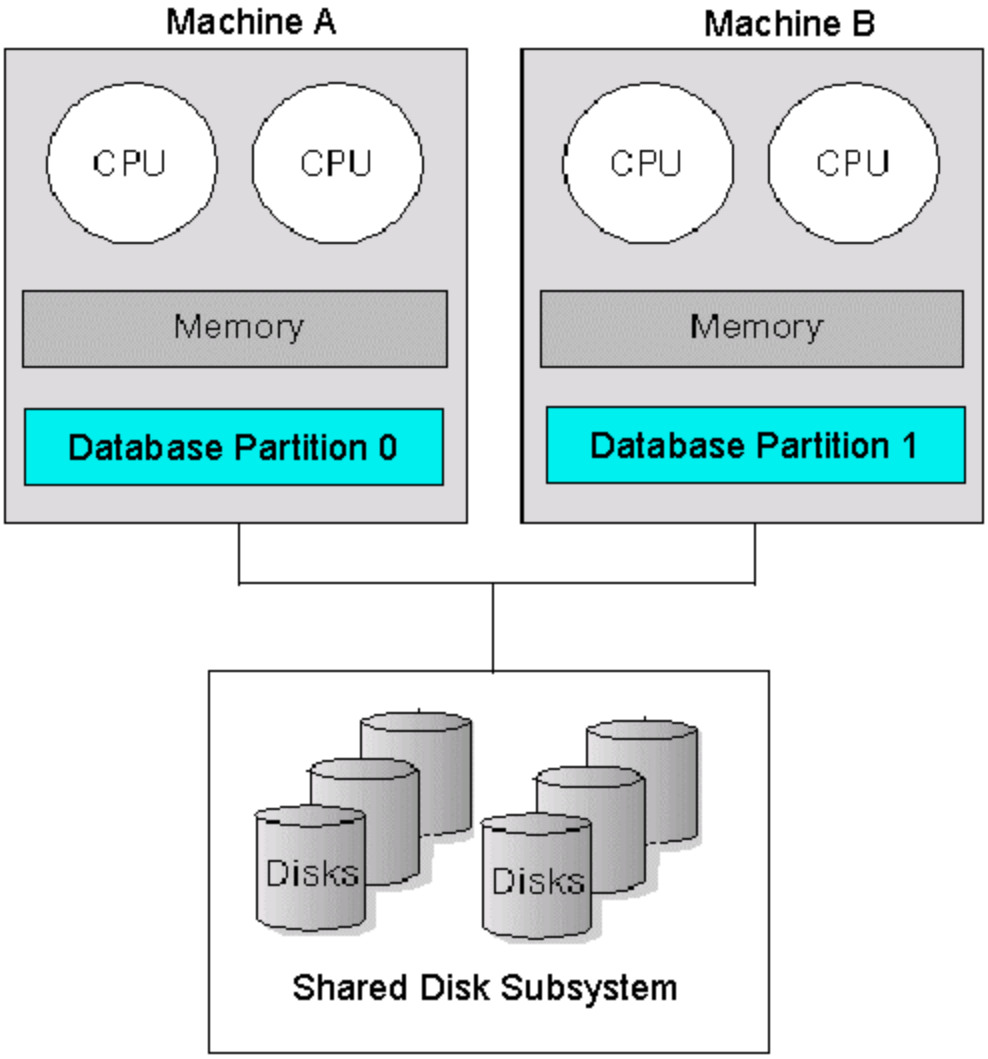


Figure 2. Typical topology of DB2 UDB ESE with multipe partitions and MSCS

Notes:

- In a single-partition environment, 'Database Partition 1' would not exist in Figure 2.
- Even though both database partitions are connected to the same disk subsystem, each partition will only access data that it owns.
- DB2 UDB uses MSCS to allow database partitions to failover in the event of a failure. This allows all partitions to be highly available and thus all data to be available.

To illustrate how DB2 UDB ESE works within the MSCS environment, we will go through a simple example of a DB2 instance that is comprised of two partitions similar to the configuration in Figure 2. With a single-partition ESE instance, only Partition 0 will exist in the `db2nodes.cfg` file.

- Initially Partition 0 is active on Machine A and we will assume the data for Partition 0 is stored in the shared disk subsystem on Disk E.
- Initially Partition 1 is active on Machine B and we will assume the data for Partition 1 is stored in the shared disk subsystem on Disk F. The initial `db2nodes.cfg` will look as follows:

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

  Note: The `db2nodes.cfg` file stores the DB2 partition information. For more information about the `db2nodes.cfg` file, please refer to the DB2 UDB documentation. For a single partition instance, the IP address in the fourth field of the `db2nodes.cfg` is not needed.

- If Machine B fails, Partition 1, Disk F and TCP/IP address 10.1.1.6 will failover to Machine A, resulting in both Partition 0 and Partition 1 active on Machine A. Partition 0 will still access the data on Disk E and Partition 1 will still access the data on Disk F. DB2 will automatically update the hostname and computer name associated with Partition 1 in the configuration file that stores the partition information [`db2nodes.cfg`]. The `db2nodes.cfg` file will now look as follows:

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

  Note: Partition 1 has the host name and computer name changed to `macha` automatically by DB2. Also, DB2 has changed the port number associated with Partition 1 to alleviate any conflicts. The TCP/IP address does not change as it is a highly available address that moves along with the DB2 partition.

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

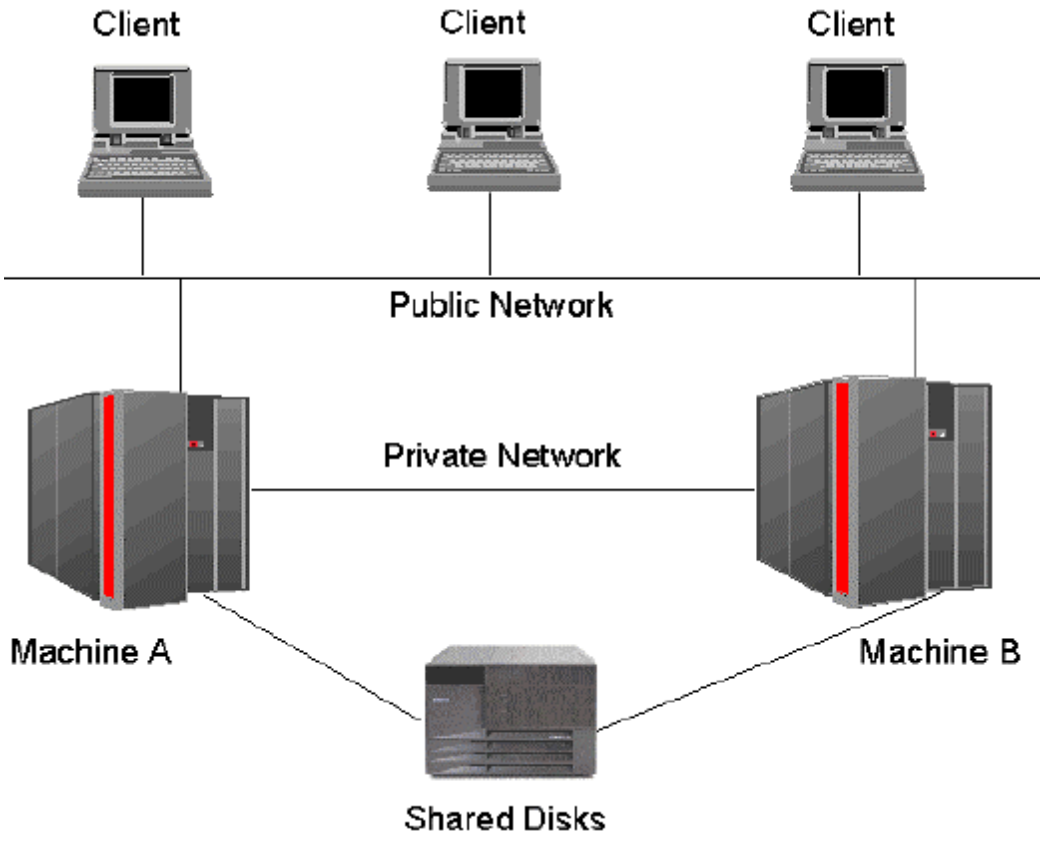
- If remote clients were connected to Partition 0, they may have to reissue any uncommitted transactions at the time of the failure. If the uncommitted transaction did not require any information from the failing partition, then the transaction will not have to be reissued.
- If remote clients were connected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

  Note: The default behaviour of the DB2 client is to connect to the partition that has port 0. Thus, a connect to Partition 1 will actually be a connect to Partition 0 after the previous failover has occurred.

Conceptual overview

Since an MSCS environment allows multiple machines to take ownership of the same disk resource, this disk must have shared direct access from all machines in the cluster. The cluster also maintains a heartbeat between the nodes to determine which nodes in the cluster are available. The heartbeat communication usually flows through an internal private network while remote clients access the cluster via a public network. Thus, a typical cluster topology may appear as follows:

Figure3. Typical cluster configuration



Upon successfull installation of MSCS, Cluster Administrator, a graphical administration tool that is part of MSCS will show the machines in the cluster, the resources, the resource types, the groups, and the networks available to the cluster.

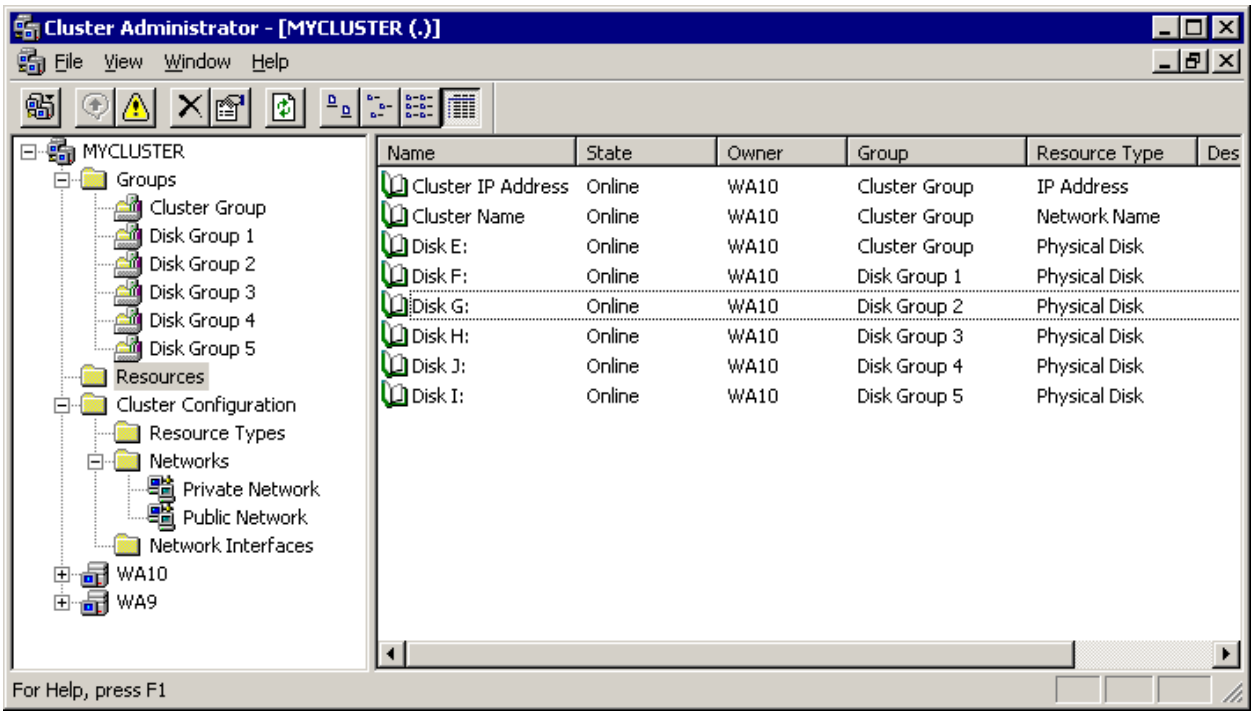


Figure 4. Cluster Administrator after initial installation of MSCS

The screenshot in Figure 4 is an initial snapshot of the cluster immediately after MSCS has been installed on a two-node MSCS configuration. It should be noted that Windows 2000 Datacenter supports as many as four nodes in a cluster while Windows NT Enterprise Edition and Windows 2000 Advanced Server only support a *maximum* of two nodes in a cluster. Windows .NET will support up to eight nodes. If a DB2 UDB ESE instance spans more nodes than are available in a single cluster, then multiple clusters can be used. The cluster shown in Figure 4 will be used as the starting point for examples used throughout this paper. Here are some of the notable items about the cluster:

- The name of the cluster is MYCLUSTER.
- The two machines in the cluster are WA9 and WA10.
- A Cluster Group and five other groups labelled Disk Group 1 through 5 exist.
- Each Disk Group contains one Physical Disk resource.
- Two networks exist; they are labelled “Private Network” and “Public Network.”
- Currently the Cluster Group is active on machine WA10 along with all the Disk Groups.

Tip: Ensure all groups within the MSCS configuration can failover successfully between all machines in the cluster before proceeding with configuring DB2 within the MSCS environment.

Within the default available resource types available to the cluster, there is no resource type that corresponds to a DB2 partition. DB2 UDB creates a resource type called DB2, and each resource of type DB2 corresponds to a DB2 partition. Since DB2 now integrates into the MSCS environment with a resource type that can be monitored by MSCS, MSCS can now ensure that DB2 stays online, along with all the resources required by DB2.

The DB2 resource type is automatically created when the DB2MSCS utility is executed. The DB2MSCS utility will be discussed in a later section.

Because each DB2 partition requires disks to store information, an IP address for internal communication (when using a multiple partition instance), as well as optionally an IP address to allow for remote connections, DB2 uses the group feature within MSCS to group multiple resources into a single logical entity. The combination of a DB2 resource, disks, and TCP/IP addresses represent almost all the resources required to succesfully run a DB2 ESE instance. A DB2 ESE instance will also use a highly available network name and file share to store instance information that will be 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 failover between machines.

A single partition or a grouping of partitions can be contained within a single MSCS group. If partitions reside in the same group, they will always reside on the same machine at the same time. If it is desired to have different partitions on different machines at the same time, then the partitions should be placed within different MSCS groups.

As already noted, each MSCS group with DB2 partitions contains one or more DB2 resources, disks, and IP addresses. The group with the instance owning partition will also contain a network name and file share to store configuration information for the entire ESE instance. The instance owning partition is always Partition 0. The order that these resources come online is critical when the group is brought online. If the DB2 resource starts first, it may encounter failures because the partition may require access to files on a disk that is not online yet. Thus, the dependency feature within MSCS is utilized. The dependency feature allows the ability to define which resources must be completely started before attempting to start another resource. The DB2 resource is automatically configured to be dependent on the disks as well as the IP addresses (along with the network name and file share if the DB2 resource corresponds to the instance owning partition). This dependency also applies to the stopping process. MSCS will now ensure the DB2 resource is completely stopped before any attempt to bring the disks and IP addresses offline (as well as the network name and file share if the DB2 resource corresponds to the instance owning partition).

Failover and failback

MSCS is responsible for deciding whether resources and groups are restarted on the current machine or whether they should failover to another machine in the cluster. It is very important that the cluster is aware of which resources have been started so it knows which resources it must try to keep online. If you bring resources or groups online using Cluster Administrator, then MSCS is aware that these resources have been started and will attempt to ensure they stay available in the case of a failure. If DB2 is started using a non-cluster interface (DB2START, NET START, or an automatic start from Service Manager), then MSCS is not aware that the DB2 partition has been started and will make no attempt to keep the DB2 partition up and running.

MSCS will monitor all resources and groups that are brought online using Cluster Administrator. If a machine in the cluster fails, MSCS will move all resources and groups to another cluster machine and ensure that any resources that were online will be brought back online. When a resource fails, MSCS will attempt to bring that resource back online on the current machine first and if this continues to fail, it will move the whole group associated with the resource to another cluster machine and try to bring it online. The number of times MSCS will retry to bring the resource online is configurable within Cluster Administrator. The machine preference in regards to where the group will failover is also configurable within Cluster Administrator. Failures of the DB2 resource could occur because of exceptions within DB2 or because an operating system resource has run low. Because failure detection of DB2 is triggered by termination of the DB2 process, a hang situation will not automatically trigger a restart of the DB2 resource.

When a failover occurs due to a machine failure or other cause, database partitions may be in a transactionally inconsistent state. When the database partitions starts up on the surviving machine, it must go through a crash recovery phase that may 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 will not be completely available until crash recovery has completed.

In a mutual takeover environment, it is very important to plan for the highest potential machine requirements if all MSCS groups are online on a single machine at any given time. If the machine is not capable of handling the workload, the results may range from performance degradatations to further abnormal terminations.

Failback is the ability for an MSCS group to move back to its preferred machine once that machine is back online within the cluster. The term “fallback” is also commonly used when referring to failback. 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 database connections may be forced off the database. MSCS drives the failback based on configurations within Cluster Administrator, with the default behavior being not to failback.

In Version 8 of DB2, the default behaviour of DB2 is to allow failback. If automatic failback is desired, configure the cluster so that the default behaviour is to allow failback. To change the default behaviour of DB2 to not allow failback, adjust the DB2_FALLBACK DB2 profile variable.

DB2MSCS utility

The DB2MSCS utility is a standalone command line utility used to transform a non-HA instance into an HA instance. The utility will create all MSCS groups, resources, and resource dependencies. It will also copy all DB2 information stored in the Windows registry to the cluster portion of the registry as well as moving the instance directory to a shared cluster disk. The DB2MSCS utility takes as input a configuration file provided by the user specifying how the cluster should be set up. The DB2MSCS utility should be run from the instance owning partition. The fields within the configuration file that are used for DB2 ESE are as follows:

DB2_INSTANCE The name of the DB2 instance. If the instance name is not specified, the default instance (the value specified by the DB2INSTANCE environment variable) is used. This parameter has a global scope and should be specified only once in the `DB2MSCS.CFG` file.

DAS_INSTANCE The name of the DB2 Administration Server instance. This parameter has a global scope and should be specified only once in the `DB2MSCS.CFG` file. This paramater can not be used in conjunction with DB2_INSTANCE.

CLUSTER_NAME The name of the MSCS cluster. All the resources specified following this line are created in this cluster until another CLUSTER_NAME parameter is specified.

DB2_LOGON_USERNAME The username of the domain account for the DB2 service (i.e., domain\user). This parameter has a global scope and should be specified 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 and should be specified only once in the `DB2MSCS.CFG` file.

GROUP_NAME The name of the MSCS group. If this parameter is specified, a new MSCS group is created i into this group until another GROUP_NAME parameter is specified. Specify this parameter once for each group.

DB2_NODE The node number of the database partition server (or node) to be included in the current MSCS 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 MSCS group.

The value for this keyword can optionally contain the network name (or IP address) that is used by DB2 for inter-partition communication. Typically when running in an MSCS 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, while the public network is used for remote client connections. To ensure that DB2 always uses the private network for inter-partition communication, you can explicitly specify the highly available network name (or IP address) that is associated with the private network as follows:

```
DB2_NODE = <node_number> <network_name>
```

IP_NAME The name of the IP Address resource. The value for the IP_NAME is arbitrary, but it must be uni recommended name is the hostname that corresponds to the IP address.

IP_ADDRESS The TCP/IP address for the IP resource specified by the preceding IP_NAME parameter. This

IP_SUBNET The TCP/IP subnet mask for the IP resource specified by the preceding IP_NAME parameter. This parameter is required if the IP_NAME parameter is specified.

IP_NETWORK The name of the MSCS network to which the preceding IP Address resource belongs. This parameter is optional. If it is not specified, the first MSCS network detected by the system is used. The name of the MSCS network must be entered exactly as seen under the Networks branch in Cluster Administrator.

Note: The previous four IP keywords are used to create an IP Address resource.

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 value for the Network Name resource. This parameter must be specified if the NETNAME_NAME parameter is specified.

NETNAME_DEPENDENCY The name for the IP 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 it is not specified, the Network Name resource has a dependency on the first IP resource in the group.

DISK_NAME The name of the physical disk resource to be moved to the current group. Specify as many disk resources as you need. The disk resources must already exist. When the DB2MSCS utility configures the DB2 instance for failover support, the instance directory is copied to the first MSCS disk in the group. To specify a different MSCS disk for the instance directory, use the INSTPROF_DISK parameter. The disk name used should be entered exactly as seen in Cluster Administrator.

INSTPROF_DISK An optional parameter to specify an MSCS disk to contain the DB2 instance directory. If this parameter is not specified the DB2MSCS utility uses the first disk that belongs to

the same group.

INSTPROF_PATH This is an alternate way to specify a path on the MSCS disk to contain the DB2 instance directory. Use this paramater if the DB2MSCS utility is unable to obtain the drive letter of the disk resource.

TARGET_DRVMAP_DISK An optional parameter to specify the target MSCS disk for database drive mapping. This parameter will specify the disk the database will be created on by mapping it from the drive the create database command specifies. If this parameter is not specified, the database drive mapping must be manually registered using the DB2DRVMP utility.

DB2_FALLBACK An optional parameter to control whether or not the applications should be forced off when the DB2 resource is brought offline. If you do not want the application to be forced off, then set DB2_FALLBACK=NO. The default value for DB2_FALLBACK is YES.

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

SERVICE_NAME The service name of the Generic Service resource. The paramater must be specified if the SERVICE_DISPLAY_NAME paramater is specified.

SERVICE_STARTUP An optional startup paramater for the Generic Service resource.

Example configuration files will be shown in subsequent sections of this paper.

Tip: Ensure the IP address used for IP_ADDRESS is a new IP address that does not already belong to any machine on the network. Also ensure all values used for DISK_NAME and IP_NETWORK are entered exactly as seen in Cluster Administrator.

Drive mapping

Drive mapping is a mandatory step in implementing a DB2 database across a multiple-partition instance if the partitions reside in multiple MSCS groups. The DB2 `create database` command requires a drive specification for where the database should be created, and if this is not specifed a default value will be used. If we choose to create a database on Disk E, then that disk drive must be available to all partitions. If partitions are spread across multiple groups, it is not possible to have a shared disk drive with the same disk letter exist within multiple groups.

Based on the example in Figure 5, if the group with Partition 0 owned Disk E, and the group with Partition 1 owned Disk F, then we do not have a shared drive availabe to both partitions with the same drive letter. Thus to alleviate this issue, DB2 has implemented a drive mapping mechanism that is used for the `create database` command. In our example we can can map Disk E to Disk F on Partition 1. Then when we do the `create database` command on Disk E, the data for Partition 0 will be created on Disk E and the data for Partition 1 will be created on Disk F. Alternatively, we could create the database on Disk F and on Partition 0 we would map Disk F to Disk E. Drive mapping is automatically performed when the TARGET_DRVMAP_DISK keyword is specifed in the DB2MSCS input file, or can be done manually using the db2drvmp command.

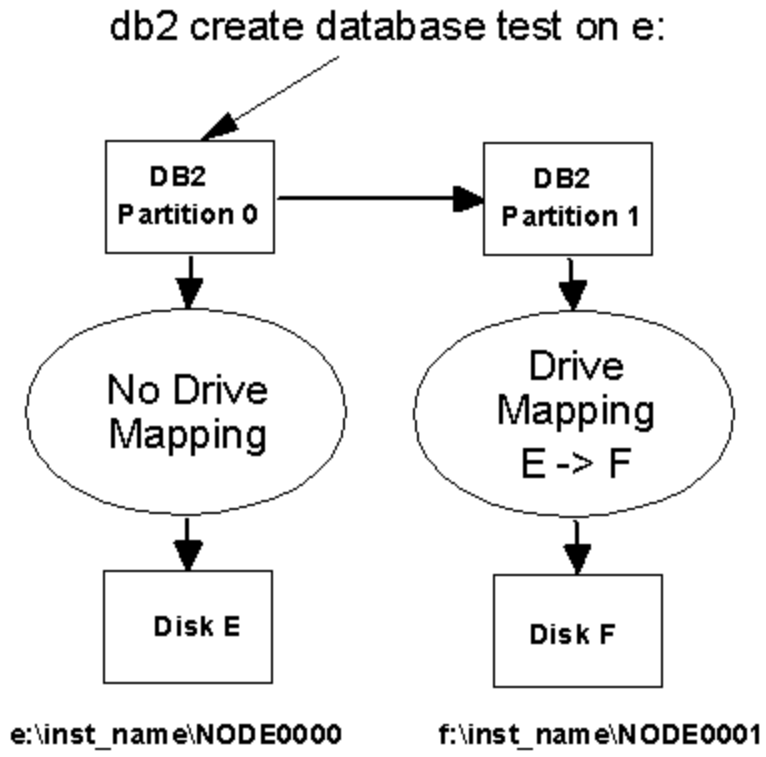


Figure 5. DB2 drive mapping

Note: In Figure 5, the `create database` command could have been issued from either Partition 0 or Partition 1.

Planning and preparation

The first step in planning for your ESE high-availability environment is to determine whether you want a hot standby configuration, or a mutual takeover configuration, or a combination of both. You should then define the MSCS cluster configuration based on the requirements of your ESE instance. For example, in the hot standby configuration described in this paper, a two-node MSCS cluster was used with one machine being used as a hot spare.

The next step is determining if multiple partitions will always failover together. If multiple partitions do failover together, they should be placed in the same MSCS group. For the purposes of this paper, we will refer to any MSCS group that contains one or more partitions as a DB2 group. Partitions that reside in the same DB2 group can share disk resources as well as TCP/IP resources. The preferred machine owner of each DB2 group should then be determined along with the failover preferences of the DB2 group. Then, decide whether automatic failback is desired within your environment.

Each DB2 group will require one or more MSCS disk resources to store information for the partitions within that same DB2 group. Allocate enough disk resources that will be needed to satisfy the requirements of your database.

For a multiple partition instance residing in multiple groups, each DB2 group should have one MSCS TCP/IP resource on the private network. This TCP/IP resource will be used to tell DB2 which network it should use for internal communication.

Optionally, one or more DB2 groups may need a MSCS TCP/IP resource defined on the public network. This TCP/IP resource will be used if remote clients are directly connecting to partitions within the same DB2 group. Only partitions that will be used to receive incoming 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: For all TCP/IP resources defined for use with DB2 in an MSCS environment, it is important that they are static IP addresses. All the TCP/IP resources should be registered in your Domain Name Server or exist within the hosts file on each machine in the cluster.

For each DB2 instance, determine the maximum number of DB2 partitions that can reside on any one machine at any one time. Set the DB2 environment variable DB2_NUM_FAILOVER_NODES to one less than the maximum number of partitions that can reside on one machine. The default value is two and it is not necessary to set this value less than the default. For example, if there is a possibility of four partitions residing on one machine, issue the following command:

```
db2set DB2_NUM_FAILOVER_NODES=3
```

Hot standby single-partition configuration

In a hot standby configuration, at least one machine in the MSCS cluster is idle and dedicated as a backup 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 6, Partition 0 is active on one machine with a single hot spare available if the current machine fails.

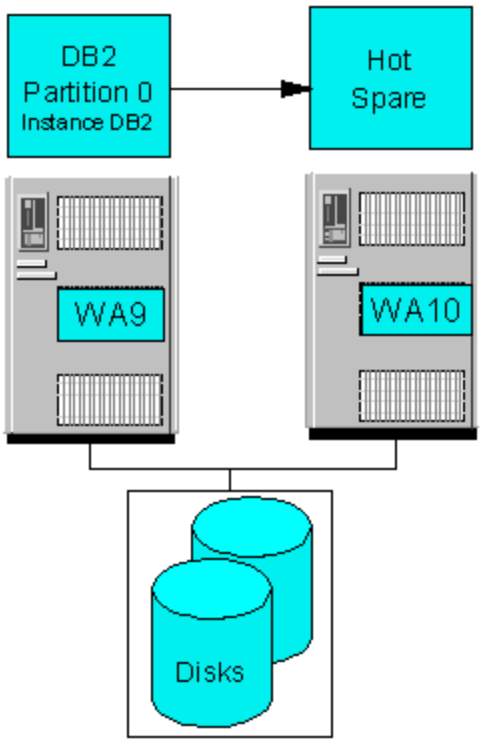


Figure 6. Hot standby configuration

The following example will detail the steps required to set up a hot standby configuration using a DB2 single-partition instance. The DB2MSCS utility should be executed from a domain user that has local Administrator authority on each machine in the cluster.

After DB2 UDB has been installed on each machine in the cluster, reboot each machine before proceeding with the DB2MSCS configuration.

1. Configure the MSCS cluster and ensure it is healthy.
2. Install DB2 UDB ESE on every machine in the cluster and allow the install to configure a single-partition instance. The installation of DB2 must be on a local drive. (Steps 1 and 2 can

be done in reverse order if necessary). The installation will create an instance called DB2 on each machine. Since the second machine (WA10) will be a hot standby machine, the DB2 instance on this node must be removed using the db2idrop command.

```
C:\>db2idrop db2
```

Becausee 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.

Note: The instance DB2 created by the install will have four ports reserved in the services file for inter-partition 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.

Ensure the instance is stopped on the primary machine (WA9) using the DB2STOP command. Instance DB2 will be used to configure the HA instance. An input configuration file must be configured for use with the DB2MSCS utility. Before you transform the instance to become an HA instance, note that the instance directory is currently stored on the local drive where DB2 was installed:

```
C:>db2set db2instprof
C:\SQLLIB
```

The input configuration file for the DB2MSCS utility will appear in the following form:

```
DB2_INSTANCE=DB2
CLUSTER_NAME=MYCLUSTER
DB2_LOGON_USERNAME=mydom\db2admin
DB2_LOGON_PASSWORD=xxx

GROUP_NAME=DB2 Group 0
DB2_NODE=0
IP_NAME=mscs11
IP_ADDRESS= 9.26.75.25
IP_SUBNET=255.255.255.0
IP_NETWORK=Public Network
IP_NAME=ether0
IP_ADDRESS=10.1.1.1
IP_SUBNET=255.0.0.0
IP_NETWORK=Private Network
NETNAME_NAME=mynetname
NETNAME_VALUE=mynetname
NETNAME_DEPENDENCY=ether0
```

```
DISK_NAME=Disk F:
```

To run the DB2MSCS utility, specify the -f option for the file name and ensure the command returns successfully:

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

Note: The NETNAME_DEPENDENCY in the DB2MSCS input file is configured to use a TCP/IP address defined on the private network.

Tip: Back up all existing databases within an instance before transforming it to a high-availability clustered instance.

The execution of the DB2MSCS utility transforms the instance DB2 to a clustered instance that resides within MYCLUSTER. "DB2 Group 0" is created, which contains a single partition, two new IP addresses, and one disk resource. "DB2 Group 0" also contains a network name and file share since it contains the instance-owning partition. The instance directory is moved to the new file share in "DB2 Group 0." The `db2ilist` command also now indicates that the instance is clustered by showing `C:<cluster name>` after the instance name:

```
C:>db2set db2instprof
 \\mynetname\DB2MSCS-DB2
c:\>db2ilist
 DB2             C : MYCLUSTER
```

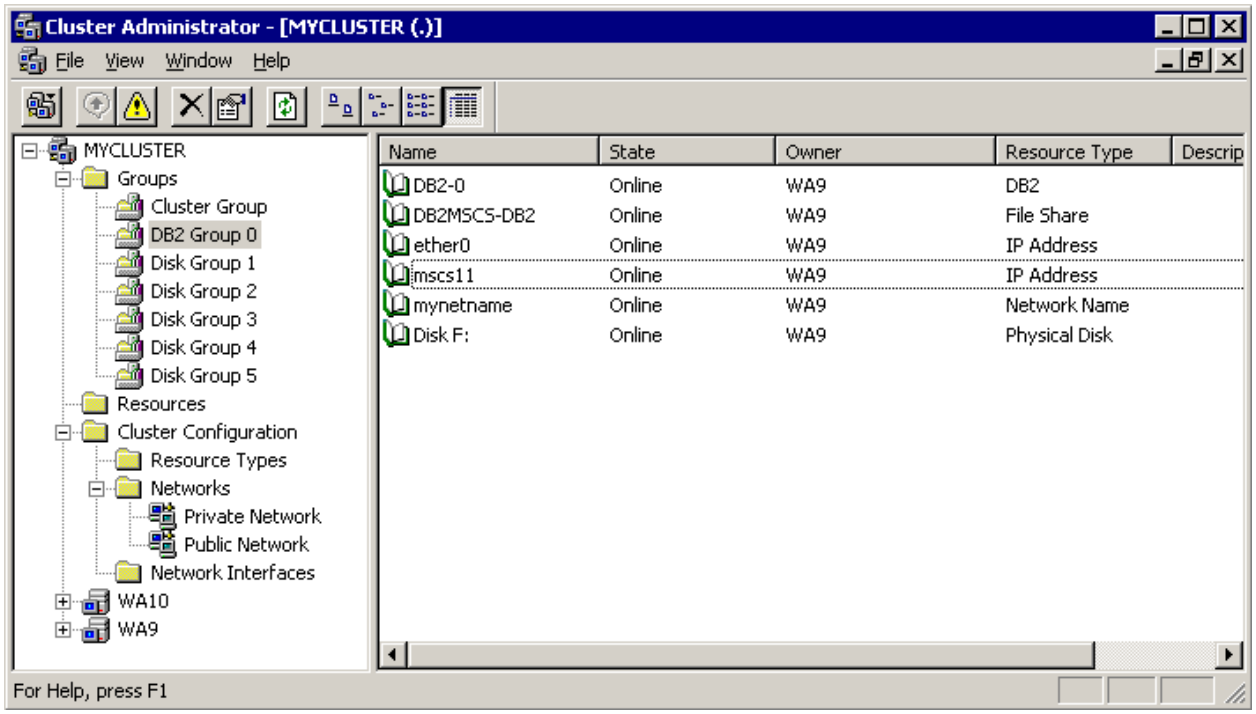


Figure 7. Cluster Administrator with instance DB2

The new screen shot of Cluster Admnistrator shows the following:

- A Group called "DB2 Group 0" is created.
- "DB2 Group 0" contains one disk resource.
- "DB2 Group 0" contains two IP addresses.
- A resource of type DB2 has been created for the DB2 partition.
- A network name and file share has been created in "DB2 Group 0" to hold the instance directory. This network name has been created on the "Private Network."

3. The next step is to determine which machine is intended as the primary machine and which machine is intended as the standby machine. In our case we will pick WA9 as the preferred owner of "DB2 Group 0" and WA10 as the hot spare. The properties for "DB2 Group 0" are initially configured with no preferred owners. From Cluster Administrator, modify this configuration for "DB2 Group 0" to specify a preferred owner.

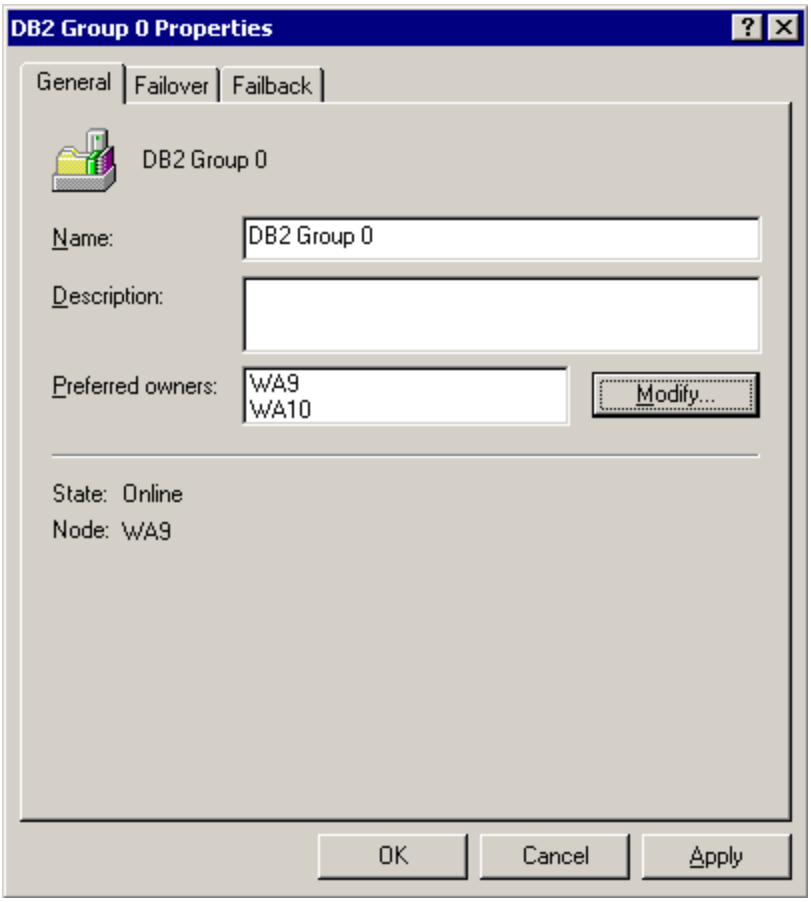


Figure 8. Cluster Administrator properties box for "DB2 Group 0"

4. After specifying the preferred owner, you must determine if you want the DB2 group to automatically failback to the preferred machine after it recovers from a failure. If you don't want automatic failback, then no further actions are required because the default behavior is to prevent failback. If you do want automatic failback, then you must modify the properties box for each DB2 group.

   On the **Failback** tab, select **Allow failback**. If you want to enable failback when the preferred machine is back online, set **Allow failback** to **Immediately.** If you only want to enable failback for a particular period of the day, for example between 1 A.M. and 8 A.M., configure that in the **Failback between** options. If the preferred machine comes back online during this time window, then the failback will occur. If the preferred machine comes back online at 12:59 A.M., the group will not automatically failback at 1 A.M.

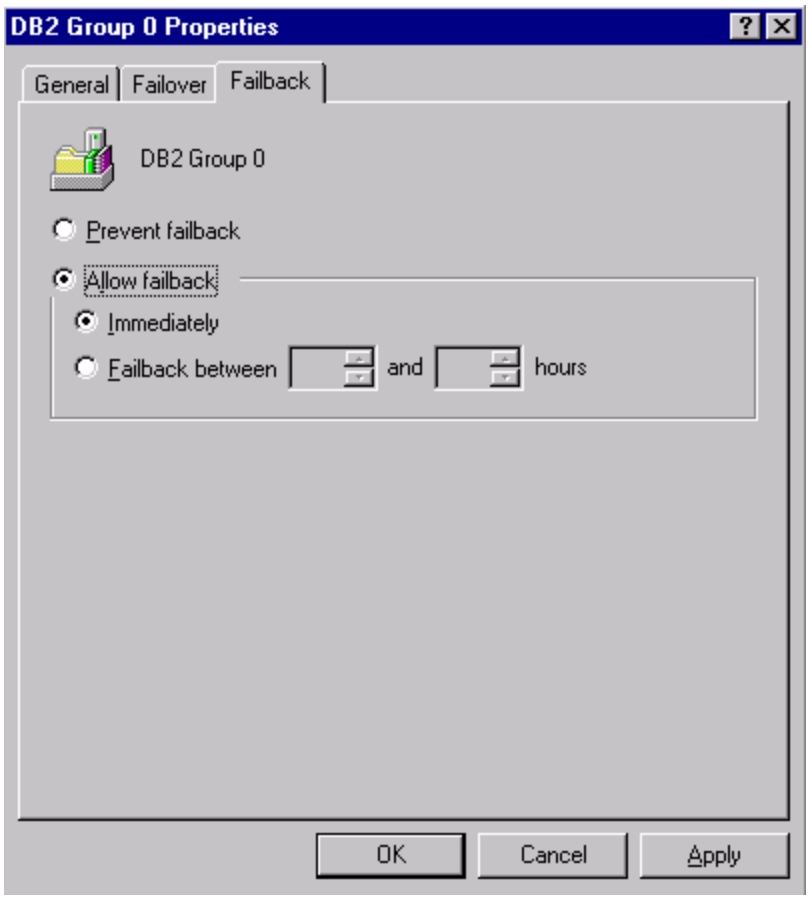


Figure 9. Failback settings for "DB2 Group 0"

- Now that you have configured the instance DB2, you should place all databases within the instance on the disk drives that exist within the same group as instance DB2. For instance DB2, all data should reside on Disk F. If existing databases are being used and they currently do not reside on Disk F, use a redirected restore to move the data to the shared drive. For information on how to perform the redirected restore, refer to the DB2 Universal Database documentation. If you create a new database, make sure you create the database and its tablespaces on Disk F. If the DB2 instance fails over to another machine and does not have access to all of its data files, expect to receive database errors.

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

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

Mutual takeover single-partition configuration

A mutual takeover single-partition configuration has a DB2 instance running on each machine in the cluster. If a machine fails in the cluster, the result will be multiple DB2 instances running on the same machine. This scenario will configure two single-partition instances in the same cluster, which will require the instance names to be different. You must plan carefully to ensure the surviving machine in the cluster is capable of handling the workload that will be placed on it.

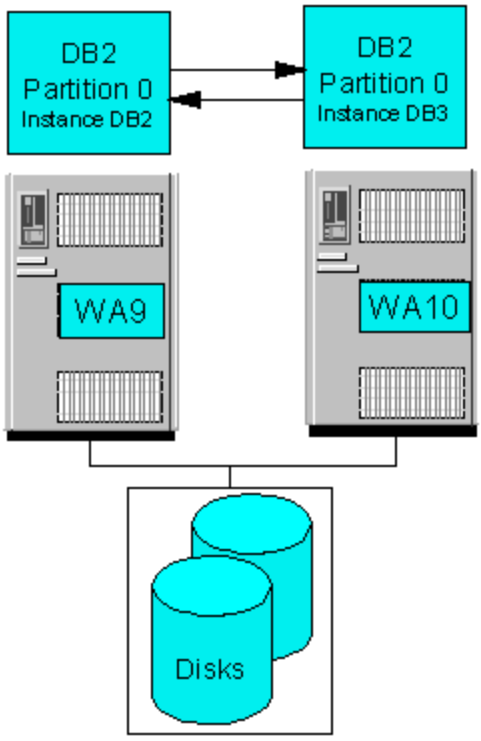


Figure 10. Mutual takeover single-partition configuration

Configuring a mutual takeover environment is very similar to configuring a hot standby environment, except now a second instance will be configured. The following example will detail the steps required to set up a mutual takeover configuration.

1. Perform all the steps in the prior section for configuring a hot standby single partition configuration. This will create an instance called DB2 with its preferred owner being machine WA9.
2. Create a DB2 UDB ESE instance called DB3 on machine WA10. Ensure the ports reserved for DB2 inter-partition communication are available on all machines in the cluster.

```
  c:\>db2icrt db3 -s ese -p n:\sqllib -u
mydom\db2admin,xxx  -r 60004,60007
  c:\>db2ilist
  DB3
  DB2           C : MYCLUSTER
```

The output from db2ilist shows that there are two instances, DB3 and DB2. The information following the output for DB2 states that it is a clustered instance and it is in the cluster called MYCLUSTER. Currently, DB3 is not clustered and does not exist on machine WA9.

3. Create the input file for the DB2MSCS utility so it can transform DB3 to a clustered instance. The resources used for this group will have to be different from the resources used for instance DB2, because instance DB2 and instance DB3 will not necessarily reside on the same machine. The input configuration file for the DB2MSCS utility will look like the following example:

```
DB2_INSTANCE=DB3
CLUSTER_NAME=MYCLUSTER
DB2_LOGON_USERNAME=mydom\db2admin
DB2_LOGON_PASSWORD=xxx

GROUP_NAME=DB2 Group 1
DB2_NODE=0
IP_NAME=mscs12
IP_ADDRESS=9.26.75.26
IP_SUBNET=255.255.255.0
IP_NETWORK=Public Network
IP_NAME=ether1
IP_ADDRESS=10.1.1.2
IP_SUBNET=255.0.0.0
IP_NETWORK=Private Network
NETNAME_NAME=mynetname1
NETNAME_VALUE=mynetname1
NETNAME_DEPENDENCY=ether1
DISK_NAME=Disk G:
DISK_NAME=Disk H:
```

To run the DB2MSCS utility, specify the -f option for the input file name and ensure the command executes successfully. Execute the DB2MSCS utility on machine WA10, as that is where the instance DB3 currently exists.

```
c:\>db2mscs –f:db2mscs.cfg.db3
DB21500I The DB2MSCS command completed successfully.
```

The configuration of instance DB3 creates a group called “DB2 Group 1” with a different IP address from the one that was created for instance DB2 and with Disk G and Disk H that are also not being used by instance DB2. The field INSTPROF_DISK was not used for configuring DB3 like it was used for instance DB2. Thus for DB3, INSTPROF_DISK will default to the first disk identified, which will be Disk G. The output of db2ilist will also show that both instances are now clustered

```
c:\>set db2instance=DB3
c:\>db2set db2instprof
\\mynetname1\DB2MSCS-DB3
c:\>db2ilist
DB3      C : MYCLUSTER
DB2      C : MYCLUSTER
```

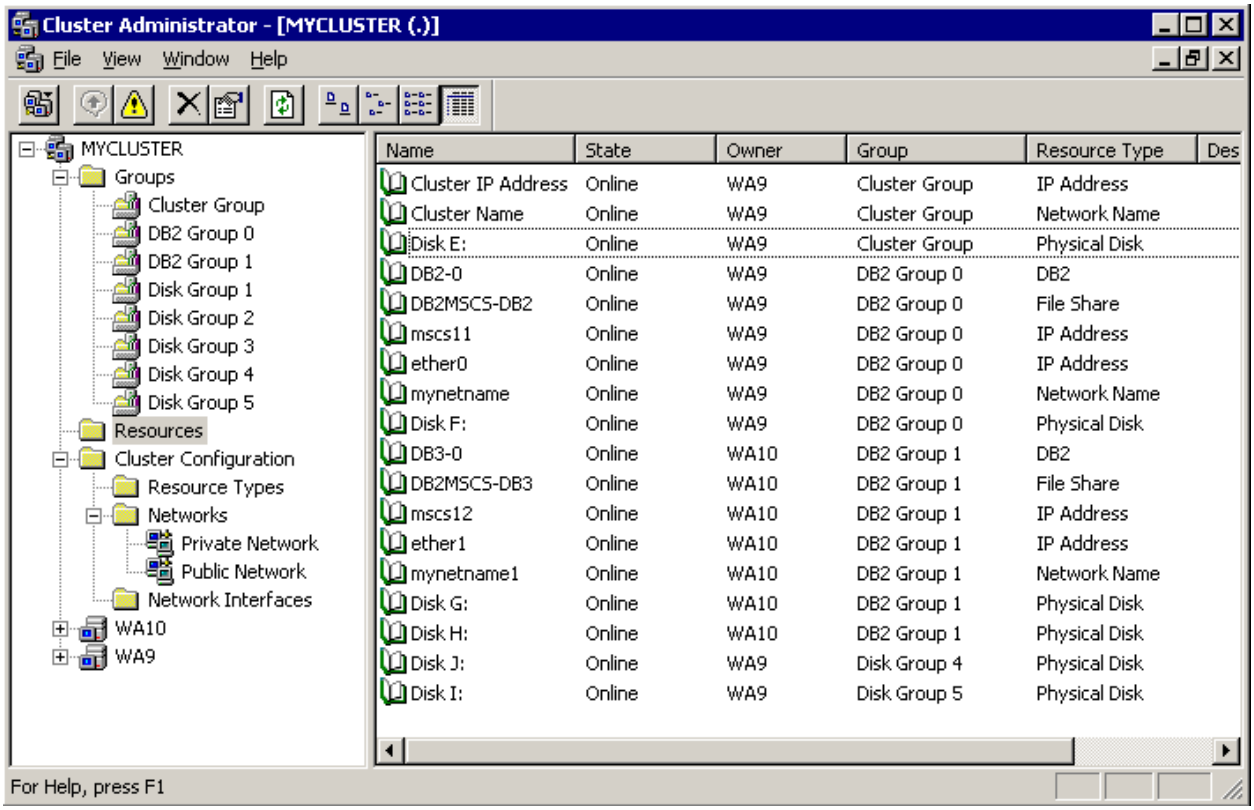


Figure 11. Cluster Administrator with instance DB3 in “DB2 Group 1”

Figure 11 shows the following additions to the initial hot standby setup:

- A group called “DB2 Group 1” is created.
- Disk G and Disk H have been moved into “DB2 Group 1”.
- IP addresses “mscs12” and “ether1” have been created and resides in “DB2 Group 1”.
- A resource called “DB3-0” of type DB2 has been created and resides in “DB2 Group 1”.

- "DB2 Group 1" currently resides on machine WA10.

4. Because this configuration is a mutual takeover configuration and instance DB2 is configured to use WA9 as its primary machine, instance DB3 should be configured to use WA10 as its primary machine. The properties for DB3 Group should be modified to represent WA10 as the primary machine and WA9 as the secondary machine.

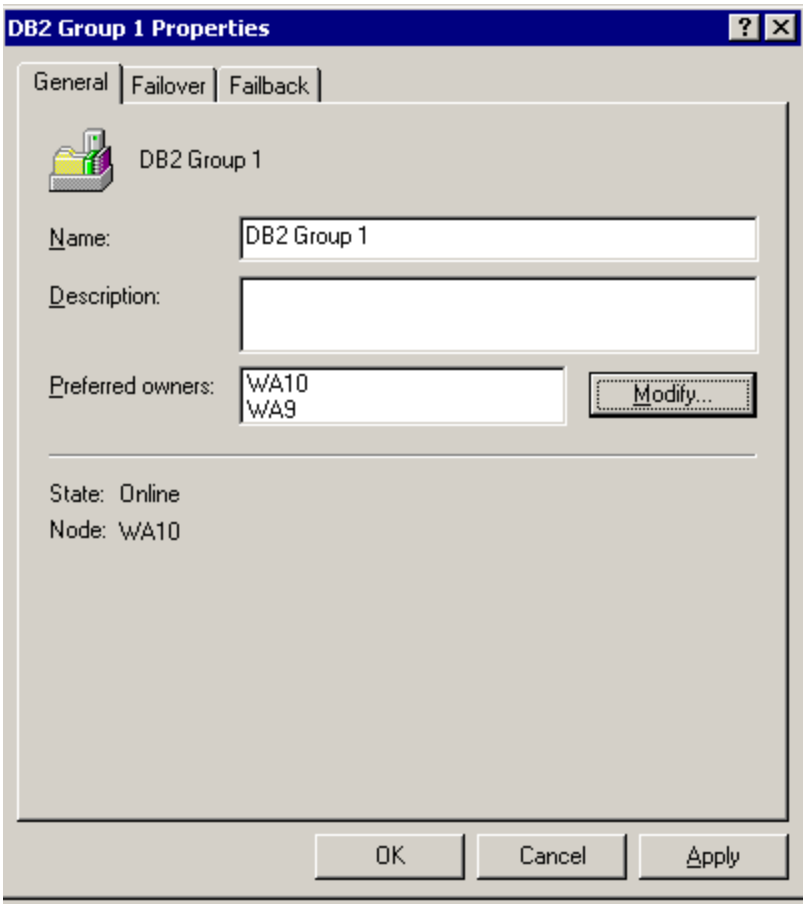


Figure 12. Properties box for "DB2 Group 1"

5. After the preferred owner is configured for "DB2 Group 1," you must determine if you want automatic failback. If you do want automatic failback, set the failback options under the **Failback** tab for "DB2 Group 1."

6. The final step ensures that all databases and corresponding tablespaces associated with DB3 reside on either Disk G or Disk H. If an old database is being used and does not already exist on Disk G or Disk H, a redirected restore will be needed. Refer to the DB2 UDB documentation for details regarding redirected restore. If a new database is being created,

ensure the database and its tablespaces are created on either Disk G or Disk H.

```
c:\>db2 create db sampleb on G:
DB20000I  The CREATE DATABASE command completed
successfully.

c:\>db2 create tablespace tsb in nodegroup
ibmdefaultgroup managed by database using(file
‘h:\container1' 50000) on node(0)
DB20000I  The SQL command completed successfully.
```

The previous command was executed on WA10 and DB2INSTANCE was set to point to DB3. DB3 does not currently reside on WA9, so any attempts to execute DB2 commands against DB3 on WA9 will fail.

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 will be multiple DB2 partitions running on the same machine. You must plan carefully to ensure each machine in the cluster is capable of handling the workload that will be placed on it.

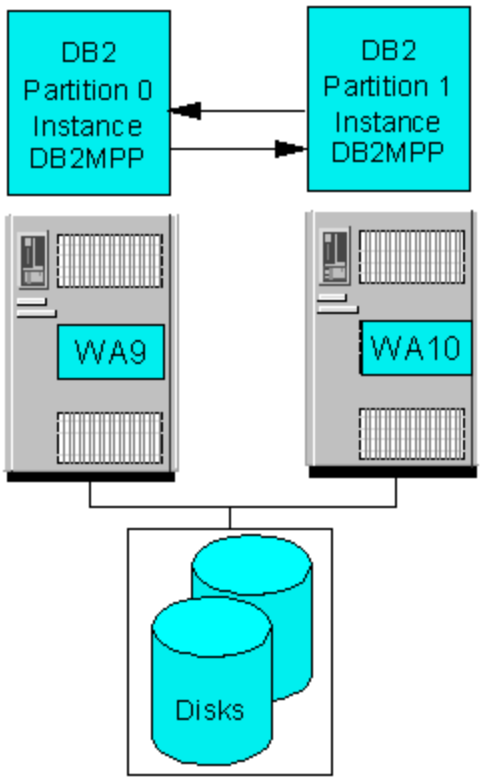


Figure 13. Mutual takeover multiple-partition configuration

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

1. Initially, you must install MSCS and configure and stop a two-partition DB2 instance.

```
C:\>db2nlist /s
List of nodes for instance "DB2MPP" is as follows:
Node: "0" Host: "wa9" Machine: "wa9" Port: "0" - "stopped"
```

```
Node: "1" Host: "wa10" Machine: "wa10" Port: "0" -
"stopped"
```

Create a DB2MSCS input file and then execute the DB2MSCS utility using this input file. This configuration will assume that only Partition 0 will receive incoming remote client requests and thus will have an MSCS TCP/IP resource defined on the Public Network.

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

GROUP_NAME=DB2 Group 0
DB2_NODE=0 10.1.1.1
IP_NAME=mscs11
IP_ADDRESS=9.26.75.25
IP_SUBNET=255.255.255.0
IP_NETWORK=Public Network
IP_NAME=ether0
IP_ADDRESS=10.1.1.1
IP_SUBNET=255.0.0.0
IP_NETWORK=Private Network
NETNAME_NAME=mynetname
NETNAME_VALUE=mynetname
NETNAME_DEPENDENCY=ether0
DISK_NAME=Disk F:
TARGET_DRVMAP_DISK=Disk F:

GROUP_NAME=DB2 Group 1
DB2_NODE=1 10.1.1.2
IP_NAME=ether1
IP_ADDRESS=10.1.1.2
IP_SUBNET=255.0.0.0
IP_NETWORK=Private Network
DISK_NAME=Disk G:
TARGET_DRVMAP_DISK=Disk G:
```

After running the db2mscs utility with the above configuration file, Cluster Administrator will show the following view of the cluster:

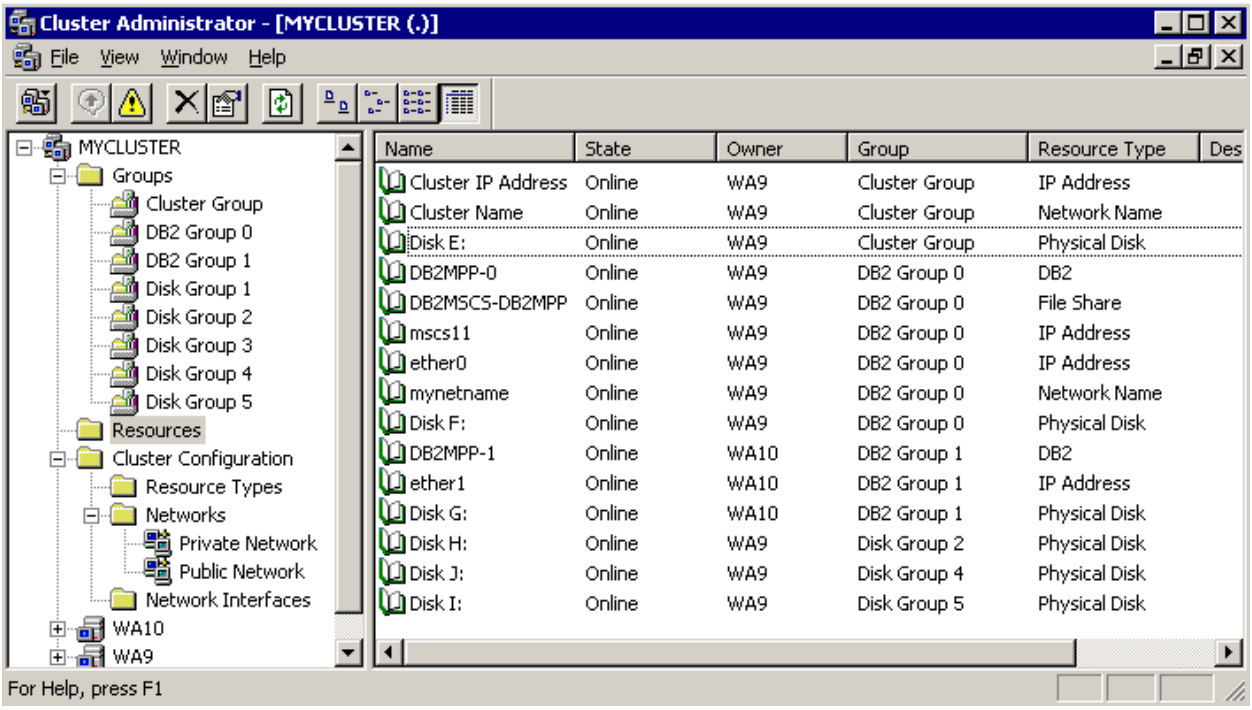


Figure 14. Cluster Administrator with instance DB2MPP clustered

2. Determine which machine is intended as the primary machine for each of the DB2 groups. In our case we will set the primary machine for "DB2 Group 0" as WA9 and the primary machine for "DB2 Group 1" as WA10. Also, configure automatic failback for the DB2 groups if desired.

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

Configuring the DB2 Administration Server

The DB2 Administration Server is used by the DB2 Control Center to administer DB2 instances and databases. If high availability of the Administration Server is desired, then it must also be clustered. The steps to cluster the Administration Server are similar to the steps used to transform any other regular instance. In the example below, you will use the Administration Server to administer the DB2MPP instance created in the Mutual Takeover Multiple Partition Configuration. The example below will also reuse the shared disk and IP address that was configured for Partition 0, resulting in Partition 0 and the Administration Server sharing these resources. Since they are sharing the same resources, you must place the Administration Server in the same group as Partition 0. Below are the instructions for configuring the DB2 Administration Server for instance DB2MPP:

1. Stop the Administration Server on all machines:

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

   If an Administration Server does not exist on WA9, use the command `db2admin create` to create this instance on WA9.

2. Drop the Administration Server on all cluster nodes except the first node by performing the following command on each machine:

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

3. On the first node (WA9) where the Administration Server resides, go into the Windows Services dialog box and modify the Administration Server instance so it is set to start manually. The name of the Administration server is DB2DAS00, so it will show up as DB2DAS-DB2DAS00 in the Services dialog box.

   Create a configuration input file to be used with the DB2MSCS utility to cluster the Administation Server:

```
DAS_INSTANCE=DB2DAS00
CLUSTER_NAME=MYCLUSTER
DB2_LOGON_USERNAME=mydom\db2admin
DB2_LOGON_PASSWORD=xxx
GROUP_NAME=DB2 Group 0
DISK_NAME=Disk F:
```

```
INSTPROF_DISK=Disk F:
```

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 Partition 0. Also, you are reusing Disk F and not configuring an IP address, because the IP address already configured for Partition 0 can be reused. Even though Disk F was already configured for DB2 Group 0, it must be specified again because that is where all information associated with the Administration Server will be placed. A Generic Service will be created which will allow the DB2 Administration Server to be monitored by MSCS.

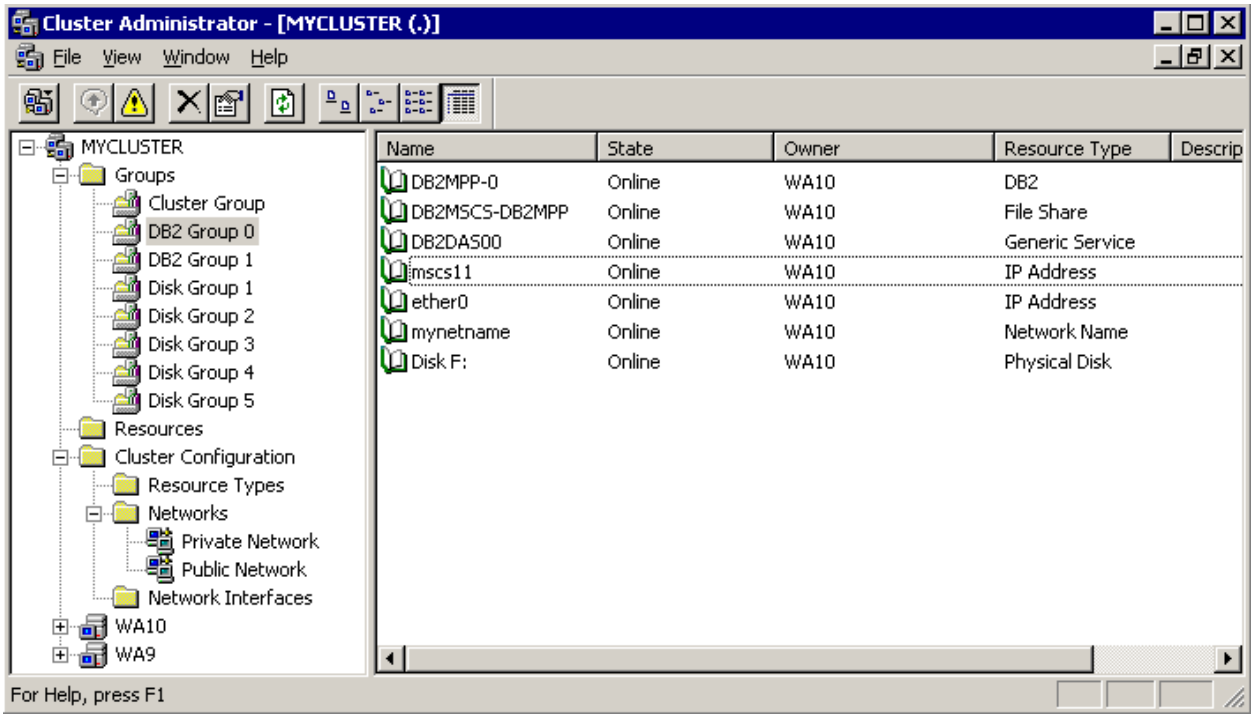


Figure 15. Cluster Administrator with DB2DAS00 added to DB2 Group 0

4. Execute the DB2MSCS utility from WA9:

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

5. On all clients used for DB2 administration, remove any references to the Administration Server using the DB2 Control Center. Then, use the DB2 Control Center to recatalog a reference to the Administration Server utilizing the highly available cluster IP address defined on the public network

The Administration Server is now integrated within the cluster. Care should be taken to ensure any remote client connections go through the highly available IP address and any scripts or data that are needed by the Administration Server are placed on the highly available disks associated with the DB2DAS00 instance. Cluster Administrator now shows instance DB2MPP configured in the cluster, and the group containing Partition 0 also contains the Administration Server. The above steps can also be used to cluster the Administration Server for instance DB2 in the single partition configuration examples.

Note: Because only one DB2 Administration Server can be running on a single machine at one time, we cannot cluster the DB2 Administration Server for each instance in a mutual takeover single-partition configuration.

Remote client connections

MSCS provides the ability to create a highly available TCP/IP resource. Remote clients should use the highly availabe IP address defined 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 configured for the cluster that is located in the same group as the partition 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, the coordinator functionality of DB2 is then also spread across multiple partitions. If an IP address is used that is associated with either a physical machine or an IP address in a different group than the partition, there is no guarantee the IP address will point to the machine where the database partition is actually residing.

The other factor that needs to be taken into consideration is that the client will connect to the server using a specified port. That identical port must be available to the instance on all machines within the cluster, and should be defined in the services file on each machine in the cluster. The following is an example of how to catalog connections from a remote client to database sample for instance DB2, which was created in the Hot Standby Single Partition section:

1. Add the following entries to the services file for all machines in the cluster to reserve unused ports for DB2:

```
db2cDB2         50000/tcp #connection port for the DB2 instance DB2
db2iDB2         50001/tcp #interrupt port for the DB2 instance DB2
```

2. Ensure DB2COMM is set to TCPIP for each partition:

```
C:\>set DB2INSTANCE=DB2
C:\> db2set DB2COMM=TCPIP
```

3. Update the database manager configuration for instance DB2 so it knows which port to listen on for incoming TCP/IP clients:

```
C:\>db2 update dbm cfg using svcename db2cdb2
```

4. For the changes in steps 2 and 3 to take effect, the partition must be brought offline and then back online. From Cluster Administrator, right click on the resource representing the DB2 partition and select **Take Offline**. After all DB2 resources are in an offline state, right click on the same resources and select **Bring Online**.

5. Now that all partitions are ready to receive incoming remote requests, the database partition must be cataloged from a remote machine. The services file on the remote client must be updated with the similar entries made on the server machines.

   To catalog database SAMPLE for instance DB2:

```
C:\>db2 catalog tcpip node nodea remote mscs11 server db2cdb2
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, `mscs11` can be replaced with its corresponding IP address in the cluster.

   Optionally, you can use the DB2 Client Configuration Assistant to manually catalog connections to the database using a graphical interface. If you use this mechanism, ensure the highly available IP address is used to catalog the database connection.

User scripts

DB2 provides the ability to execute a batch script before and after each DB2 resource is brought online as well as before and after each DB2 resource is brough offline. This batch script resides in the instance directory of each instance, so each instance will have a separate copy of these script files. To determine the instance directory, issue the db2set command shown below and then append the instance name to the results:

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

In this particular case, the instance directory for instance DB2MPP will be located under `\\mynetname\DB2MSCS-DB2MPP\DB2MPP`. The scripts that execute before and after each DB2 partition are brought online are called `db2cpre.bat` and `db2cpost.bat`, respectively. These scripts are also referred to as the pre-online and post-online scripts. The scripts that execute before and after each DB2 partition are brought offline are called `db2apre.bat` and `db2apost.bat`, respectively. 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 will only get executed if they exist. These batch files are launched by the MSCS Cluster Service and are run in the background. The script files must redirect standard output to record any output as a result of commands run from within the script file.

The pre-online script will execute and complete 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 MSCS will not timeout on its attempt to bring the DB2 resource online. Since the pre-offline script also runs synchronously before taking DB2 offline, you should ensure it executes efficiently so it does not significantly affect failback time.

Note: The user scripts are executed with the DB2INSTANCE and DB2NODE environment variable set to the values corresponding to the resource executing the script. The DB2NODE value corresponds to partition number. The pre-offline and post-offline scripts will not be executed if the DB2 process terminates abnormally.

Testing the configuration

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

The objective of this section is not to provide detail on how to test the whole system, but it will 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. The query should go against a table 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 due to a hard crash such as a machine failure, DB2 will not attempt to force off any database applications. Any uncommitted transactions that used the failing partition will be 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. Reissue steps 1 through 5 using various simulated hardware and software failures and use a client workload that more closely simulates the actual workload expected in the production environment.

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

Rolling upgrade

A rolling upgrade is the ability to upgrade software on the cluster while still keeping the application online. An MSCS environment ideally lends itself for performing a rolling upgrade of DB2 because the partitions can be online on one machine while the other machine is being upgraded. Rolling upgrades are supported for DB2 installs that do not require either database or instance migration.

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

The following example will demonstrate how to do a rolling upgrade of the mutual takeover multiple-partition configuration that was described earlier. Since different partitions of the same instance cannot run different levels of DB2 at the same time, care should be taken during the upgrade. Our strategy will be to move all partitions to the second half of the machines, and then upgrade the first half. We will then take all partitions offline, then move them back to the first half of the machines, and then bring them online. Finally, we can then upgrade the second half of the machines and move partitions back to their desired location.

1. Initially Partition 0 and 1 reside on machines WA9 and WA10, respectively. Move Partition 0 to WA10 so WA9 is idle.
2. Stop the cluster service on WA9 (net stop clussvc).
3. Apply the DB2 FixPak on WA9.
4. Bring all DB2 resource offline. Then start the cluster service on WA9 (net start clussvc). Move all DB2 groups to WA9 and then bring them online, resulting in WA10 being idle.
5. Stop the cluster service on WA10.
6. Apply the DB2 FixPak on WA10.
7. Start the cluster service on WA10.
8. Then move Partition 1 back to WA10 at a convenient time.

Repairing a cluster after catastrophic machine failure

When a machine is damaged beyond repair, the operating system must be reloaded or the machine replaced. Let's assume that machine WA10 is damaged beyond repair so that the operating system has to be reinstalled. The following example depicts the steps to repair instance DB2MPP and the Administration Server in the Mutual Takeover Multiple-Partition Configuration. Currently DB2MPP and the Administration Server are active on WA9 and clients can still fully access the databases.

1. From Cluster Administrator on WA9, right click on **WA10** and select **Evict Node** to remove WA10 from the cluster.
2. DB2 uses a DB2 profile variable called DB2CLUSTERLIST, which is used by DB2 to determine which machines are in the cluster. Initially, DB2CLUSTERLIST will show two machines in the cluster, as demonstrated by the following example using the instance DB2MPP:

   ```
   C:\>db2set DB2CLUSTERLIST
   WA9 WA10
   ```

   On machine WA9, remove machine WA10 from the cluster list for instance DB2MPP:

   ```
   set DB2INSTANCE=DB2MPP
   db2set DB2CLUSTERLIST= "WA9"
   ```

3. On machine WA10 reinstall the operating system.
4. On machine WA10 reinstall MSCS adding it back to the initial cluster, leaving all groups on machine WA9.
5. On machine WA10, reinstall DB2 UDB and drop any instances that may have been created by the install. Ensure you do not select the DB2 install option to add another node to an existing instance.
6. Create the Administration Server on WA10 if it does not exist and ensure it is not active:

   ```
   db2admin create
   ```

   Note: This step is only required if you are using a version of DB2 UDB V8.1 without any FixPaks.

7. From WA9, remove WA10 from the cluster list for the Administration Server. This will also remove the Adminstration Server created in the prior step and leave the highly available Administration Server that was initially configured.

   ```
   db2iclus drop /m:wa10 /DAS:db2das00 /c:mycluster
   ```

```
/u:mydom\db2admin,xxx
```

8. On machine WA9, add machine WA10 back to the cluster list for the instance DB2MPP and the Administration Server. This step will also create the necessary DB2 services on WA10.

```
db2iclus add /m:wa10 /u:mydom\db2admin,xxx /i:db2mpp
/c:mycluster
db2iclus add /m:wa10 /u:mydom\db2admin,xxx /DAS:db2das00
/c:mycluster
```

9. The groups containing DB2MPP partitions and the Administration Server are now ready to failover to machine WA10.
10. At an appropriate time, use the Move Group option in Cluster Administrator to ensure the groups containing the DB2MPP partitions and the Administration Server can successfully start on machine WA10.

If upgrading machine WA10 with a new machine, first move all groups that are on WA10 to WA9, and then follow steps 1-10.

Managing security

After DB2 is working properly under MSCS, you may want to specify which users can administer the cluster. To administer a cluster, users must have either administrative permissions on both nodes or specific permissions to administer the cluster. By default, the local Administrators group on all nodes have permissions to administer the cluster. To give a user permission to administer a cluster without giving the user Administrative permissions on all nodes:

1. Bring up Cluster Administrator.
2. Right-click on the cluster name, and then click **Properties.**
3. Click **Security** or **Permissions.**
4. Specify which users and groups may administer the cluster.

The users also must have access to the DB2 registry variables that are stored in the cluster registry under the `HKEY_LOCAL_MACHINE\Cluster\IBM\DB2\PROFILES` registry key. By default, the local Administrators group on all nodes have full control to the cluster registry. To give a user permission to access the DB2 registry variables:

1. Run `regedt32.exe`.
2. Expand the Cluster key until the `HKEY_LOCAL_MACHINE\Cluster\IBM\DB2\PROFILES` key is displayed.
3. Select the key, then click **Security.**
4. Click **Permissions.**
5. Specify which users and groups may need to run DB2.

DB2 authentication

We recommend that you use domain security (domain users and domain groups) so that when DB2 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 convention, using eight characters or less.
2. Add any domain users who will have DB2 SYSADM authority for this domain group.
3. From the machine that runs DB2, update the database manager configuration parameter SYSADM_GROUP to the name of the domain group.
4. Restart the DB2 instance.

Other sample configurations

You can configure DB2 in many different ways within an MSCS cluster. Below are a couple additional sample configurations.

Mutual takeover load-balancing configuration

The objective of this configuration is to ensure that if one machine fails, the workload will be equally distributed across the remaining machines. Care should be taken that each machine in cluster is capable of handling the potential resource requirements that may be needed from that machine. If all machines have the same configuration and each DB2 partition is used equally, then it would be desirable to spread the partitions across the active machines in an equal fashion. In our example, we will start with six DB2 partitions spread equally across a cluster containing only three machines.

If one machine in the cluster fails, it is not desirable to have the two failing partitions failover to the same machine. If that was the case, then we would have four DB2 partitions on one machine and two DB2 partitions on another machine. Thus, the two partitions on each machine should be configured to failover to different machines as illustrated in Figure 16.

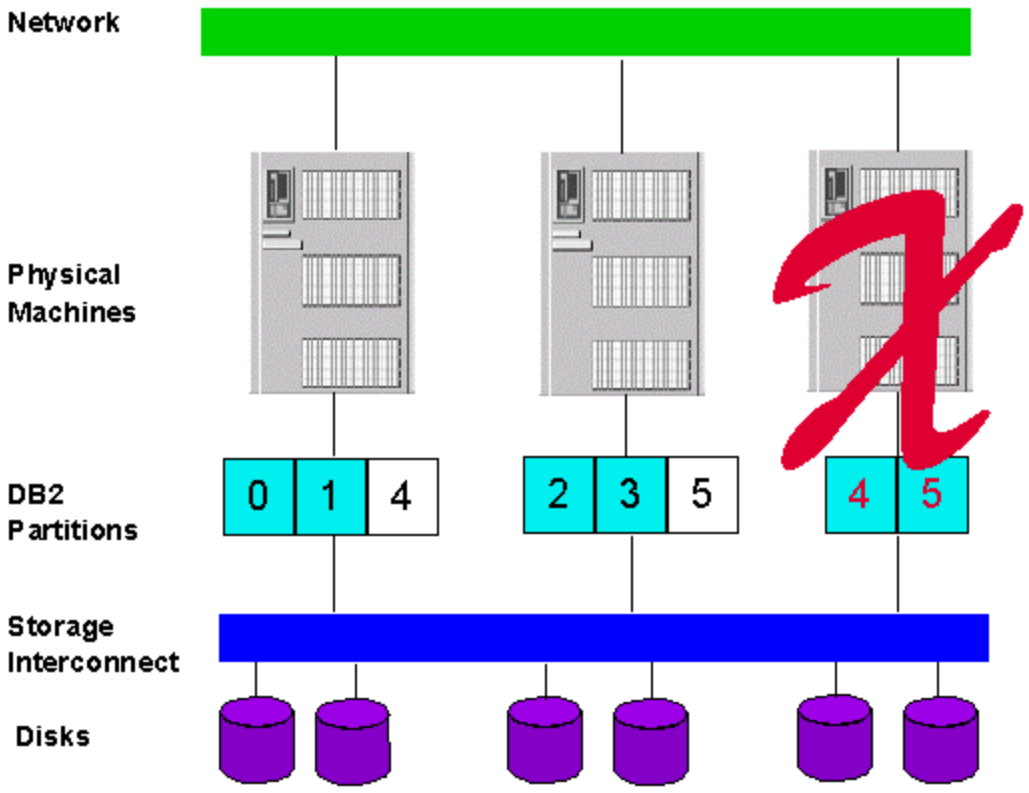


Figure 16. Mutual takeover load-balancing example

For this configuration, if the third machine fails, then Partition 4 and Partition 5 will move to machines 1 and 2, respectively. This maintains an evenly distributed workload across the system.

This particular load balancing example requires each DB2 partition to be located within different MSCS groups as each MSCS group can be configured to failover to different machines.

1. Initially MSCS is installed and a six partition DB2 instance is configured and stopped.

```
C:\>db2nlist /s
List of nodes for instance "DB2MPP" is as follows:
Node: "0" Host: "stress01" Machine: "stress01" Port: "0" - "stopped"
Node: "1" Host: "stress01" Machine: "stress01" Port: "1" - "stopped"
Node: "2" Host: "stress02" Machine: "stress02" Port: "0" - "stopped"
Node: "3" Host: "stress02" Machine: "stress02" Port: "1" - "stopped"
Node: "4" Host: "stress03" Machine: "stress03" Port: "0" - "stopped"
Node: "5" Host: "stress03" Machine: "stress03" Port: "1" - "stopped"
```

2. Create a DB2MSCS input file and then execute the DB2MSCS utility using this input file. This configuration will assume that only Partition 0 will receive incoming remote client requests and thus will be the only partition that has an MSCS TCP/IP resource defined on the public network.

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

GROUP_NAME=DB2 Group 0
DB2_NODE=0 10.1.1.5
IP_NAME=mscs3
IP_ADDRESS=9.26.97.16
IP_SUBNET=255.255.254.0
IP_NETWORK=Public Network
IP_NAME=ether0
IP_ADDRESS=10.1.1.5
IP_SUBNET=255.0.0.0
IP_NETWORK=Private Network
NETNAME_NAME=mynetname
NETNAME_VALUE=mynetname
NETNAME_DEPENDENCY=ether0
DISK_NAME=Disk N:
TARGET_DRVMAP_DISK=Disk N:

GROUP_NAME=DB2 Group 1
DB2_NODE=1 10.1.1.6
IP_NAME=ether1
IP_ADDRESS=10.1.1.6
IP_SUBNET=255.0.0.0
IP_NETWORK=Private Network
```

```
DISK_NAME=Disk O:
TARGET_DRVMAP_DISK=Disk O:

GROUP_NAME=DB2 Group 2
DB2_NODE=2 10.1.1.7
IP_NAME=ether2
IP_ADDRESS=10.1.1.7
IP_SUBNET=255.0.0.0
IP_NETWORK=Private Network
DISK_NAME=Disk P:
TARGET_DRVMAP_DISK=Disk P:

GROUP_NAME=DB2 Group 3
DB2_NODE=3 10.1.1.8
IP_NAME=ether3
IP_ADDRESS=10.1.1.8
IP_SUBNET=255.0.0.0
IP_NETWORK=Private Network
DISK_NAME=Disk Q:
TARGET_DRVMAP_DISK=Disk Q:

GROUP_NAME=DB2 Group 4
DB2_NODE=4 10.1.1.9
IP_NAME=ether4
IP_ADDRESS=10.1.1.9
IP_SUBNET=255.0.0.0
IP_NETWORK=Private Network
DISK_NAME=Disk R:
TARGET_DRVMAP_DISK=Disk R:

GROUP_NAME=DB2 Group 5
DB2_NODE=5 10.1.1.10
IP_NAME=ether5
IP_ADDRESS=10.1.1.10
IP_SUBNET=255.0.0.0
IP_NETWORK=Private Network
DISK_NAME=Disk S:
TARGET_DRVMAP_DISK=Disk S:
```

3. Determine the failover preferences for each DB2 group and enable automatic failback if desired. Table 1 illustrates a failover preference such that if a machine fails, the failing partitions will failover to different machines:

	First Preference	Second Preference
DB2MPP Partition 0	STRESS01	STRESS02
DB2MPP Partition 1	STRESS01	STRESS03
DB2MPP Partition 2	STRESS02	STRESS01

DB2MPP Partition 3	STRESS02	STRESS03
DB2MPP Partition 4	STRESS03	STRESS01
DB2MPP Partition 5	STRESS03	STRESS02

Table 1. Preferred owners for mutual takeover load-balancing configuration

4. Start all DB2 resources from Cluster Administrator and create or restore all databases on the highly available disks.

Multiple cluster configuration

MSCS clusters can span up to four machines currently and up to eight machines with Windows .NET. However, if the ESE instance being used spans more machines than are available in the cluster, then multiple clusters can be used. Each DB2 partition can only failover to machines within the same cluster. If all machines in any particular cluster are in a failed state, then all partitions that reside within that cluster will not be available.

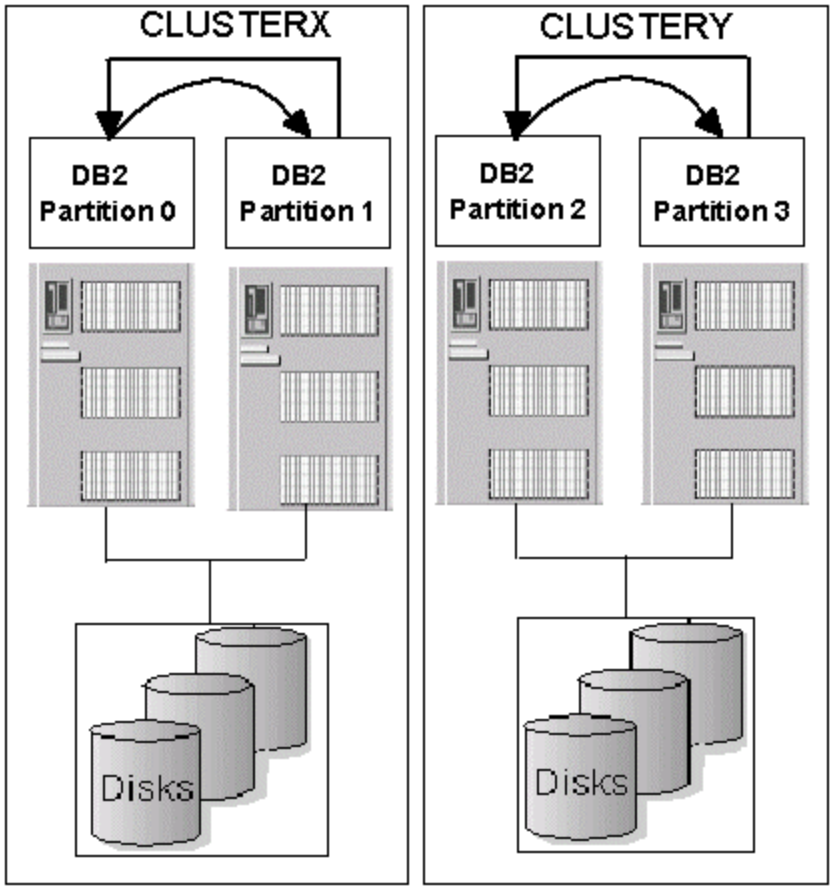


Figure 17. DB2 instance spread across multiple clusters

Figure 17 provides an illustration of a four-partition ESE instance spread across two clusters with each cluster comprised of two machines. The initial configuration of each cluster is a mutual takeover configuration as there is a DB2 partition running on each node within the cluster.

Note: More than two clusters can be used, and each cluster does not necessarily have to have the same type of configuration.

The following provides an example of how to configure the example in Figure 17. Assume Partition 0 owns Disk N, Partition 1 owns Disk O, Partition 2 owns Disk P, and Partition 3 owns Disk Q.

1. Initially you install MSCS and configure and stop a four-partition DB2 instance. CLUSTERX consists of machines STRESS01 and STRESS02 while CLUSTERY consists of machines STRESS03 and STRESS04.

```
C:\>db2nlist /s
List of nodes for instance "DB2MPP" is as follows:
Node: "0" Host: "stress01" Machine: "stress01" Port: "0" - "stopped"
Node: "1" Host: "stress02" Machine: "stress02" Port: "0" - "stopped"
Node: "2" Host: "stress03" Machine: "stress03" Port: "0" - "stopped"
Node: "3" Host: "stress04" Machine: "stress04" Port: "0" - "stopped"
```

2. Create a DB2MSCS input file and then execute the DB2MSCS utility using this input file. This configuration will assume that only Partition 0 will receive incoming remote client requests and thus will be the only partition that has an MSCS TCP/IP resource defined on the public network.

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

GROUP_NAME=DB2 Group 0
DB2_NODE=0 10.1.1.5
IP_NAME=mscs3
IP_ADDRESS=9.26.97.16
IP_SUBNET=255.255.254.0
IP_NETWORK=Public Network
IP_NAME=ether0
IP_ADDRESS=10.1.1.5
IP_SUBNET=255.0.0.0
IP_NETWORK=Private Network
NETNAME_NAME=mynetname
NETNAME_VALUE=mynetname
NETNAME_DEPENDENCY=ether0
DISK_NAME=Disk N:
TARGET_DRVMAP_DISK=Disk N:
```

```
GROUP_NAME=DB2 Group 1
DB2_NODE=1 10.1.1.6
IP_NAME=ether1
IP_ADDRESS=10.1.1.6
IP_SUBNET=255.0.0.0
IP_NETWORK=Private Network
DISK_NAME=Disk O:
TARGET_DRVMAP_DISK=Disk O:

CLUSTER_NAME=CLUSTERY
GROUP_NAME=DB2 Group 2
DB2_NODE=2 10.1.1.7
IP_NAME=ether2
IP_ADDRESS=10.1.1.7
IP_SUBNET=255.0.0.0
IP_NETWORK=Private Network
DISK_NAME=Disk P:
TARGET_DRVMAP_DISK=Disk P:

GROUP_NAME=DB2 Group 3
DB2_NODE=3 10.1.1.8
IP_NAME=ether3
IP_ADDRESS=10.1.1.8
IP_SUBNET=255.0.0.0
IP_NETWORK=Private Network
DISK_NAME=Disk Q:
TARGET_DRVMAP_DISK=Disk Q:
```

3. Determine the failover preferences for each DB2 group and enable automatic failback if desired.

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

Note: When issuing the `db2set` command in a multiple cluster environment, ensure the command is run on each cluster.

Appendix A - Limitations and restrictions

When running DB2 in an MSCS environment, the following limitations and restrictions apply:

- Since MSCS uses the drive letter when referring to Physical Disk resources, we recommend that you use drive letters when determining the path for DB2 databases and tablespace containers.
- Since MSCS cannot manage raw disk devices, DB2 should not be configured to use raw disk devices.
- The DB2 partition must be managed from an MSCS interface such as Cluster Administrator. MSCS will not monitor resources started outside of its control, because it will not be aware the resource has been started. MSCS will also treat any attempts to stop a DB2 partition outside of its control as a resource failure, if that resource was initially brought online by Cluster Administrator. Thus, do not use mechanisms such as DB2STOP, DB2START, and the DB2 Control Center to start and stop a DB2 instance within this environment.

Appendix B - Frequently asked questions

1. Q. My ESE instance will not start on other machines in the cluster.
   A. The DB2 ESE instance reserves ports in the services files which are used during startup. If the ports are not allocated in the services file, DB2 will automatically add these entries into the services file. Ensure these ports are available on all machines in the cluster. Below is a sample entry from a services file with 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.db2
   Warning: The message file is missing
   DB2MSCS processing complete, rc = 126
   ```

   A. After DB2 has been installed, each machine in the cluster must be rebooted before proceeding with running the DB2MSCS utility.

3. 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". Win32
   error: ""
   ```

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

4. 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 specified does not exist within the cluster. Ensure the name of the disk in the input configuration file is exactly identical to the name specifed within Cluster Administrator (eg., “Disk O:”).

5. Q. I ran the DB2MSCS utility and it does not execute successfully.
   A. Refer to the message reference to determine the course of action based on the return code

from the DB2MSCS utility.

6. Q. When I execute the `db2nlist` or `db2nchg` command it fails after a couple minutes with a communiction error.
   A. Ensure that all TCP/IP addresses used have a corresponding entry in the Domain Name Server or have an entry in the hosts file on each machine.

7. Q. My partition does not failover.
   A. The partition must be started through a cluster interface, such as Cluster Administrator, or the cluster will not be aware it must try to keep this DB2 partition online.

8. Q. I do a `db2stop` and the partitions automatically restart.
   A. If the DB2 partitions were started through Cluster Administrator, it must be stopped through a cluster interface. MSCS treats `db2stop` as a resource failure and will attempt to bring the DB2 resource back online.

9. Q. I try to take the group containing the DB2 partition offline, but the DB2 resource does not successfully come offline.
   A. Ensure DB2_FALLBACK is set to ON or YES for that partition (`db2set DB2_FALLBACK=ON`). To successfully bring the group containing the DB2 partition offline at this point, stop the cluster service on the machine where the DB2 partition is trying to come offline.

10. Q. My `create database` command fails in my multiple partition configuration.
    A. Ensure the drive mapping has been done successfully. All DB2 resources will need to be taken offline and then brought back online for the manual drive mapping to take effect.

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

12. 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 resides.

13. Q. After the failover, it appears some of my transactions are waiting upon a lock.
    A. It is possible that their may be indoubt transactions within your database. Use the following command to manually resolve the indoubt transactions:

    ```
    db2 list indoubt transactions with prompting
    ```

    For more information on indoubt transactions, refer to the DB2 UDB documentation.

14. Q. I ran the DB2MSCS utility with the –u option to decluster my instance but it failed.
    A. When declustering an instance, ensure the DB2MSCS utility is run from the instance owning partition. If the instance is only partially declustered, continue the process using the manual steps described in Appendix C of this paper.

15. Q. The commands in the pre-online, post-online, pre-offline and post-offline script fail with authentication errors.
    A. These scripts are run under the owner of the Cluster Server service. Ensure the owner of the Cluster Server has sufficient privileges to execute the commands.

Note: A trace of the execution of the DB2MSCS utility can be envoked by executing the following command:

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

This trace will be beneficial to IBM support for problem determination if necessary.

Appendix C - Declustering an instance

If you no longer want to keep your Administration Server or instance clustered, you can decluster ths instance using either the DB2MSCS utility or it can be done manually. The following examples will show how to decluster the Mutual Takeover Multiple Partition Configuration.

Using DB2MSCS to decluster an instance

1. Back up the database that resides within instance DB2MPP. If DB2 drive mapping was used or if you need to place the database on a different drive, then drop the database.
2. Put all the DB2 Groups on the machine they were originally on after the `db2mscs` utility was executed.
3. From the instance owning partition, run the `db2mscs` utility with the –u option. The Administration Server should be declustered before the ESE instance.

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

4. Restore the databases that have been backed up.

Manually declustering an instance

1. Back up the database that resides within instance DB2MPP and then drop the database.
2. Bring only the DB2 resources offline from Cluster Administrator leaving the disks, IP addresses, network name, and fileshare online.
3. Drop the Administration Server and the DB2MPP instance from one of the machines in the cluster. This will drop them from all machines.

```
db2admin drop
db2idrop db2mpp
```

Tip: When dropping the instance, all instance information will be lost. Ensure that you save any instance information that may be required in the future, such as database manager configuration parameters and DB2 profile variable settings:

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

4. From Cluster Administrator, drop the DB2 resources corresponding to each DB2 partition and the Administration Server as well as the corresponding IP addresses, network name, and

fileshare. Move all Physical Disk resources back to their initial groups. Then drop the groups associated with each partition, as no resources will exist within these groups.

5. If no other DB2 instances are configured for the MSCS cluster, the DB2 resource type can be dropped. The following command only needs to be run from one machine in the cluster:

```
db2wolfi u
```

6. Recreate the Administration Server and the DB2MPP instance.

7. Restore the databases on the new instance and reconfigure the configuration paramaters as desired.

Appendix D - Manually performing steps done by the DB2MSCS utility

The example presented below will manually configure the DB2MPP instance identically to the way it was configured in the Mutual Takeover Multiple Partition Configuration section of this paper. We recommend that you use the DB2MSCS utility; however, decomposing the steps may help during problem determination if errors occur using the DB2MSCS utility.

1. Initially MSCS and DB2 UDB ESE are installed on all machines.
2. Create a new instance called DB2MPP if it does not exist.

```
db2icrt DB2MPP -s ese -P \\wa9\c$\sqllib -u mydom\db2admin,xxx -h wa9 -
r 60000,60003
db2ncrt /n:1 /u:mydom\db2admin,xxx /i:db2mpp /h:wa10 /m:wa10 /p:0
/o:wa9

C:\>db2nlist /s
List of nodes for instance "DB2MPP" is as follows:
Node: "0" Host: "wa9" Machine: "wa9" Port: "0" - "stopped"
Node: "1" Host: "wa10" Machine: "wa10" Port: "0" - "stopped"
```

3. Stop the instance DB2MPP with the DB2STOP command if the instance is running.
4. Install the DB2 resource type from WA9:

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

If the `db2wolfi` command comes back with an "Error : 183", then it is already installed. To confirm, the resource type can be dropped and added again. Also, the resource type will not show up in Cluster Administrator if it does not exist:

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

5. From Cluster Administrator, create two new groups called "DB2 Group 0" and "DB2 Group 1," and from Cluster Administrator move these groups onto WA9 and WA10, respectively. The preferred owner should be set to the current machine the group is residing on.

6. From Cluster Administrator do a "**Change Group**" to move disks F and G into "DB2 Group 0" and "DB2 Group 1," respectively.

Note: For the "**Change Group**" command to work, the two groups must be on the same

machine.

7. For "DB2 Group 0," from Cluster Administrator, create a new resource type of type "IP Address," that resides on the Public Network. This will be a highly available IP address, and this address should not correspond to any machine on the network. In this example, we will be using `mscs11` as defined in the Mutual Takeover Multiple Partition Configuration section. Bring the IP Address resource online and ensure that the address can be pinged from a remote machine.

8. Create a highly available TCP/IP address on the Private Network for each DB2 group from Cluster Administrator. We will call them **ether0** and **ether1**.

9. Assign the new TCP/IP addresses created in step 8 to its corresponding partition:

```
C:\>db2nchg /n:0 /g:10.1.1.1
C:\>db2nchg /n:1 /g:10.1.1.2
```

10. From Cluster Administrator create a network name in "DB2 Group 0." Create a network name called **mynetname** that is dependant on the IP Address **ether0** and then bring this resource online.

11. The next step is to create a file share. First create a subdirectory on Disk F called `db2profs`. From Cluser Administrator create a file share called DB2MSCS-DB2MPP dependent on mynetname and Disk F. When prompted for the path of the file share, specify `f:\db2profs`. Bring the file share online from Cluster Administrator.

12. Verify the fileshare is accessible:

```
dir \\mynetname\db2mscs-db2mpp
```

13. Create a DB2 resoure corresponding to each DB2 partition of type DB2. Since the instance used is DB2MPP, the resources must be named DB2MPP-0 and DB2MPP-1, corresponding to each DB2 partition. Create these DB2 resources in "DB2 Group 0" and "DB2 Group 1," respectively, and each DB2 resource should be dependant on all other resources within the same group. Do not bring the new DB2 resources online yet.

14. From WA9 use the `db2iclus` command to transfrom the DB2 instance into a clustered instance. This will also place the instance directory onto the newly created file share:

```
C:\>db2iclus migrate /i:db2mpp /c:mycluster /m:wa9
/p:\\mynetname\db2mscs-db2mpp
```

15. From WA9 use the `db2iclus` command to add the remaining cluster machines to the DB2 cluster list:

    ```
    C:\> db2iclus add /i:db2mpp /c:mycluster /m:wa10
    /u:mydom\db2admin,xxx
    ```

16. Perform the DB2 drive mapping so we can create the database on the ‘f’ drive:
    `C:\>db2drvmp add 1 f g`

17. Configure the DB2 groups for failback if desired via Cluster Administrator and issue
    C:\>`db2set DB2_FALLBACK=YES`
18. Bring all DB2 resources online via Cluster Administrator.
19. Create or restore all databases putting all data on the shared drives.
20. Test the failover configuration.

Appendix E - Sample application program

The following excerpt from a sample program illustrates how to have an application retry connecting to the database upon a connection failure as well as retry SQL statements based on particular error codes:

```
int CSampleODBCObj::ExecuteSQLWithRetry(char * stmt)
{
   int rc = 0;
   int retry = 0;
   do {
      retry = 0;
      rc = ExecuteSQL( stmt );
      if (rc != 0)
      {
         if((strcmp( (char *)&(m_sqlstate[0]),"40003" ) == 0) ||
            (strcmp( (char *)&(m_sqlstate[0]),"08003" ) == 0) ||
            (strcmp( (char *)&(m_sqlstate[0]),"08007" ) == 0) ||
            (strcmp( (char *)&(m_sqlstate[0]),"08S01" ) == 0))
         {
            Disconnect();
            while (bContinue && (Connect() != 0))
               Sleep(1000);

            retry = 1;
         }
         else if (strcmp( (char *)&(m_sqlstate[0]),"40504" ) == 0)
         {
            // Just retry the transaction without reconnect
            retry = 1;
         }
      }
   } while ( bContinue && retry ); /* enddo */
   return rc;
}
```